...SITY
...LICKE
UNIVERSITEIT
BRABANT

POSTBOX 90153
5000 LE TILBURG
THE NETHERLANDS

[redacted]

DEPARTMENT OF ECONOMICS

RESEARCH MEMORANDUM

[redacted]

A NUMERICAL APPROACH TO CYCLIC-SERVICE QUEUEING MODELS

J.P.C. Blanc

FEW 312

A numerical approach to cyclic-service queueing models

J.P.C. Blanc
Tilburg University, Faculty of Economics
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Abstract

An iterative numerical technique for the evaluation of queue length distributions is applied to multi-queue systems with one server and cyclic service discipline with Bernoulli schedules. The technique is based on power-series expansions of the state probabilities as functions of the load of the system. The convergence of the series is accelerated by applying an adapted form of the epsilon algorithm. Attention is paid to economic use of memory space.

Keywords: power-series algorithm, traffic intensity, waiting time, epsilon algorithm, memory space.

1. Introduction

Queueing systems with more than one waiting line are very hard to analyse when the joint queue length distribution is not of some kind of product form. In [7], [1], [2], [3] a numerical technique has been developed for the evaluation of performance measures for such multi-queue systems. The technique is based on power-series expansions of the state probabilities as functions of one parameter (the traffic intensity) of the system. The coefficients of these power-series can be recursively calculated for a large class of multi-queue models. The coefficients of the power-series expansions of the moments of the queue length distributions follow directly from those of the state probabilities. In most instances a bilinear transformation ensures convergence of the power-series over the whole range of traffic intensities for which the system is stable. We have introduced in [2], [3] extrapolations of the coefficients of the power-series in order to accelerate the convergence of the series. One of these extrapolations will be combined with the epsilon algorithm, cf. [6], [12], in the present paper. The advantages of the present technique are that quantities are calculated iteratively, that it is relatively easy to compute additional terms of the power-series in order to increase accuracy, that algorithms for accelerating the convergence of sequences can be applied, and that, once the coefficients of the power-series have been obtained, it requires little effort to compute performance measures for different values of the traffic intensity (compare with numerical techniques based on truncation of the state space and solution of large sets of balance equations). The main drawback is the large amount of memory space necessary to store the coefficients of the power-series of the state probabilities. The available memory space mainly limits the size of the models which can be handled. Therefore, attention will be paid to optimize the use of available memory space.

The power-series algorithm will be applied to a multi-queue model with one server and cyclic service discipline with Bernoulli schedules. This kind of model is often used to study distributed computer systems with a single communication channel and a cyclic access scheme. Several authors have derived general relations or have proposed approximations for the mean waiting times in such systems, cf. [4], [5], [8], [9], [10],

[11]. Our approach provides exact data for moderate sized systems, which are of interest in itself for studying the interaction between queues, and which may be helpful in finding and validating approximations for large scale systems.

The organisation of the paper is as follows. The multi-queue model with cyclic service will be described in section 2. Section 3 is devoted to the derivation of the scheme for calculating the coefficients of the power-series. An adapted form of the epsilon algorithm is introduced in section 4. Section 5 contains remarks on the implementation of the power-series algorithm, section 6 some numerical examples of the multi-queue model. Possible extensions of the model and the algorithm will be discussed in section 7.

2. The multi-queue model

The system consists of s queues. Jobs arrive at queue j according to a Poisson process with rate λ_j, $j = 1,\ldots,s$. The single server inspects the queues in cyclic order, i.e. queue $1,2,\ldots$, and after queue s again queue 1, etc. When the server finds queue j non-empty, he serves the first arrived job in this queue. After completion of the service of a job at queue j the server starts another service at this queue with probability q_j when this queue is not empty; otherwise the server switches to the next queue ($j = 1,\ldots,s$). The times for switching from one queue to the next will be neglected in the present study. Service times at queue j are assumed to be identically, exponentially distributed with mean $1/\mu_j$, $j = 1,\ldots,s$. Each queue may contain an unbounded number of jobs. See section 6 for a discussion on the relaxation of some of these model assumptions. Note that the server visits queue j according to a Bernoulli schedule with parameter q_j; this includes exhaustive service ($q_j = 1$) and non-exhaustive or one-limited service ($q_j = 0$), $j = 1,\ldots,s$.

First the condition for ergodicity of the system will be considered, cf. [8]. The sum of the arrival processes at the different queues is a Poisson process with rate $\Lambda = \Sigma_{j=1}^{s}\lambda_j$. The service rate of an arbitrary job is μ_j with probability λ_j/Λ, $j = 1,\dots,s$. Hence, the load or traffic intensity ρ of the system is in a natural way defined by

$$\rho := \Lambda \sum_{j=1}^{s} \frac{1}{\mu_j} \frac{\lambda_j}{\Lambda} = \sum_{j=1}^{s} \frac{\lambda_j}{\mu_j}, \tag{2.1}$$

and a necessary and sufficient condition for ergodicity of the system is

$$\rho < 1. \tag{2.2}$$

Because the traffic intensity ρ will be used as variable in power-series expansions, the arrival rates will be written as

$$a_j\rho = \lambda_j, \qquad j = 1,\dots,s. \tag{2.3}$$

It will be assumed that the system is in steady state and hence (2.2) will hold. Let N_j denote the number of jobs in queue j (waiting or being served), $j = 1,\dots,s$. The supplementary variable H, indicating the queue to which the server attends, is introduced in order to transform the queue length process into a Markov process (which can be described as a multi-dimensional quasi birth-death process). Let $\bar{n} = (n_1,\dots,n_s)$ be a vector with non-negative integer entries. Note, that when the system is empty (in state $\bar{0}$) the value of H is not determined. Therefore, the probability that the system is empty at load ρ will be denoted by $p(\rho;\bar{0})$. For $\bar{n} \neq \bar{0}$ the state probabilities are defined as follows: for $h = 1,\dots,s$, $0 \le \rho < 1$,

$$p(\rho;\bar{n},h) := \Pr\{N_j = n_j,\ j = 1,\dots,s;\ H = h, \text{ at load } \rho\}. \tag{2.4}$$

Let $I\{E\}$ stand for the indicator function of the event E, and let $\bar{e}_j$ be a vector with zero entries except an entry of one at the j^{th} position ($j = 1,\dots,s$).

When the server attends queue h, a state $\bar{n}$ with $n_h = 1$ could only have been entered through an arrival at queue h if all queues were empty ($h = 1,\dots,s$). Note further that the server may reach queue h from any queue $h-j$ ($j = 1,\dots,s$) on condition that all intermediate queues $h-j+1,\dots,h-1$ are empty ($h = 1,\dots,s$; read here and below queue $i+s$ for queue i whenever

$i < 1$). With this in mind, the balance equations for the state probabilities (2.4) are readily verified to be, for $h = 1,\dots,s$, $n_h > 0$,

$$[\rho \sum_{j=1}^{s} a_j + \mu_h]p(\rho;\bar{n},h) = a_h \rho p(\rho;\bar{0}) I\{\bar{n} = \bar{e}_h\}$$

$$+ \rho \sum_{j=1}^{s} a_j p(\rho;\bar{n} - \bar{e}_j,h) I\{n_j > 0; n_j > 1 \text{ if } j = h\}$$

$$+ [q_h + (1-q_h) I\{n_j = 0 \ \forall j,\ j \neq h\}]\mu_h p(\rho;\bar{n} + \bar{e}_h,h)$$

$$+ \sum_{j=1}^{s-1} [1 - q_{h-j} + q_{h-j} I\{n_{h-j} = 0\}] I\{n_i = 0,\ i = h-j+1,\dots,h-1\} \times$$

$$\times \mu_{h-j} p(\rho;\bar{n} + \bar{e}_{h-j};h-j); \tag{2.5}$$

$$\rho \sum_{j=1}^{s} a_j p(\rho;\bar{0}) = \sum_{j=1}^{s} \mu_j p(\rho;\bar{e}_j,j). \tag{2.6}$$

3. The power-series algorithm

The power-series algorithm will be discussed briefly in this section. The reader is referred to [2], [7] for more details and a motivation of the method. First, introduce the bilinear mapping of the interval [0,1] onto itself,

$$\rho = \rho(\vartheta) = \frac{\vartheta}{1 + G - G\vartheta} \ \left(\vartheta = \frac{(1+G)\rho}{1 + G\rho}\right),\ G \geq 0. \tag{3.1}$$

Then, introduce the following power-series expansions, for $h = 1,\dots,s$,

$$p(\rho(\vartheta);\bar{n},h) = \vartheta^{n_1+\dots+n_s} \sum_{k=0}^{\infty} \vartheta^k b(k;\bar{n},h),\ \bar{n} \neq \bar{0},$$

$$p(\rho(\vartheta);\bar{0}) = \sum_{k=0}^{\infty} \vartheta^k b(k;\bar{0}). \tag{3.2}$$

Replace ρ by ϑ in the balance equations (2.5) according to (3.1), and substitute the power-series (3.2) into these equations. Equating the coefficients of corresponding powers of ϑ in the resulting equations leads to the following iterative scheme for computing the coefficients of the power-series (3.2): for $h = 1,\dots,s$, $n_h > 0$, for $k = 0,1,2,\dots$,

$$
\begin{aligned}
(1+G)\mu_h b(k;\bar{n},h) = {} & [G\mu_h - \sum_{j=1}^{s} a_j] I\{k > 0\} b(k-1;\bar{n},h) \\
& + a_h b(k;\bar{0}) I\{\bar{n} = \bar{e}_h\} \\
& + \sum_{j=1}^{s} a_j b(k;\bar{n}-\bar{e}_j,h) I\{n_j > 0; n_j > 1 \text{ if } j = h\} \\
& + [q_h + (1-q_h) I\{n_j = 0\ \forall j,\ j \neq h\}]\mu_h [(1+G) b(k-1;\bar{n}+\bar{e}_h,h) I\{k > 0\} \\
& \quad - G b(k-2;\bar{n}+\bar{e}_h,h) I\{k > 1\}] \\
& + \sum_{j=1}^{s-1} [1 - q_{h-j} + q_{h-j} I\{n_{h-j} = 1\}] I\{n_i = 0, i = h-j+1,\dots,h-1\} \times \\
& \times \mu_{h-j} [(1+G) b(k-1;\bar{n}+\bar{e}_{h-j},h-j) I\{k>0\} - G b(k-2;\bar{n}+\bar{e}_{h-j},h-j) I\{k>1\}].
\end{aligned}
\tag{3.3}
$$

To determine the coefficients of $p(\rho(\vartheta);\bar{0})$ the law of total probability is used instead of (2.6) to complete the recursive scheme, because the term with $b(k;\bar{0})$ vanishes in (2.6). Substituting (3.1) and (3.2) into the law of total probability gives:

$$b(0;\bar{0}) = 1,$$

$$b(k;\bar{0}) = - \sum_{0<n_1+\dots+n_s\le k} \cdots \sum \sum_{h=1}^{s} b(k-n_1-\dots-n_s;\bar{n},h), \quad k = 1,2,\dots \tag{3.4}$$

There are several ways to compute the coefficients $b(k;\bar{n},h)$ recursively from (3.3) and (3.4). One convenient way is the following. Calculate all coefficients $b(k;\bar{n},h)$ with $k+n_1+\dots+n_s = m$ before those with $k+n_1+\dots n_s =$

$m+1$ ($m = 0,1,2,\dots$), and on each hyperplane $k+n_1+\dots+n_s = m$, m fixed, calculate all coefficients $b(k;\bar{n},h)$ with $k=j$ before those with $k = j+1$, $j = 0,1,\dots,m-1$ ($m = 0,1,\dots$). See also (A.4) in the appendix.

Once the coefficients of the power-series expansions of the state probabilities have been determined, those of the moments of the queue length distribution can be obtained as well. Write

$$E\{N_j^{\nu}\} = \sum_{k=1}^{\infty} \vartheta^k f_{\nu}(k;j),\ j = 1,\dots,s,\ \nu = 1,2,\dots \tag{3.5}$$

It follows readily from (3.5) and (3.2) that for $j = 1,\dots,s$, $\nu = 1,2,\dots$, $k = 1,2,\dots$,

$$f_{\nu}(k;j) = \sum_{0\le n_1+\dots+n_s\le k} \dots \sum \ \sum_{h=1}^{s} n_j^{\nu}\, b(k-n_1-\dots-n_s;\bar{n},h). \tag{3.6}$$

It is more convenient for obtaining moments of the queue length distribution to compute first their coefficients via (3.6) and then to use (3.5) than to compute first the state probabilities via (3.2) and then the moments directly from the state probabilities. In the first way algorithms for accelerating the convergence can be applied to partial sums of the series (3.5); see section 4. Moreover, the second way will be more laborous, when one is not interested in the (many!) state probabilities themselves.

This section is concluded with a discussion of the stationary waiting time (W_h) distribution of jobs arriving at queue h ($h = 1,\dots,s$). The number of jobs at queue h left behind by a job departing from that queue is equal to the number of jobs that arrived at queue h during the sojourn time of the departing job. Because arrivals occur according to a Poisson process, this implies, cf. [10], for $h = 1,\dots,s$, for $|z| \le 1$,

$$E\{z^{N_h}\} = \frac{1}{1 + (1-z)\lambda_h/\mu_h}\, E\{e^{-\lambda_h(1-z)W_h}\}. \tag{3.7}$$

The moments of the waiting time distributions can be obtained from the moments of the marginal queue length distributions through this relations (3.7).

Let W be the waiting time of an arbitrary job, irrespectively of the queue at which it arrives, in steady state. Then, with (3.7) and (2.1),

$$E\{W\} = \sum_{h=1}^{s} \frac{\lambda_h}{\Lambda} E\{W_h\} = [E\{\sum_{h=1}^{s} N_h\} - \rho]/\Lambda. \tag{3.8}$$

Finally, we note that the expected values of the waiting times for jobs in the different queues of a cyclic service system satisfy the following conservation law, cf. [4],[11],

$$\sum_{h=1}^{s} \frac{a_h}{\mu_h} E\{W_h\} = \frac{\rho}{1-\rho} \sum_{h=1}^{s} a_h/\mu_h^2. \tag{3.9}$$

This relation provides a useful check on the accuracy of the computations. With the aid of Little's formula we obtain from (3.9) the following relation for the mean queue lengths:

$$\sum_{h=1}^{s} \frac{1}{\mu_h} E\{N_h\} = \frac{\rho}{1-\rho} \sum_{h=1}^{s} a_h/\mu_h^2. \tag{3.10}$$

In the special case that all mean service times are equal (i.e. $\mu_h = \mu$, $h = 1,\dots,s$) then (3.9) and (3.10) lead with (3.8) and (2.1) to

$$E\{\sum_{h=1}^{s} N_h\} = \mu E\{W\} = \frac{\rho}{1-\rho}, \tag{3.11}$$

the well-known results for the M/M/1 system. Note that the relations (3.8), (3.9), (3.10), (3.11) hold for any set of Bernoulli parameters $\{q_j, j = 1,\dots,s\}$.

4. Application of the epsilon algorithm

The epsilon algorithm aims to accelerate the convergence of slowly convergent sequences or to determine a value for divergent sequences, cf.

[12], [6]. The epsilon algorithm consists of the following triangular recursive scheme: for $m = 0,1,\dots$, $\kappa = 0,1,\dots$,

$$\varepsilon_{\kappa+1}^{(m)} = \varepsilon_{\kappa-1}^{(m+1)} + [\varepsilon_{\kappa}^{(m+1)} - \varepsilon_{\kappa}^{(m)}]^{-1}, \tag{4.1}$$

with initial values, for $m = 0,1,\dots$,

$$\varepsilon_{-1}^{(m)} = 0, \ \varepsilon_{0}^{(m)} = S_m; \tag{4.2}$$

here S_m, $m = 0,1,\dots$, is the partial sum of a series. Only the even sequences $\{\varepsilon_{2\kappa}^{(m)}, m = 0,1,\dots\}$ will be sequences which may converge faster to a limit than $\{S_m, m = 0,1,\dots\}$, $\kappa = 1,2,\dots$. The odd sequences $\{\varepsilon_{2\kappa+1}^{(m)}, m = 0,1,\dots\}$ are just intermediate steps in the calculation scheme, $\kappa = 0,1,\dots$. When S_m is the partial sum of a power-series, say

$$S_m = S_m(\vartheta) = \sum_{i=0}^{m} c_i \vartheta^i, \quad m = 0,1,\dots, \tag{4.3}$$

then the epsilon algorithm transforms this sequence of polynomials into sequences of quotients of two polynomials. More precisely, $\varepsilon_{2\kappa}^{(m-2\kappa)}$ will be a quotient of a polynomial of degree $m-\kappa$ over a polynomial of degree κ, and

$$|S_m - \varepsilon_{2\kappa}^{(m-2\kappa)}| = O(\vartheta^{m+1}), \ \vartheta \to 0, \ \kappa = 1,2,\dots, \ m = 2\kappa, 2\kappa+1,\dots; \tag{4.4}$$

see [12]. Because many queueing systems have the property that the ν^{th} moments of the queue length distribution are of order $(1-\rho)^{-\nu}$ as $\rho\uparrow 1$, $\nu = 1,2,\dots$, we propose to modify the initial values for the epsilon algorithm as follows when this algorithm is applied to accelerate the convergence of power-series for moments, cf. (3.5). Before applying the epsilon algorithm we first extrapolate the coefficients of the power-series to take into account the pole at $\vartheta = 1$. This extrapolation has been introduced in [1] and [2]. It means that we take for first order moments

$$\varepsilon_{0}^{(m)} = S_m + c_m \frac{\vartheta^{m+1}}{1-\vartheta}, \quad m = 1,2,\dots, \tag{4.5}$$

and for second order moments

$$\varepsilon_0^{(m)} = S_m + [c_m + \frac{c_m - c_{m-1}}{1 - \vartheta}] \frac{\vartheta^{m+1}}{1-\vartheta}, \quad m = 1,2,\ldots, \tag{4.6}$$

instead of the second relation of (4.2); here S_m is of the form (4.3) and c_m, $m = 1,2,\ldots$, stand for coefficients of a series as defined in (3.5). It is our experience that the use of (4.5) and (4.6) instead of (4.2) leads to considerably faster convergence, cf. [2], and this property is preserved in higher order sequences $\{\varepsilon_{2\kappa}^{(m)}, m = 1,2,\ldots\}$, $\kappa = 1,2,\ldots$, produced by the epsilon algorithm. For instance, when relation (4.5) is used as initial sequence, then

$$\varepsilon_2^{(m-2)} = S_m + c_m \frac{\vartheta^{m+1}}{1-\vartheta} + \frac{\sigma\vartheta^{m+1}(c_m - c_{m-1})}{(1-\vartheta)(1-\sigma\vartheta)}, \tag{4.7}$$

with

$$\sigma = \frac{c_m - c_{m-1}}{c_{m-1} - c_{m-2}}, \quad m = 2,3,\ldots. \tag{4.8}$$

For comparison, when relation (4.2) were used as initial sequence, then

$$\varepsilon_2^{(m-2)} = S_m + \frac{\tilde{\sigma}\vartheta^{m+1} c_m}{(1-\tilde{\sigma}\vartheta)}, \ \tilde{\sigma} = \frac{c_m}{c_{m-1}}, \ m = 2,3,\ldots. \tag{4.9}$$

It will be clear that (4.7) provides a better approximation of S_∞ than (4.9) when S_∞ possesses indeed a pole at $\vartheta = 1$. We notice that $\varepsilon_2^{(m-2)}$ as given in (4.7) is identical to the approximation proposed in [3] (formula 4.19).

From the theory of the epsilon algorithm, cf. [6], [12], it follows that if S_∞ is a rational function of ϑ with as denominator a polynomial of degree $r+\nu$, $r = 0,1,\ldots$, which contains a factor $(1-\vartheta)^\nu$, then

$$\varepsilon_{2r}^{(m)} = S_\infty, \quad \text{for } m \geq m_0, \tag{4.10}$$

when (4.5) or (4.6) is used as initial value for $\nu = 1$, or $\nu = 2$ respectively; the constant m_0 depends on the degrees of the numerator and of the denominator of S_∞. This result holds for ϑ smaller as well as larger than the radius of convergence of the series $S_\infty(\vartheta)$, cf. (4.3). Therefore, if the moments of the queue length distribution are rational functions of ρ, it would not be necessary to use the transformation (3.1). However, experience learns that it is still advisable to use the mapping (3.1) in such a case, because the convergence of the series may be slower and the power-series algorithm may be numerically instable when G is too small. The latter seems to occur when some state probabilities possess more singularities than the moments, as functions of ρ. To obtain a good value of G a test run of the power-series algorithm with G = 0 is needed in order to estimate the radius of convergence of the different power-series.

The performance of the modified epsilon algorithm, cf. (4.5), (4.1), is illustrated in table 1 on the basis of an asymmetrical two-queue system with alternating service discipline (i.e. $q_1 = q_2 = 0$). The arrival rates are $\lambda_1 = 0.64$, $\lambda_2 = 0.32$, and the service rates are $\mu_1 = 1$, $\mu_2 = 2$ (hence $\rho = 0.8$). We have chosen G = 2. It should be noted that the rate of convergence of the sequences $\{\varepsilon_{2\kappa}^{(m)}, m = 1,2,\ldots\}$ does not increase monotonously with increasing κ (see the columns with $\kappa=3$ in table 1). This may be caused by pairs of complex conjugate singularities of the mean queue lengths as function of ρ. In general, it is quite unpredictable which sequence produced by the epsilon algorithm will converge most fastly. It may depend on the value of G. When the model is more symmetrical or when the traffic intensity is lower, the performance of the epsilon algorithm will be better than in the case of table 1 (and vice versa). More research is needed to discover how the power-series algorithm can be combined most effectively with the epsilon algorithm or any other algorithm for accelerating the convergence of sequences, cf. [6].

Table 1. Performance of the modified epsilon algorithm.

	$E\{N_1\}$					
m	$\varepsilon_0^{(m)}$	$\varepsilon_2^{(m-2)}$	$\varepsilon_4^{(m-4)}$	$\varepsilon_6^{(m-6)}$	$\varepsilon_8^{(m-8)}$	$\varepsilon_{16}^{(m-16)}$
20	3.2545	3.2702	3.2742	3.2712	3.2721	3.271366
24	3.2639	3.2704	3.2726	3.2711	3.2716	3.273994
28	3.2683	3.2707	3.2720	3.2713	3.2715	3.271606
32	3.2702	3.2710	3.2717	3.2714	3.2716	3.271599
36	3.2711	3.2712	3.2716	3.2715	3.2716	3.271596
40	3.2714	3.2714	3.2716	3.2714	3.2716	3.271595
	$E\{N_2\}$					
m	$\varepsilon_0^{(m)}$	$\varepsilon_2^{(m-2)}$	$\varepsilon_4^{(m-4)}$	$\varepsilon_6^{(m-6)}$	$\varepsilon_8^{(m-8)}$	$\varepsilon_{16}^{(m-16)}$
20	.69101	.65968	.65162	.65762	.65583	.657269
24	.67225	.65924	.65484	.65784	.65682	.652011
28	.66345	.65863	.65605	.65749	.65694	.656788
32	.65953	.65803	.65653	.65720	.65689	.656801
36	.65788	.65753	.65672	.65707	.65685	.656807
40	.65722	.65718	.65679	.65721	.65683	.656810

5. Implementation

The main restriction in applying the power-series algorithm is the required amount of memory space. Therefore, this section is devoted to ideas for an efficient implementation of the power-series algorithm. One way to limit the required amount of memory space is the reduction of the number of coefficients $b(k;\bar{n},h)$ which have to be calculated, cf. (3.2), (3.3), (3.4), by applying algorithms for accelerating the convergence of sequences such as the epsilon algorithm discussed in section 4. Other ways may be found in preventing that a part of the available memory positions remains unused and in reusing the memory positions which are occupied by coefficients $b(k;\bar{n},h)$ which will not be needed anymore in later computations. These topics will be addressed below.

Suppose that the coefficients of the power-series expansions of the state probabilities and the moments of the queue length distribution have to be computed up to the M^{th} power of ϑ for a particular model. This implies that the coefficients $b(k;\bar{n},h)$ must be calculated for $k+n_1+\ldots+n_s = m$, $m = 0,1,\ldots,M$ and $h = 1,\ldots,s$, cf. (3.2), (3.5), (3.6), i.e.

$$s\binom{M+s+1}{s+1} \tag{5.1}$$

of those coefficients are needed. When these coefficients would be stored in rectangular arrays, then

$$s(M+1)^{s+1} \tag{5.2}$$

memory positions were required. Hence, there is a considerable reduction in storage requirement when a two-dimensional array of size (5.1) is used to store the coefficients $b(k;\bar{n},h)$. In order to be able to locate the coefficients the following mapping of the lattice points $(k,\bar{n})$, $k+n_1+\ldots+n_s \leq M$ onto the set of integers $0,1,\ldots,\binom{M+s+1}{s+1}-1$, can be used, cf. [2],

$$C(k;\bar{n}) = \sum_{j=0}^{s} \binom{k + j + \Sigma_{i=1}^{j} n_i}{j+1}. \tag{5.3}$$

The drawback of this procedure is that it costs quite some computation time to determine the locations of the 3s+2 coefficients which are in general involved in each step of the iteration (3.3) by using (5.3) directly. Therefore, we give a more efficient algorithm for determining the locations of these 3s+2 coefficients simultaneously in the appendix.

A further reduction of the storage requirement can be obtained by the following considerations. In many circumstances one is not interested in all the individual state probabilities (3.2), but only in some aggregated measures of performance such as the first and second order moments of the queue length distribution and a few characteristic probabilities. In this case it will be more efficient to store the coefficients of the power-series expansions of this limited number of performance measures in separate arrays. The coefficients $b(k;\bar{n},h)$ can then be deleted from memory as soon as they are not needed anymore in later steps of the iteration (3.3), and we can use the following mapping to locate these coefficients:

$$C_M(k;\bar{n}) = (k;\bar{n}) \bmod D_M. \tag{5.4}$$

Here D_M is the maximal distance which occurs between the value $C(k;\bar{n})$ and any of the values $C(k-1;\bar{n})$, $C(k;\bar{n}-\bar{e}_h)$, $C(k-1;\bar{n}+\bar{e}_h)$, $C(k-2;\bar{n}+\bar{e}_h)$, $h = 1,\dots,s$, cf. (3.3), over all points $(k;\bar{n})$ with $k+n_1+\dots+n_s \le M$. It is readily verified that this implies (see also the appendix) that

$$D_M = \max\{C(k;\bar{n}) - C(k-2;\bar{n}+\bar{e}_s);\ k+n_1+\dots+n_s \le M\}, \tag{5.5}$$

if the coefficients $b(k;\bar{n},h)$, $h = 1,\dots,s$, are computed in order of increasing value of $C(k;\bar{n})$, cf. (5.3). It turns out the maximum in (5.5) is attained at the point $(M;\bar{0})$, so that

$$D_M = \binom{M+s+1}{s+1} - 2\binom{M+s-1}{s+1} + \binom{M+s-2}{s+1} = \binom{M+s}{s} + \binom{M+s-2}{s-1}. \tag{5.6}$$

This approach requires sD_M memory positions to store the coefficients $b(k;\bar{n},h)$. Beside these coefficients also those of the aggregated performance measures have to be stored. But in order to apply the epsilon algorithm to the coefficients of these measures they must be determined also when the modulus operator in (5.4) would not be used. To illustrate the gain which is obtained by applying (5.4), (5.6), we show in table 2 the maximum number M of terms of the power-series (3.5) which can be computed when respectively rectangular arrays, cf. (5.2), the mapping (5.3), cf. (5.1), or the mapping (5.4) are used and when there is storage capacity for 10^6 coefficients $b(k;\bar{n},h)$.

Table 2. Maximum number of terms M at a storage capacity of 10^6 coefficients.

# queues	2	3	4	5	6
rectangular	78	23	11	6	< 5
triangular (5.3)	142	50	28	19	14
with modulus (5.4)	997	123	46	26	18

6. Examples

In this section numerical data for cyclic-service systems which have been obtained with the aid of the power-series algorithm will be presented. The value of G in the mapping (3.1), the number of terms M of the power-series, cf. (3.5), and the number of steps κ in the epsilon algorithm, cf. (4.1), which were needed to obtain these data, depended on various properties of the models. Generally, these quantities increase with increasing traffic intensity, with increasing number of queues, with increasing asymmetry between the parameters of the different queues, and with decreasing value of the Bernoulli parameters q_j, $j = 1,\dots,s$.
Table 3 shows the way in which the expected values and standard deviations of the waiting times depend on the values of the Bernoulli parameters q_1 and q_2, for a two-queue system with $\mu_1 = \mu_2 = 1$ and $\lambda_1 = \lambda_2 = 0.45$ (i.e. $\rho = 0.9$). For comparison, the standard deviation of the waiting time in an M/M/1 system with $\rho = 0.9$, $\mu = 1$ and FIFO service discipline is 9.950.

Table 3. Dependency of the waiting time distributions on the parameters of the Bernoulli schedules.

q_1	q_2	$E\{W_1\}$	$E\{W_2\}$	$\sigma(W_1)$	$\sigma(W_2)$
0	0	9.000	9.000	11.19	11.19
½	½	9.000	9.000	11.55	11.55
1	1	9.000	9.000	12.36	12.36
0	½	14.70	3.302	18.22	3.712
½	1	15.69	2.306	19.14	2.378
0	1	16.36	1.636	19.39	1.809

In table 4 the standard deviation of the waiting time distribution has been listed for symmetrical systems with either exhaustive service ($q_j = 1$, $j = 1,\dots,s$) or 1-limited service ($q_j = 0$, $j = 1,\dots,s$). The mean waiting time follows directly from (3.11) for symmetrical systems, and does not depend on the Bernoulli schedule.

Table 4. Standard deviation of the waiting time for symmetrical systems ($\mu_j = 1$, $j = 1,\dots,s$).

ρ	s=2 exh.	s=2 1-lim.	s=3 exh.	s=3 1-lim.	s=4 exh.	s=4 1-lim.
0.10	0.491	0.490	0.493	0.492	0.494	0.493
0.30	1.071	1.056	1.081	1.070	1.086	1.078
0.50	1.901	1.835	1.925	1.881	1.935	1.906
0.70	3.693	3.460	3.730	3.598	3.743	3.680
0.80	5.876	5.414	5.919	5.680	5.933	5.846
0.90	12.36	11.19	12.41	11.87	12.42	12.31
0.95	25.29	22.67	25.33	24.20	25.35	25.21

Table 5 shows the influence of a relatively heavily loaded queue on the mean waiting times at queues which are four times more lightly loaded, for different service schedules. The parameters of the system are in the case of $s = 3(4)$ queues: $\mu_1 = 1$, $\mu_j = 2$, $j = 2,3(,4)$; $a_1 = 2a_j$, $j = 2,3(,4)$; $q_j = q_2$, $j = 3\ (,4)$; and $\rho = 0.8$. Note that the differences in mean waiting times of the lightly loaded queues are not negligible, although their arrival and service rates are the same.

Table 5. <u>The influence of one relatively heavily loaded queue</u>.

$s = 3$	q_1	q_2	$E\{W_1\}$	$E\{W_2\}$	$E\{W_3\}$		$E\{W\}$
	0	0	4.170	1.644	1.677		2.915
	1	0	1.515	6.936	7.004		4.242
	1	1	2.453	4.869	5.319		3.773
$s = 4$	q_1	q_2	$E\{W_1\}$	$E\{W_2\}$	$E\{W_3\}$	$E\{W_4\}$	$E\{W\}$
	0	0	4.190	1.719	1.745	1.774	2.724
	1	0	1.334	5.499	5.554	5.610	3.866
	1	1	2.344	3.979	4.183	4.460	3.462

Table 6 is concerned with the order in which the server attends the queues. The system consists of 4 queues A, B, C, D, with parameters $a_A = a_B = 0.16$, $a_C = a_D = 0.64$, $\mu_A = \mu_C = 1$, $\mu_B = \mu_D = 4$, $q_j = 0$ for j = A, B, C, D, and $\rho = 0.9$. For each queue the minimal mean waiting time has been underlined in the table. It can be seen that for each queue A, B and D separately it is best to follow after queue C and it is worst to precede queue C, the most heavily loaded queue. This difference is relatively the largest at queue B (4.3%).

Table 6. The effect of the order in which the server attends the queues.

Order	$E\{W_A\}$	$E\{W_B\}$	$E\{W_C\}$	$E\{W_D\}$	$E\{W\}$
A B C D	1.654	1.549	9.547	7.585	7.173
A D C B	1.654	<u>1.485</u>	9.544	7.611	7.176
A C B D	1.670	1.486	<u>9.543</u>	7.597	7.172
A D B C	1.639	1.548	9.547	7.597	7.177
A B D C	<u>1.638</u>	1.531	9.545	7.613	7.181
A C D B	1.671	1.499	9.547	<u>7.583</u>	<u>7.169</u>

7. Comments

The power-series algorithm has been applied in this paper to a single server, multi-queue system with cyclic service discipline, Bernoulli schedules, infinite buffers, Poisson arrival streams, exponential service time distributions and negligible switching times. It can also be applied to several variations and extensions of this model. Service disciplines as random allocation or priority for the longest queue can be treated in the same way as cyclic service. Disciplines as gated service or K-limited service (a fixed number of jobs is served at each visit of the server to a queue) pose some problems, because supplementary variables with rather large value ranges are needed to transform the queue length process into a Markov process. The power-series algorithm can in principle also be applied to models with finite buffers. The main difference with infinite buffer systems is that steady state occurs at any traffic intensity ρ, $\rho > 0$. Therefore, we propose to use the conformal mapping

$$\vartheta = \frac{\rho}{C+\rho} \; (\rho = \frac{C\vartheta}{1-\vartheta}), \; C \geq 0, \tag{7.1}$$

of $[0,\infty)$ onto $[0,1)$, instead of the conformal mapping (3.1). Further, the modification of the epsilon algorithm as in (4.5), (4.6), should not be applied since the moments of the queue length distribution are finite for all values of the traffic intensity when buffers are finite. Application of the power-series algorithm to finite buffer systems is a subject for further research. Note, however, that it may be more efficient to solve the set of balance equations directly when buffer sizes are very small. Exponential distributions in the model may be replaced by phase-type distributions, cf. [3]. This requires the introduction of one supplementary variable for each non-Poissonean arrival process and one supplementary variable when one or more service time distributions are non-exponential. Non-zero switching times can also be incorporated in the model and the algorithm, but they must have phase-type distributions. Then a two-valued variable should be added indicating whether the server is serving a job or moving from one queue to the next. In this case the variable H indicating the position of the server is also defined when all queues are empty. The distribution of H when the system is empty cannot be recursively calculated, because the server may turn an arbitrary number of cycles around the queues when they are all empty. Sets of s linear equations have to be solved to compute the coefficients of the power-series expansions of the probabilities that the system is empty and the server is moving between two queues. The coefficients of all other state probabilities can be calculated in a recursive manner.

Acknowledgement

The author thanks Dr. J. Kapenga for a helpful discussion.

References

[1] Blanc, J.P.C. A Note on waiting times in systems with queues in parallel, J. Appl. Probab. 24 (1987), 540-546.

[2] Blanc, J.P.C. On a numerical method for calculating state probabilities for queueing systems with more than one waiting line, J. Comput. Appl. Math. 20 (1987), 119-125.

[3] Blanc, J.P.C. A numerical study of a coupled processor model, in: Computer Performance and Reliability, eds. G. Iazeolla, P.J. Courtois, O.J. Boxma (North-Holland, Amsterdam, 1988), 289-303.

[4] Boxma, O.J., W.P. Groenendijk. Pseudo convervation laws in cyclic-service systems, J. Appl. Probab. 24 (1987), 949-964.

[5] Boxma, O.J., B.W. Meister. Waiting-time approximations in multi-queue systems with cyclic service, Performance Evaluation 7 (1987), 59-70.

[6] Brezinski, C. Accélération de la Convergence en Analyse Numérique. Lect. Notes in Maths. 584, Springer, Heidelberg, 1977.

[7] Hooghiemstra, G., M. Keane, S. van de Ree. Power-series for stationary distributions of coupled processor models, Report 87-70, Fac. Techn. Maths. & Informatics, Delft Univ. of Technology, 1987.

[8] Kühn, P.J. Multi-queue systems with non-exhaustive cyclic service, Bell. Syst. Techn. J. 58 (1979), 671-698.

[9] Servi, L.D. Average delay approximation of M/G/1 cyclic service queues with Bernoulli schedules, IEEE J. Sel. Areas Comm., SAC-4 (1986), 813-822.

[10] Takagi, H. Analysis of Polling Systems, The MIT Press, Cambridge (Mass.), 1986.

[11] Watson, K.S. Performance evaluation of cyclic service strategies - a survey, in: Performance '84, ed. E. Gelenbe (North-Holland, Amsterdam, 1985), 521-533.

[12] Wynn, P. On the convergence and stability of the epsilon algorithm, SIAM J. Numer. Anal. 3 (1966), 91-122.

Appendix

In section 5 a mapping of the lattice points $(k,\bar{n})$, with $k+n_1+\ldots+n_s \le M$, to the set of integers has been discussed, cf. (5.3). In this appendix we give an efficient procedure for determining the value of this mapping in an integrated way for all lattice points which occur at one step of the power-series algorithm, cf. (3.3). This procedure is based on the following properties of the mapping $C(k;\bar{n})$ that are to verify straightforwardly:

$$C(k;\bar{n}-\bar{e}_j) = C(k;\bar{n}-\bar{e}_{j+1}) - \binom{k+j-1+\Sigma_{i=1}^{j} n_i}{j}, \tag{A.1}$$

$$C(k-1;\bar{n}+\bar{e}_{j+1}) = C(k-1;\bar{n}+\bar{e}_j) - \binom{k+j-1+\Sigma_{i=1}^{j} n_i}{j}, \tag{A.2}$$

for $j = 0,1,\ldots,s$; here and below, both $(k;\bar{n}-\bar{e}_{s+1})$ and $(k-1;\bar{n}+\bar{e}_0)$ stand for $(k;\bar{n})$, while both $(k;\bar{n}-\bar{e}_0)$ and $(k-1;\bar{n}+\bar{e}_{s+1})$ are equivalent to $(k-1;\bar{n})$. Further, the iteration (3.3), (3.4) will proceed along points $(k;\bar{n})$ according to increasing values of $C(k;\bar{n})$. This order will be indicated later, cf. (A.4). The precedessor of the point $(k;\bar{n})$ in this order is denoted by $(k^*,\bar{n}^*)$. The procedure to locate the points which are needed in the iteration step (3.3) then reads:

$C(k;\bar{n}) = C(k^*;\bar{n}^*) + 1;$

For j := 0 to s calculate $v(j) := \binom{k+j-1+\Sigma_{i=1}^{j} n_i}{j}$;

For j := s downto 1 do $C(k;\bar{n}-\bar{e}_j) := C(k;\bar{n}-\bar{e}_{j+1}) - v(j)$;

If $k \ge 1$ then for j := 0 to s do

$\quad C(k-1;\bar{n}+\bar{e}_{j+1}) := C(k-1;\bar{n}+\bar{e}_j) - v(j)$;

If $k \ge 2$ and $G > 0$ then for j := 0 to $s-1$ do

$$C(k-2;\bar{n}+\bar{e}_{j+1}) := C(k-2;\bar{n}+\bar{e}_j) - v(j)\,\frac{k-1+\Sigma_{j=1}^{j} n_i}{k+j-1+\Sigma_{i=1}^{j} n_i}\,. \tag{A.3}$$

Finally, we present a procedure for calculating the coefficients of the power-series expansions of the state probabilities and the moments of the queue length distribution according to (3.3), (3.4) and (3.6) up to the M^{th} power of ϑ, along points with increasing values of $C(k;\bar{n})$.

```
b(0;0̄) := 1;
sum1 := 0;
for m := 1 to M
for n_s := m downto 0
for n_{s-1} := m-n_s downto 0
        ⋮
for n_1 := m-n_s-...-n_2 downto 0 do
[k := m-n_s-...-n_1;
 if k=m then {b(m;0̄) := -sum1; sum1 := 0} else
 {determine the locations of the points needed in (3.3) according
  to (A.3);
  sum2 := 0;
  for h := 1 to s do
  [calculate b(k;n̄,h) according to (3.3);
   sum2 := sum2 + b(k;n̄,h)];
  sum1 := sum1 + sum2;
  for ν := 1 to 2 and j := 1 to s do
   f_ν(m;j) := f_ν(m;j) + n_j^ν × sum2 (cf. (3.6))}].              (A.4)
```

The variables $f_\nu(m;j)$, $\nu = 1,2$, $j = 1,\ldots,s$, $m = 1,\ldots,M$, are initially equal to zero in (A.4). The variabele sum1 in (A.4) is used to determine $b(k;\bar{0})$, $k = 1,\ldots,M$, according to (3.4).

IN 1987 REEDS VERSCHENEN

242 Gerard van den Berg
Nonstationarity in job search theory

243 Annie Cuyt, Brigitte Verdonk
Block-tridiagonal linear systems and branched continued fractions

244 J.C. de Vos, W. Vervaat
Local Times of Bernoulli Walk

245 Arie Kapteyn, Peter Kooreman, Rob Willemse
Some methodological issues in the implementation of subjective poverty definitions

246 J.P.C. Kleijnen, J. Kriens, M.C.H.M. Lafleur, J.H.F. Pardoel
Sampling for Quality Inspection and Correction: AOQL Performance Criteria

247 D.B.J. Schouten
Algemene theorie van de internationale conjuncturele en strukturele afhankelijkheden

248 F.C. Bussemaker, W.H. Haemers, J.J. Seidel, E. Spence
On (v,k,λ) graphs and designs with trivial automorphism group

249 Peter M. Kort
The Influence of a Stochastic Environment on the Firm's Optimal Dynamic Investment Policy

250 R.H.J.M. Gradus
Preliminary version
The reaction of the firm on governmental policy: a game-theoretical approach

251 J.G. de Gooijer, R.M.J. Heuts
Higher order moments of bilinear time series processes with symmetrically distributed errors

252 P.H. Stevers, P.A.M. Versteijne
Evaluatie van marketing-activiteiten

253 H.P.A. Mulders, A.J. van Reeken
DATAAL - een hulpmiddel voor onderhoud van gegevensverzamelingen

254 P. Kooreman, A. Kapteyn
On the identifiability of household production functions with joint products: A comment

255 B. van Riel
Was er een profit-squeeze in de Nederlandse industrie?

256 R.P. Gilles
Economies with coalitional structures and core-like equilibrium concepts

257 P.H.M. Ruys, G. van der Laan
Computation of an industrial equilibrium

258 W.H. Haemers, A.E. Brouwer
Association schemes

259 G.J.M. van den Boom
Some modifications and applications of Rubinstein's perfect equilibrium model of bargaining

260 A.W.A. Boot, A.V. Thakor, G.F. Udell
Competition, Risk Neutrality and Loan Commitments

261 A.W.A. Boot, A.V. Thakor, G.F. Udell
Collateral and Borrower Risk

262 A. Kapteyn, I. Woittiez
Preference Interdependence and Habit Formation in Family Labor Supply

263 B. Bettonvil
A formal description of discrete event dynamic systems including perturbation analysis

264 Sylvester C.W. Eijffinger
A monthly model for the monetary policy in the Netherlands

265 F. van der Ploeg, A.J. de Zeeuw
Conflict over arms accumulation in market and command economies

266 F. van der Ploeg, A.J. de Zeeuw
Perfect equilibrium in a model of competitive arms accumulation

267 Aart de Zeeuw
Inflation and reputation: comment

268 A.J. de Zeeuw, F. van der Ploeg
Difference games and policy evaluation: a conceptual framework

269 Frederick van der Ploeg
Rationing in open economy and dynamic macroeconomics: a survey

270 G. van der Laan and A.J.J. Talman
Computing economic equilibria by variable dimension algorithms: state of the art

271 C.A.J.M. Dirven and A.J.J. Talman
A simplicial algorithm for finding equilibria in economies with linear production technologies

272 Th.E. Nijman and F.C. Palm
Consistent estimation of regression models with incompletely observed exogenous variables

273 Th.E. Nijman and F.C. Palm
Predictive accuracy gain from disaggregate sampling in arima - models

274 Raymond H.J.M. Gradus
The net present value of governmental policy: a possible way to find the Stackelberg solutions

275 Jack P.C. Kleijnen
A DSS for production planning: a case study including simulation and optimization

276 A.M.H. Gerards
A short proof of Tutte's characterization of totally unimodular matrices

277 Th. van de Klundert and F. van der Ploeg
Wage rigidity and capital mobility in an optimizing model of a small open economy

278 Peter M. Kort
The net present value in dynamic models of the firm

279 Th. van de Klundert
A Macroeconomic Two-Country Model with Price-Discriminating Monopolists

280 Arnoud Boot and Anjan V. Thakor
Dynamic equilibrium in a competitive credit market: intertemporal contracting as insurance against rationing

281 Arnoud Boot and Anjan V. Thakor
Appendix: "Dynamic equilibrium in a competitive credit market: intertemporal contracting as insurance against rationing

282 Arnoud Boot, Anjan V. Thakor and Gregory F. Udell
Credible commitments, contract enforcement problems and banks: intermediation as credibility assurance

283 Eduard Ponds
Wage bargaining and business cycles a Goodwin-Nash model

284 Prof.Dr. hab. Stefan Mynarski
The mechanism of restoring equilibrium and stability in polish market

285 P. Meulendijks
An exercise in welfare economics (II)

286 S. Jørgensen, P.M. Kort, G.J.C.Th. van Schijndel
Optimal investment, financing and dividends: a Stackelberg differential game

287 E. Nijssen, W. Reijnders
Privatisering en commercialisering; een oriëntatie ten aanzien van verzelfstandiging

288 C.B. Mulder
Inefficiency of automatically linking unemployment benefits to private sector wage rates

289 M.H.C. Paardekooper
A Quadratically convergent parallel Jacobi process for almost diagonal matrices with distinct eigenvalues

290 Pieter H.M. Ruys
Industries with private and public enterprises

291 J.J.A. Moors & J.C. van Houwelingen
Estimation of linear models with inequality restrictions

292 Arthur van Soest, Peter Kooreman
Vakantiebestemming en -bestedingen

293 Rob Alessie, Raymond Gradus, Bertrand Melenberg
The problem of not observing small expenditures in a consumer expenditure survey

294 F. Boekema, L. Oerlemans, A.J. Hendriks
Kansrijkheid en economische potentie: Top-down en bottom-up analyses

295 Rob Alessie, Bertrand Melenberg, Guglielmo Weber
Consumption, Leisure and Earnings-Related Liquidity Constraints: A Note

296 Arthur van Soest, Peter Kooreman
Estimation of the indirect translog demand system with binding non-negativity constraints

IN 1988 REEDS VERSCHENEN

297 Bert Bettonvil
Factor screening by sequential bifurcation

298 Robert P. Gilles
On perfect competition in an economy with a coalitional structure

299 Willem Selen, Ruud M. Heuts
Capacitated Lot-Size Production Planning in Process Industry

300 J. Kriens, J.Th. van Lieshout
Notes on the Markowitz portfolio selection method

301 Bert Bettonvil, Jack P.C. Kleijnen
Measurement scales and resolution IV designs: a note

302 Theo Nijman, Marno Verbeek
Estimation of time dependent parameters in lineair models using cross sections, panels or both

303 Raymond H.J.M. Gradus
A differential game between government and firms: a non-cooperative approach

304 Leo W.G. Strijbosch, Ronald J.M.M. Does
Comparison of bias-reducing methods for estimating the parameter in dilution series

305 Drs. W.J. Reijnders, Drs. W.F. Verstappen
Strategische bespiegelingen betreffende het Nederlandse kwaliteitsconcept

306 J.P.C. Kleijnen, J. Kriens, H. Timmermans and H. Van den Wildenberg
Regression sampling in statistical auditing

307 Isolde Woittiez, Arie Kapteyn
A Model of Job Choice, Labour Supply and Wages

308 Jack P.C. Kleijnen
Simulation and optimization in production planning: A case study

309 Robert P. Gilles and Pieter H.M. Ruys
Relational constraints in coalition formation

310 Drs. H. Leo Theuns
Determinanten van de vraag naar vakantiereizen: een verkenning van materiële en immateriële factoren

311 Peter M. Kort
Dynamic Firm Behaviour within an Uncertain Environment