Approximate Analysis of Queueing Networks with Markovian Arrival Processes and Phase Type Service Times

Johann Christoph Strelen
Rheinische Friedrich–Wilhelms–Universität Bonn

Abstract

Open queueing networks with Markovian arrival processes and phase type service times are considered. A technique is proposed for the specification of the models. Using such a specification, a Markov chain can be generated automatically. Furthermore, this description can be used immediately for simulation. An approximate disaggregation–aggregation technique for finite state Markov chains is applied. Thus large models with many states can be solved, so large that the state probabilities could neither be stored nor calculated in any computer. Accurate approximations are obtained with different degrees of accuracy, depending on the investigated computational effort.

1 Introduction

Queueing networks are widely used in the stochastic modeling of computer, communication, and manufacturing systems. For the restricted class of separable queueing networks, efficient evaluation algorithms are known but if the models are more general exact analysis is expensive or even impossible due to computational complexity. In order to obtain performance measures like throughput of jobs, utilization and availability of system components, and response times, such general queueing networks are transformed into Markov chains which have typically a very large state space and a sparse transition probability matrix. The measures are calculated from the state probabilities of the Markov chain.

Unfortunately, the number of states grows quickly as the complexity of the model increases. This is why much research was done about the solution of large Markov chains [2, 13, 3]. But the state space is subject to a combinatorial explosion law. Therefore exact methods will never suffice for all models, approximation techniques are needed. We propose the *disaggregation-aggregation (DA) iteration* [15]. This is a general approximate technique for finite state Markov chains with discrete time which is based on aggregation. The calculated state probabilities are in product form; we call them *disaggregation-aggregation product-form (DA*Π*)* solutions. Here, neither the complete state space, nor all

Correspondence to: J. Ch. Strelen, Roemerstr. 164, 53117 Bonn, Germany
June 25, 1997

state probabilities, nor the transition probability matrix are handled explicitly, marginal probabilities are considered instead. Therefore Markov chains with really large state spaces can be analyzed, for example billions of states.

Time and space demands are by orders of magnitude lower than for the exact solution of large Markov chains. The results of DA iteration are usually accurate, and the more aggregates are considered the more accuracy can be obtained, if the aggregation is chosen suitably. We call aggregations with many aggregates *fine*.

It is interesting to note that the steady state probabilities of separable queueing networks are equal to DAΠ solutions calculated according to suitably chosen aggregates. Here our DA method is exact.

In [15] we propose also a description technique which we call *transition classes*. It is defined in the terms system components, component states, state transitions, transition probabilities. This technique helps to structure the design of the Markov models.

In this paper, the paradigm of transition classes is developed further. Some restrictions are introduced which render possible to perform DA iteration automatically for a class of coarse and fine aggregations. Previously, for example in [16], the development of the algorithms for DA iteration was a tedious task which had to be repeated for each aggregation. We use solutions of Markov chains to verify the accuracy of DA approximations. The state spaces of these Markov chains and the transition probabilities are generated automatically from the transition classes. Moreover, simulation results are obtained using a simple general algorithm which acts directly upon the transition classes.

In [17] transition classes and the DA technique are applied to queueing networks with different service disciplines like priorities and polling, Coxian service and interarrival time distributions, blocking, or fork and join.

Queueing networks with repetitive service blocking are discussed more thoroughly in [16].

Here we use transition classes and the DA technique for approximate solutions of queueing networks with phase type (PH) service time distributions and Markovian arrival processes (MAP) according to Neuts [12]. The latter comprises phase type interarrival time distributions, single or multiple arrival processes at a node, and single or batch arrivals.

Some research was done about approximate solutions for open queueing networks with general interarrival time distributions and service time distributions, and with infinite buffers. Typically the first and second moments of the distributions are taken into account, and it is assumed that the arrivals to each node are renewal processes. Kühn [10] proposed a decomposition method with these assumptions. Gelenbe and Mitrani [5] applied diffusion approximation for the analysis of the G/G/1 nodes. Whitt uses approximations of Marshall [11] and of Krämer and Langenbach-Belz [9] which are not always accurate [14]. Haverkort [6, 7] replaces these approximations and applies matrix-geometric and Markovian techniques instead, and he considers finite buffers.

In this paper we consider steady state solutions. In section 2 we describe phase type distributions and the considered queueing networks, in section 3 we state how queueing network models can be transformed into Markov chain models using transition classes, in section 4 DAΠ solutions are discussed, and some numerical examples are given in section 5.

2 Queueing Networks with Markovian Arrival Processes and Phase Type Service Times

In this paper disributions of *phase type* (PH-distributions) are used for interarrival times and service times. According to [12, page 231] a probability distribution $F(\cdot)$ on $[0,\infty)$ is of phase type if it is the distribution of the time until absorption in a finite-state Markov chain with a single absorbing state, that is, there exists a probability vector $(\boldsymbol{\beta}, \beta_\kappa)$ and a generator matrix

$$\begin{bmatrix} \boldsymbol{\tau} & \boldsymbol{\tau}^0 \\ \mathbf{0}^\top & 0 \end{bmatrix}$$

such that

$$F(x) = 1 - \boldsymbol{\beta}^\top e^{\boldsymbol{\tau} x}\mathbf{1}, \quad x \geq 0,$$

where $\boldsymbol{\beta}^\top = (\beta_0, \ldots, \beta_{\kappa-1})$, $\mathbf{1} = (1, \ldots, 1)^\top$, $\mathbf{0} = (0, \ldots, 0)^\top$. The $\kappa \times \kappa$ matrix $\boldsymbol{\tau}$ is nonsingular, has negative diagonal elements and nonnegative off-diagonal elements. The vector $\boldsymbol{\tau}^0$ is nonnegative and satisfies $\boldsymbol{\tau}\mathbf{1} + \boldsymbol{\tau}^0 = \mathbf{0}^\top$.

The PH-distribution $F(\cdot)$ has a point mass β_κ at 0 and a density

$$f(x) = \boldsymbol{\beta}^\top e^{\boldsymbol{\tau} x}\boldsymbol{\tau}^0$$

on $(0,\infty)$.

We will use the triplet $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ to denote the phase type distribution and the according renewal process which is defined as follows.

Upon absorption in the Markov process we instantaneously restart the process by selecting a new initial state independently of the past, using the probability vector $(\boldsymbol{\beta}, \beta_\kappa)$. This resetting is repeated indefinitely. It is obvious that the epochs of successive visits to the "instantaneous" state κ form a renewal process with the underlying life time distribution $F(\cdot)$.

If the PH-distribution is an interarrival time distribution this is exactly the process of interarrival times. If it is a service time distribution the service process is slightly different: If upon completion of a service the waiting room is not empty, that means if there is work to do, the Markov process is instantaneously restarted. If there is no job the restart is delayed until the next job arrives.

In the sequel we will use $\beta_\kappa = 0$. That means that the distribution is continuous, there is no point mass at 0.

Let us consider examples. A mixed Erlang distribution

$$F(x) = \sum_{k=1}^{\kappa} \alpha_k E_k(\lambda; x), \quad \kappa \geq 1,$$

is of phase type ($E_k(\lambda; x)$ denotes an Erlang distribution with parameters k and λ). Here $\boldsymbol{\beta}^\top = (\alpha_\kappa, \ldots, \alpha_1)$, $\beta_\kappa = 0$, and

$$\boldsymbol{\tau} = \begin{bmatrix} -\lambda & \lambda & & & \\ & \ddots & & & \\ & & & -\lambda & \lambda \\ & & & & -\lambda \end{bmatrix}.$$

The same but $\boldsymbol{\beta}^\top = (1, 0, \ldots, 0)$ holds for a pure Erlang-κ distribution. Clearly this is an exponential distribution if $\kappa = 1$ holds.

A hyperexponential distribution

$$F(x) = \sum_{k=0}^{\kappa-1} \beta_k (1 - e^{-\lambda_k x}), \quad \kappa \geq 2,$$

is of phase type and may be represented by $\boldsymbol{\beta}^\top = (\beta_0, \ldots, \beta_{\kappa-1})$, $\beta_\kappa = 0$, and $\boldsymbol{\tau} = -\operatorname{diag}(\lambda_0, \ldots, \lambda_{\kappa-1})$.

If $\kappa = 2$, $C^2 \geq 1$, and $\mathrm{E}[X] > 0$ is given, and

$$\begin{aligned} \beta_0 &= \frac{1}{2} - \frac{1}{2}\sqrt{\frac{C^2-1}{C^2+1}}, \quad \beta_1 = 1 - \beta_0 \\ \lambda_0 &= 2\beta_0/\mathrm{E}[X], \quad \lambda_1 = 2\beta_1/\mathrm{E}[X] \end{aligned}$$

hold, the hyperexponential distribution has the expectation $\mathrm{E}[X]$ and the squared coefficient of varation C^2.

Here we did not use the third degree of freedom: The third moment can also be taken into account [1, page 151].

Now we consider bursty arrivals using a hyperexponential distribution. To this end we predefine $\bar{B}$, the mean number of arrivals within a burst, λ, the overall rate of arrivals, and $V = t_\mathrm{P}/t_\mathrm{B}$, the ratio of the mean length of an epoch with no arrivals, t_P, and the mean length of an epoch with burst arrivals, t_B, restricted to $\bar{B} - 1 > 1/V$.

If the coefficients of a hyperexponential distribution with $\kappa = 2$ are according to

$$\begin{aligned} \beta_0 &= 1/\bar{B}, \quad \beta_1 = 1 - \beta_0, \\ \lambda_1 &= (1+V)\beta_1\lambda, \\ \lambda_0 &= \frac{\lambda_1}{V\beta_1/\beta_0 - 1}, \end{aligned}$$

and $\tau_{0,1} = \lambda_0$, the measures $\bar{B}$, λ, and V are satisfied by the distribution. The number of jobs in a burst has a geometric distribution.

Such a process may suitably model arrivals which originate from splitting up messages into packets or cells. Another application could be as follows. A lot of articles is to be treated on a machine, but the machine must be set before. Here the service time is modeled suitably by such a bursty type distribution.

We consider open queueing networks with phase type interarrival times and service times. Closed or mixed networks may be analyzed quite similarly. $\mathcal{K}' \subset I\!N^+$ denotes the set of nodes. Each node $k \in \mathcal{K}'$ consists of some system components.

Component k is the waiting room. It has the component state $z_k \in [0 : \nu_k]$ which counts the number of jobs in the node ($[a : b]$ denotes the set $\{i \in Z\!\!\!Z \mid a \leq i \leq b\}$). The buffer capacity of the node is ν_k. If a job tries to enter the node and there is no buffer space available, blocking occurs and a blocking policy is applied, or the job is lost.

Component $k + 1$ models the service process which is according to a phase type distribution. The state z_{k+1} of this component indicates the phase of this distribution.

The components $k - 1$, $k - 2$, ... are optional. They model arrival processes. z_{k-i} indicates the phase which the i-th arrival process is in. Obviously a node may have multiple arrival processes.

To each arrival process belongs a phase type distribution which models the interarrival times. At the end of an interarrival time a single arrival occurs or a bulk of a arrivals with probability γ_a, $a \in [1:\bar{a}]$.

As usual, a job whose service is finished in node k, leaves the network (destination 0), or tries to go to node i. The selection of the destination i is done according to transition probabilities $t_{k,i},\ k \in \mathcal{K}',\ i \in \mathcal{K}' \cup \{0\}$.

3 Markov Chains for Queueing Networks

Irreducible aperiodic Markov chains $(\mathbf{Z}(t),\ t \in I\!N)$, discrete and homogeneous in time, with finite state space $\mathcal{Z}^T,\ |\mathcal{Z}^T| = n \in I\!N$, are considered. They stem from stochastic models of computer, communication, or manufacturing systems. These systems consist of interacting subsystems, their *(system) components*. At every moment, each component is in one of its possible *component states*. This is why we denote the whole system states by *state tuples* $\mathbf{z} = (z_1, \ldots, z_K)$, where z_k indicates the state of component k. The states z_k of component k belong to the component state space $\mathcal{S}_k$ for all components $k = 1, \ldots, K$. Quite often we use $\mathcal{S}_k = [0 : N_k]$.

Sometimes virtual system components are also useful, for example a "component" which indicates the number of jobs in an open queueing network.

Aggregates are sets of system states. An aggregate can be defined by the state of a certain system component, for example the aggregate $\mathcal{Z}_{k,j}$ is the set of all system states where the component k is in state j, irrespective of the states of the other components:

$$\mathcal{Z}_{k,j} = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k = j\},\ j \in \mathcal{S}_k,\ k = 1, \ldots, K.$$

We call these aggregates *canonical*. For k fixed, all canonical aggregates $\mathcal{Z}_{k,0}, \ldots, \mathcal{Z}_{k,N_k}$ form a *canonical* partition $\mathcal{Z}_k$ of the state space. All the partitions $\mathcal{Z}_k,\ k = 1, \ldots, K$, constitute the *canonical aggregation*.

In general, a partition $\mathcal{Z}_c$ consists in aggregates $\mathcal{Z}_{c,j},\ j \in \mathcal{N}_c$. A class $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_C\},\ C \in I\!N$, of partitions is an *aggregation*. For canonical aggregation, there are $C = K$ partitions, and $\mathcal{N}_c = [0 : N_c],\ c = 1, \ldots, C$, holds.

A partition can be defined by two or more system components $k, k', \ldots, \hat{k}$, for example the partition with the aggregates

$$\begin{aligned}\mathcal{Z}_{(k,k'),(j,j')} = \{&\mathbf{z} \in \mathcal{Z}^T \mid z_k = j,\ z_{k'} = j'\},\\ &j \in \mathcal{S}_k,\ j' \in \mathcal{S}_{k'},\ k \neq k',\ k, k' \in [1:K],\end{aligned} \tag{1}$$

combines the components k and k'. If more than one component define aggregates, the partition and the aggregation are termed *fine*. The sets of components which are used to define a partition $\mathcal{Z}_c$ are denoted $\mathcal{H}_c = \{k, k', \ldots, \hat{k}\}$, for example $\mathcal{H}_c = \{c\},\ c = 1, \ldots, K$, for the canonical aggregation.

$p^{(t)}(\mathbf{z}) \stackrel{\triangle}{=} \mathrm{P}\{\mathbf{Z}(t) = \mathbf{z}\},\ \mathbf{z} \in \mathcal{Z}^T$, denotes the state probabilities of the Markov chain at time t. These probabilities are collected in the n-vector $\mathbf{p}^{(t)}$. $p(\mathbf{x}, \mathbf{y}),\ \mathbf{x}, \mathbf{y} \in \mathcal{Z}^T$, are the state transition probabilities. They are the elements of the transition probability matrix $\mathbf{P}$.

Sometimes, if we are not interested in a particular time, or if steady state probabilities are considered, we omit the specification (t).

Each aggregate $\mathcal{Z}_{c,j}$ defines a *macro probability* or *aggregate probability* $\mathrm{P}\{\mathbf{Z} \in \mathcal{Z}_{c,j}\}$:

$$r_c(j) = \sum_{\mathbf{z} \in \mathcal{Z}_{c,j}} p(\mathbf{z}),\ j \in \mathcal{N}_c,\ c = 1, \ldots, C. \tag{2}$$

We collect the aggregate probabilities of a partition $\mathcal{Z}_c$ into tuples $\mathbf{r}_c$ and, in turn, these into $\mathbf{R} \stackrel{\triangle}{=} (\mathbf{r}_1, \ldots, \mathbf{r}_C)$.

By (2), the *aggregation function*

$$\mathcal{A} : [0,1]^n \to [0,1]^{|\mathcal{N}_1|} \times \ldots \times [0,1]^{|\mathcal{N}_C|},\ \mathbf{p} \mapsto \mathbf{R},$$

is defined. It calculates the macro probabilities for given state probabilities.

The states of the system components are described by the random variables $Z_k : \mathcal{Z}^T \to \mathcal{S}_k,\ \mathbf{z} \mapsto z_k$. The marginal distribution $\mathbf{r}_k$, calculated according to the canonical aggregation, is the probability distribution of Z_k. The state of the whole system is given by the random vector $\mathbf{Z} = (Z_1, \ldots, Z_K)$ with the probability distribution $\mathrm{P}\{\mathbf{Z} = \mathbf{z}\} = p(\mathbf{z}),\ \mathbf{z} \in \mathcal{Z}^T$.

Example. In a queueing network with blocking, Poisson arrivals, if any, and $\cdot$/M/1/ν_k/FCFS nodes, the system components are the nodes. Their states are the numbers z_k of jobs in them, $z_k \leq \nu_k,\ k = 1, \ldots, K$. Therefore, a system state is given by the $K-$tuple $\mathbf{z} = (z_1, \ldots, z_K)$. In an open queueing network, $\mathcal{Z}^T$ equals $\mathcal{S}_1 \times \ldots \times \mathcal{S}_K$ where $\mathcal{S}_k = [0 : \nu_k]$ is the component state space, and in a closed queueing network

$$\mathcal{Z}^T = \{\mathbf{z} \in \mathcal{S}_1 \times \ldots \times \mathcal{S}_K \mid z_1 + \ldots + z_K = N\}, \tag{3}$$

where $\mathcal{S}_k = [\tau_k : \nu_k],\ \tau_k = \max\{0, N - \sum_{j \neq k} \nu_j\}$: A state $\tau_k > 0$ in node k occurs if in no other node space is left. □

We assume that the Markov chain transitions from state $\mathbf{y}$ into state $\mathbf{z},\ \mathbf{y} \neq \mathbf{z},\ \mathbf{y}, \mathbf{z} \in \mathcal{Z}^T$, can be classified as follows: Each state transition belongs to one and only one of the *transition classes*, which are triplets $(\mathcal{U}_\nu, u_\nu, \alpha_\nu),\ \nu = 1, \ldots, \bar{\nu}$. The introduction of these transition classes helps to structure the design of the Markov models, renders possible to generate the Markov chain automatically, and turns out to be very useful for the DA technique. Moreover, it is not difficult to design an universal algorithm for the simulation of Markov chains which are defined by transition classes. Such an algorithm needs not to enumerate all system states, nor the matrix of transition probabilities is required, only the component states and the transition classes are considered.

A transition $\mathbf{y}$ into $\mathbf{z}$ of transition class ν may occur only if $\mathbf{y} \in \mathcal{U}_\nu$, where $\mathcal{U}_\nu \subset \mathcal{Z}^T$ denotes the *source states set* of this class. The *destination state function* $u_\nu : \mathcal{Z}^T \to \mathcal{Z}^T$ defines the new states, i.e. a transition of class ν leads from state $\mathbf{y} \in \mathcal{U}_\nu$ into state $u_\nu(\mathbf{y})$. The tuple u_ν consists of functions $u_{\nu,k},\ u_\nu(\mathbf{y}) = \big(u_{\nu,1}(\mathbf{y}), \ldots, u_{\nu,K}(\mathbf{y})\big)$. Transitions of class ν occur with probability $\alpha_\nu(\mathbf{y})$. For DA iteration some restrictions on the transition classes will be required later on.

Example. We consider a queueing network with repetitive service blocking. It consists of $K \in I\!N$ nodes. In node $k,\ k = 1, \ldots, K$, there is a single exponential server, service rate μ_k. Buffer space is available for no more than ν_k jobs. If the model is open, jobs arrive according to a Poisson process with parameter λ_k but only if buffer space is available. When the service of a job in node k is finished the job tries to go to node l with probability $t_{k,l},\ l \in [1 : K]$. If there is no buffer space available, the job remains

in node k and instantaneously receives another service (*repetitive-service with random destination policy*). In an open network, a job which is completely served in node k leaves the network with probability $t_{k,0}$. In a closed network, there are $N \in I\!N$ jobs, and we assume without loss of generality $\nu_k \leq N,\ k = 1, \ldots, K$.

The system may change its state due to an arrival at node k from outside. The source state set of this transition class is $\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k < \nu_k\}$. That means, there is space left in node k for an arriving job. Only the state of node k changes, the other remain unchanged. Hence the destination state function is given by $u_{\nu,k}(\mathbf{z}) = z_k + 1$, identity otherwise: $u_{\nu,i}(\mathbf{z}) = z_i$ if $i \neq k$. The transition probability is $\alpha_\nu = \lambda_k \Delta$.

Δ is the randomization constant: Initially, the Markov chain is continuous in time. Let $\mathbf{Q}$ denote its generator matrix. We consider the discrete time Markov chain with the transition probability matrix $\mathbf{P} = \mathbf{Q}\Delta + \mathbf{I}$ instead, where $\Delta = 0.99/\max_x |q_{x,x}|$. It has the same steady state probabilities as the continuous time Markov chain. By the factor 0.99, numerical problems are avoided, which may occur due to rounding errors, and the discrete time Markov chain is not periodic.

Other state transitions take place when a job of node k leaves the network. The according transition class ν is defined as follows. The source states set is

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0\}, \tag{4}$$

the destination state function $u_{\nu,k}(\mathbf{z}) = z_k - 1$, identity otherwise, and the transition probability is $\alpha_\nu = \mu_k t_{k,0} \Delta$.

The third kind of state transitions occur when a job leaves node k for node i, $i \neq k$. In node k, there must be at least one job, and in node i buffer space must be available. Therefore, the source states set is

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0,\ z_i < \nu_i\}. \tag{5}$$

Only the states of the nodes k and i change, $u_{\nu,k}(\mathbf{z}) = z_k - 1,\ u_{\nu,i}(\mathbf{z}) = z_i + 1$, and the transition probability is $\alpha_\nu = \mu_k t_{k,i} \Delta$. □

Applying transition classes, the state transition probabilities of a Markov chain can be expressed as follows:

$$\begin{aligned} p(\mathbf{y}, \mathbf{z}) &= \sum_{\nu=1}^{\bar{\nu}} I(\mathbf{y} \in \mathcal{U}_\nu) I\Big(u_\nu(\mathbf{y}) = \mathbf{z}\Big) \alpha_\nu(\mathbf{y}), \\ &\quad \mathbf{y} \neq \mathbf{z},\ \ \mathbf{y}, \mathbf{z} \in \mathcal{Z}^T, \\ p(\mathbf{y}, \mathbf{y}) &= 1 - \sum_{\nu=1}^{\bar{\nu}} I(\mathbf{y} \in \mathcal{U}_\nu) \alpha_\nu(\mathbf{y}),\ \mathbf{y} \in \mathcal{Z}^T, \end{aligned} \tag{6}$$

where $I : \{\text{true, false}\} \to \{0, 1\}$ denotes the indicator function.

It is not difficult to design an algorithm which enumerates the system states by application of the source states sets and the destination state functions. Using such an algorithm and the transition probabilities (6), the Markov chain can be generated automatically. We used such generated Markov chains for many examples.

Similarly a model which is described by means of transition classes can be immediately simulated. In section 4 an example is given.

Now we present the transition classes for the loss networks with MAP arrivals and phase type service times which we described in section 1.

The service process $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ of a node k may change from phase $e \in [0 : \kappa - 1]$, its component state, into the new phase $j \in [0 : \kappa - 1]$ if $\tau_{e,j} > 0$. According to the definition of phase type distributions the service is not finished by this transition. There is such a transition class ν for each service process, each old phase e, and each new phase j if $\tau_{e,j} > 0$.

These transitions occur with rate $\alpha_\nu = \tau_{e,j}$ but only if actually a job is served, i.e. if the system is in one of the states

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0\}.$$

The new phase is

$$u_{\nu,k+1}(\mathbf{z}) = j,$$

the other system components remain unchanged,

$$u_{\nu,i}(\mathbf{z}) = z_i \text{ if } i \neq k.$$

A service in node k may be finished if the service process $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ is in phase $e \in [0 : \kappa - 1]$ where $\tau_{e,\kappa} > 0$. The new phase may be $j \in [0 : \kappa - 1]$ if $\beta_j > 0$. The finished job may go to node $i \in \mathcal{K}'$ if $t_{k,i} > 0$. There is such a transition class ν for each pair (k, i) of nodes if $t_{k,i} > 0$, each old phase e if $\tau_{e,\kappa} > 0$, and each new phase $j \in [0 : \kappa - 1]$ if $\beta_j > 0$.

These transitions occur with rate

$$\alpha_\nu = \tau_{e,\kappa} \beta_j t_{k,i},$$

but only if there is actually a job in node k, if the service process is in phase e, and if buffer space is left in node i:

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0, z_{k+1} = e, z_i < \nu_i\}.$$

The job leaves node k,

$$u_{\nu,k}(\mathbf{z}) = z_k - 1,$$

goes to node i,

$$u_{\nu,i}(\mathbf{z}) = z_i + 1,$$

and the new phase is j,

$$u_{\nu,k+1}(\mathbf{z}) = j.$$

Similar transition classes account for the case where in the destination node i no buffer space is available. The job is lost, node i remains unchanged. Again there is such a transition class ν for each pair (k, i) of nodes if $t_{k,i} > 0$, each old phase e if $\tau_{e,\kappa} > 0$, and each new phase $j \in [0 : \kappa - 1]$ if $\beta_j > 0$.

These transitions occur with rate

$$\alpha_\nu = \tau_{e,\kappa} \beta_j t_{k,i},$$

but only if there is actually a job in node k, if the service process is in phase e, and if no buffer space is left in node i:

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0, z_{k+1} = e, z_i = \nu_i\}.$$

The job leaves node k,

$$u_{\nu,k}(\mathbf{z}) = z_k - 1,$$

and the new phase is j,

$$u_{\nu,k+1}(\mathbf{z}) = j.$$

A service in node k may be finished if the service process $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ is in phase $e \in [0 : \kappa - 1]$ where $\tau_{e,\kappa} > 0$. The new phase may be $j \in [0 : \kappa - 1]$ if $\beta_j > 0$. The finished job may leave the network if $t_{k,0} > 0$. There is such a transition class ν for each node k if $t_{k,0} > 0$, each old phase e if $\tau_{e,\kappa} > 0$, and each new phase $j \in [0 : \kappa - 1]$ if $\beta_j > 0$.

These transitions occur with rate

$$\alpha_\nu = \tau_{e,\kappa} \beta_j t_{k,0},$$

but only if there is actually a job in node k and if the service process is in phase e:

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k > 0, z_{k+1} = e\}.$$

The job leaves node k,

$$u_{\nu,k}(\mathbf{z}) = z_k - 1,$$

and the new phase is j,

$$u_{\nu,k+1}(\mathbf{z}) = j.$$

An arrival process $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ may change from phase $e \in [0 : \kappa - 1]$, which is indicated by its component state, $z_k = e$, into the new phase $j \in [0 : \kappa - 1]$ if $\tau_{e,j} > 0$. According to the definition of phase type distributions no arrival occurs. There is such a transition class ν for each arrival process and each new phase j if there exists an arrival process state $z_k = e$ with $\tau_{e,j} > 0$.

These transitions occur with rate $\alpha_\nu = \tau_{z_k,j}$ but only if $\tau_{z_k,j} > 0$:

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid \tau_{z_k,j} > 0\}.$$

The new phase is

$$u_{\nu,k}(\mathbf{z}) = j,$$

the other system components remain unchanged,

$$u_{\nu,i}(\mathbf{z}) = z_i \text{ if } i \neq k.$$

An arrival may take place at node k' due to one of its arrival processes $(\kappa, \boldsymbol{\tau}, \boldsymbol{\beta})$ where this arrival process changes its state z_k from phase $e \in [0 : \kappa - 1]$ into the new phase $j \in [0 : \kappa - 1]$ via the absorbing state κ if $\tau_{e,\kappa} > 0$ and $\beta_j > 0$. First we restrict ourselves to arrivals of single jobs, no batches.

There is such a transition class ν for each arrival process, each old phase e if $\tau_{e,\kappa} > 0$, and each new phase j if $\beta_j > 0$.

These transitions occur with rate $\alpha_\nu = \tau_{e,j}\beta_j$ but only if the arrival processes is in state e,

$$\mathcal{U}_\nu = \{\mathbf{z} \in \mathcal{Z}^T \mid z_k = e\}.$$

The arriving job enters waiting room k' if buffer space is left, or is lost otherwise,

$$u_{\nu,k}(\mathbf{z}) = \begin{cases} z_{k'} + 1 & \text{if } z_{k'} < \nu_{k'}, \\ \nu_{k'} & \text{otherwise.} \end{cases}$$

The new phase is

$$u_{\nu,k}(\mathbf{z}) = j,$$

the other system components remain unchanged,

$$u_{\nu,i}(\mathbf{z}) = z_i \text{ if } i \neq k.$$

If we allow for batch arrivals, a jobs arrive upon an arrival instant with probability γ_a, $a \in [1 : \bar{a}]$. Under these circumstances there is such a transition class ν for each arrival process, each old phase e if $\tau_{e,\kappa} > 0$, each new phase j if $\beta_j > 0$, and for each batch size a if $\gamma_a > 0$. Here the transition rate is $\tau_{e,j}\beta_j\gamma_a$, and the component k of the destination state function is

$$u_{\nu,k}(\mathbf{z}) = \begin{cases} z_{k'} + a & \text{if } z_{k'} + a \leq \nu_{k'}, \\ \nu_{k'} & \text{otherwise.} \end{cases}$$

4 Disaggregation–Aggregation Iteration

In this section we survey the DA method, details are given in [15].

Our proposed solution technique needs *disaggregation*: Given positive macro probabilities $\mathbf{R}$, a distribution $\boldsymbol{\pi}$ over $\mathcal{Z}^T$ is defined as follows where $\boldsymbol{\pi}$ is a n-vector with the probabilities $\pi(\mathbf{z})$, $\mathbf{z} \in \mathcal{Z}^T$. First, the distribution satisfies the macro probabilities, $\mathbf{R} = \mathcal{A}(\boldsymbol{\pi})$, and secondly, $\boldsymbol{\pi}$ maximizes the entropy. According to the *product-form theorem*, see [8, theorem 13.2.1, page 409], this maximum entropy distribution (MED) $\boldsymbol{\pi}$ is uniquely determined and has a product form,

$$\pi(\mathbf{z}) = \rho_{1,z_1} \cdot \ldots \cdot \rho_{C,z_C}/G, \tag{7}$$

where the $\rho_{c,z_c} > 0$ are some real numbers, and G is a normalizing constant. In fact, the MED is completely determined by all *MED parameters* $\rho_{c,j}$, $\boldsymbol{\rho}$ for short.

By (7) and (2), the *disaggregation function* is defined:

$$\mathcal{D} : [0,1]^{|\mathcal{N}_1|} \times \ldots \times [0,1]^{|\mathcal{N}_C|} \to [0,1]^n, \ \mathbf{R} \mapsto \boldsymbol{\pi}.$$

It calculates the MED over the state space for given aggregate probabilities.

Instead of calculating the steady state probabilities $\mathbf{q}$ of a Markov chain we determine an approximation $\mathbf{p} = \mathcal{D}(\mathbf{R})$ where $\mathbf{R} = \mathcal{T}(\mathbf{R})$ holds. Here the function $\mathcal{T}$ is defined as follows:

$$\begin{aligned} \mathcal{T} \quad &: \quad [0,1]^{|\mathcal{N}_1|} \times \ldots \times [0,1]^{|\mathcal{N}_C|} \to \\ &\qquad [0,1]^{|\mathcal{N}_1|} \times \ldots \times [0,1]^{|\mathcal{N}_C|}, \\ \mathbf{R} \quad &\mapsto \quad \mathcal{A}\big(\mathcal{D}(\mathbf{R})\mathbf{P}\big). \end{aligned}$$

This approximation $\mathbf{p}$ and its marginal probabilities $\mathbf{R}$ are referred to as *steady state disaggregation-aggregation product-form (DAΠ) solution.* In fact, only the marginal

probabilities $\mathbf{R}$ and the MED parameters $\boldsymbol{\rho}$ and G must be calculated. This is practicable even if the state space is large, so large that neither all state probabilities nor the transition probabilities could be handled.

We calculate the DAΠ solutions using *disaggregation-aggregation (DA) iteration*,

$$\mathbf{R}^{(t+1)} = \mathcal{T}(\mathbf{R}^{(t)}),\ t = 1, 2, \ldots \tag{8}$$

until $\max_{c,j} |r_c^{(t+1)}(j) - r_c^{(t)}(j)|$ is sufficiently small, beginning with feasible marginal probabilities $\mathbf{R}^{(1)}$.

This DA iteration resembles vector iteration, $\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)}\mathbf{P},\ \ t = 1, 2, \ldots$, beginning with any distribution $\mathbf{q}^{(1)}$. $\mathbf{q}^{(t)}$ are the time-dependent state probabilities and $\mathcal{A}(\mathbf{q}^{(t)})$ the time-dependent aggregate probabilities. $\mathbf{q}^{(t)}$ converges to the steady state probability vector and $\mathcal{A}(\mathbf{q}^{(t)})$ to the steady state aggregate probabilities. Accordingly, when DA iteration $\mathbf{R}^{(t+1)} = \mathcal{T}(\mathbf{R}^{(t)}),\ t = 1, 2, \ldots$, is performed, $\mathbf{R}^{(t)}$ are approximations of the time-dependent aggregate probabilities and $\mathcal{D}(\mathbf{R}^{(t)})$ approximations of the time-dependent state probabilities. If DA iteration is done for a sufficient number of iterations, $\mathbf{R}^{(t)}$ is an approximation of the steady state aggregate probabilities, and $\mathcal{D}(\mathbf{R}^{(t)})$ an approximation of the steady state probabilities.

The function $\mathcal{T}$ is evaluated with the *aggregate transition equations*

$$r_c^{(t+1)}(l) = \sum_{j \in \mathcal{N}_c} \pi_c(j, l) r_c^{(t)}(j),\ l \in \mathcal{N}_c,\ c = 1, \ldots, C, \tag{9}$$

using the *aggregate transition probabilities*

$$\begin{aligned} \pi_c(j, l) \ &\triangleq\ \mathrm{P}\{\mathbf{Z}_c^{(t+1)} \in \mathcal{Z}_{c,l} \mid \mathbf{Z}_c^{(t)} \in \mathcal{Z}_{c,j}\}, \\ &\quad\ j, l \in \mathcal{N}_c,\ c = 1, \ldots, C. \end{aligned} \tag{10}$$

For canonical aggregation these are efficiently calculated with

$$\pi_k(j, l) = \frac{1}{r_k(j)} \sum_{\nu=1}^{\bar{\nu}} \begin{cases} \alpha_\nu \mathrm{P}\{\mathbf{Z} \in \mathcal{U}_\nu \cap \mathcal{Z}^T \cap \mathcal{X}\} \\ \quad \text{if } j \in \mathcal{U}_{\nu,k},\ u_{\nu,k} = l, \\ 0 \text{ otherwise}, \end{cases} \tag{11}$$

where $\mathcal{X} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_{k-1} \times \{j\} \times \mathcal{S}_{k+1} \times \ldots \times \mathcal{S}_K$, and similarly if a partition $\mathcal{Z}_c$ is defined by the states of more system components. For example, if partition $\mathcal{Z}_c$ is defined by the states of the components k and k' the aggregate transition probabilities are

$$\pi_c(j, j', l, l') = \frac{1}{r_c(j, j')} \sum_{\nu=1}^{\bar{\nu}} \begin{cases} \alpha_\nu \mathrm{P}\{\mathbf{Z} \in \mathcal{U}_\nu \cap \mathcal{Z}^T \cap \mathcal{X}\} \\ \quad \text{if } j \in \mathcal{U}_{\nu,k},\ j' \in \mathcal{U}_{\nu,k'},\ u_{\nu,k} = l,\ u_{\nu,k'} = l', \\ 0 \text{ otherwise}, \end{cases}$$

where $\mathcal{X} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_{k-1} \times \{j\} \times \mathcal{S}_{k+1} \times \ldots \times \mathcal{S}_{k'-1} \times \{j'\} \times \mathcal{S}_{k'+1} \times \ldots \times \mathcal{S}_K$.

Here we must observe some restrictions on the transition classes. Each source states set must be a cartesian product, $\mathcal{U}_\nu = \mathcal{U}_{\nu,1} \times \ldots \times \mathcal{U}_{\nu,K}$ or the union of not too many such products. Under these circumstances the probabilities in (11) can be evaluated efficiently. Furthermore, broadly speaking, if there is a partition $\mathcal{Z}_c$ which is defined by the system components $h, h', \ldots, \hat{h} \in \mathcal{H}_c$, and if the transition class ν changes the state z_k of the

component k where $k \in \mathcal{H}_c$, then the transition probability $\alpha(\mathbf{z})$ and the destination state function $u_\nu(\mathbf{z})$ must not depend on component states z_i, $i \notin \mathcal{H}_c$.

More precisely, let $\mathcal{M}_\nu \in [1:K]$ denote the set of system components whose states are modified by transitions of class ν, $\mathcal{V}_{\nu,k} \subseteq [1:K]$ the set of system components whose states are used for the calculation of the destination state function $u_{\nu,k}(\mathbf{z})$, except k, i.e. $k \notin \mathcal{V}_{\nu,k}$, $\mathcal{V}_\nu = \bigcup_{i \in \mathcal{M}_\nu} \mathcal{V}_{\nu,k}$, and $\mathcal{W}_\nu \subseteq [1:K]$ the set of system components whose states are used for the calculation of the transition probability $\alpha_\nu(\mathbf{z})$.

The aggregate transition probabilities (11) can be calculated if the following *independence conditions on transition classes*

$$i \in \mathcal{H}_c \cap \mathcal{M}_\nu \Rightarrow \mathcal{V}_{\nu,i} \subseteq \mathcal{H}_c, \tag{12}$$

and

$$\mathcal{H}_c \cap \mathcal{M}_\nu \neq \emptyset \Rightarrow \mathcal{W}_\nu \subseteq \mathcal{H}_c \tag{13}$$

are fulfilled for all partitions $c \in \mathcal{C}$ and all transition classes $\nu = 1, \ldots, \bar{\nu}$.

The problem is as follows. If the aggregate transitions (9) are considered for $c \in \mathcal{C}$, only the states z_i, $i \in \mathcal{H}_c$, are available. Only if (12) is fulfilled, the new states can be calculated, and only if (13) is fulfilled, the transition probabilities $\alpha_\nu(\mathbf{z})$ can be determined.

If (12) is violated for i, or if (13) is violated with $i \in \mathcal{W}_\nu$, $i \notin \mathcal{H}_c$, this difficulty can be eliminated as follows. Split up the transition class ν into new classes, one for each $z_i \in \mathcal{S}_i$. In each of these new transition classes z_i is fixed, the destination state functions and the transition probabilities do not depend on z_i. Or alternatively, if only (12) is violated for i, split up the transition class ν into new classes, one for each new value $u_{\nu,i}(\mathbf{z})$.

For the calculation of the probabilities in (11), the MED parameters $\boldsymbol{\rho}$ are needed. In general, the nonlinear equations (2) must be solved in order to calculate all MED parameters $\rho_{k,j}$ of the MED $\boldsymbol{\pi}$, given the marginal probabilities $\mathbf{R}$. This is easily accomplished in some special cases. For example, if the aggregation is canonical and if the state space is *rectangular*, i.e. if $\mathcal{Z}^T$ is a cartesian product, $\mathcal{Z}^T = \mathcal{S}_1 \times \ldots \times \mathcal{S}_K$, the MED is given by $\pi(\mathbf{z}) = r_1(z_1) \cdot \ldots \cdot r_K(z_K)$, $\mathbf{z} \in \mathcal{Z}^T$. If such a simple disaggregation is not available, an approximation is sufficient which becomes more and more accurate during the DA iteration:

$$\rho_{c,j}^{(t+1)} = 0.5 \left(1 + r_c^{(t+1)}(j)/r_c^{(t)}(j)\right) \rho_{c,j}^{(t)}.$$

Unfortunately, we have not yet found error estimates or a proof of convergence of the iteration. Also other authors who developed fixed point techniques observed that proofs of existence of a solution, of the uniqueness, and of convergence are difficult to state, as well as bounds for the accuracy, for example Haverkort [7]. But in [15] the existence of DAΠ solutions is proven, and it is proven that there is always an aggregation with only a small number of aggregates for which an approximating product form (7) exists which is as accurate as desired. Accuracy is verfied empirically: We solved many models using DA iteration, among others polling, fork join, stochastic Petri net models, queueing networks with blocking, losses, different service disciplines, phase type service times, and MAP arrivals, and we found very accurate results; the iteration did never diverge. Moreover, the accuracy could be substantially ameliorated in all examples if more and suitably chosen aggregates are considered. These matters are discussed in [15].

5 Numerical Examples

Here we consider the steady state probabilities of the component states. Other performance measures are calculated from them. Therefore their accuracy is of particular interest. In the examples with different aggregations the reader may note the gain in accuracy when finer aggregation with more aggregates is used.

Example: A loss network with Erlang-2 distributed service times. An open tandem network with 5 nodes is considered, the capacity is $\nu = 2$ in all nodes, Poisson arrivals occur at node 1 with rate 1, and all service times are Erlang-2 distributed with mean service time 0.5. The jobs go from node 1 to node 2, then to node 3, node 4, and finally to node 5, if they are not lost before due to full storage in a destination node.

First the states of each node with the waiting room and the system component which describes its service process are used to define a partition. The maximum error of the steady component state probabilities, i.e. the job numbers in waiting rooms and the phases of the service processes, is 0.06... .

The DA approximations are compared with exact Markov chain state probabilities. These were obtained from a Markov chain which was generated with the transition class model description. This is also true for the following and the last example.

If the joint states of all pairs of connected nodes are used for the definition of the aggregation, the maximum error is 0.007... .

If the joint states of all triplets of connected nodes are used for the definition of the aggregation, the maximum error is 0.00008... . □

Example: A loss network with bursty arrivals. An open tandem network with 8 nodes is considered. The capacity is $\nu = 2$ in all nodes. All service times are exponentially distributed with rate 2. The arrival process is defined by a hypergeometric distribution with mean arrival rate 1 and squared coefficient of variation 80.8... . The arrivals occur in bursts, the mean number of jobs per burst is 10, and an epoch without arrivals is ten times as long as an epoch with arrivals, in mean.

First the states of each node with the waiting room are used to define a partition, in node 1 also the system component which describes the arrival process. The maximum error of the steady component state probabilities, i.e. the job numbers in waiting rooms and the phases of the arrival process, is 0.048... .

If the joint states of all pairs of connected nodes are taken for the definition of the aggregation, this maximum error is 0.019... .

If the joint states of all triplets of connected nodes are taken for the definition of the aggregation, this maximum error is 0.009... . □

Example: A loss network with 8 different nodes. An open tandem network with 8 nodes is considered, the capacity is $\nu = 3$ in all nodes. Arrivals occur at node 1 with rate 1 according to a MAP which realizes hyperexponential interarrival times with a squared coefficient of variation 4. The jobs go from node 1 to node 2, then to node 3, and so on, finally to node 8, if they are not lost before due to full storage in a destination node. The service times are exponential (node 2 and 8), Erlang-4 (node 1 and 4), hyperexponential with squared coefficient of variation 4 (node 3 and 7), and hyperexponential with squared coefficient of variation 2 (node 5 and 6). The mean service rates are 2, 3.33, 3.33, 1.43, 2, 1.11, 1.43, and 1, respectively. This example is quite large, the Markov chain has 44,302,336 states.

First the states of each node with the waiting room and the system component which

describes its service process are used to define a partition, in node 1 also the system component which describes the arrival process. This aggregation consideres 72 MED parameters.

The DA approximations are compared with simulation results. These were obtained from the direct simulation of the transition class model description.

We consider the steady component state probabilities, i.e. the probabilities of job numbers in waiting rooms and of the phases of the arrival process and of the service processes.

The approximate component state probabilities of DA iteration are compared with estimated values of these probabilities from simulation. The maximum absolute difference is 0.084... . For all the estimated probabilities, 99% confidence intervals are as follows: $p \pm 0.02\sqrt{p - p^2}$ if p is an estimated probability.

Other performance measures are calculated from the approximate component state probabilities of DA iteration and from the estimated simulation values of these probabilities. The following table shows the absolute differences, relative to the simulation results, in %.

Node	1	2	3	4	5	6	7	8
Throughput	3.3	0.5	2.7	4.4	4.6	2.7	1.6	7.0
Mean Number of Jobs	0.9	0.8	0.6	7.7	1.8	5.4	1.2	0.4
Mean Response Time	4.1	1.3	3.5	11.7	2.6	2.6	0.3	6.1
Mean Waiting Time	9.9	6.1	10.9	28.0	7.1	5.9	0.9	14.4
Arrival Rate	0	3.3	0.5	2.7	4.4	4.6	2.7	1.6

The maximum absolute differences of the loss probabilities are as follows.

Node	1	2	3	4	5	6	7	8
Loss Probabilities	0.02	0.03	0.02	0.06	0.001	0.01	0.009	0.04

We tried also a finer aggregation with 7 partitions whose aggregates are defined according to the (joint) states of nodes 1 and 2, nodes 1 and 3, nodes 1 and 4, nodes 3 and 4, node 5, node 6, and nodes 7 and 8, respectively. This aggregation has 796 MED parameters. Here the maximum absolute difference of the component state probabilities is 0.034... , the maximum relative difference of the other performance measures is 10.2...%, and the maximum absolute difference of the loss probabilities is 0.049... . □

Example: A single queue with some Markovian arrival processes. Here we consider a single queue with $\nu = 3$ buffers for jobs. The service times are exponentially distributed. The queue is fed by some identical Markovian arrival processes. Each of them has hyperexponential interarrival times with a squared coefficient of variation 4. There is a partition for each MAP which defines the aggregates according to the state of the MAP and the number of jobs in the waiting room.

The maximum error of the steady component state probabilities of the waiting room and the arrival processes is as follows.

Number of MAPs	2	4	6	8	10
Maximum Error	0.29	0.14	0.09	0.09	0.03

For a small number of MAPs this aggregation is too coarse. But this example shows two interesting aspects:

It is not surprising that the error diminishes if the number of arrival processes grows. In conformity with the theorem of Palm-Khinchine, see for example [4], the overall arrival stream converges to a Poisson process, and for such an arrival process the DA method is here exact.

Because of the symmetry of the model, the corresponding aggregate probabilities of all partitions are equal. This can be easily exploited: Only the MED parameters of a single partition must be calculated which are also valid for the other partitions. □

Conclusion

Steady state probabilities of open queueing networks with Markovian arrival processes and phase type service times are investigated. Product form approximations of them are calculated with the disaggregation-aggregation (DA) method which is suitable for very large state spaces. The accuracy of this method was empirically verified for many models, among others priority, polling, fork join, blocking, loss, and stochastic Petri net models. By choosing aggregations with more aggregates – fine aggregation – the accuracy can be substantially ameliorated.

The queueing models are described using a specification technique which we call transition classes. Such a specification can be used to generate the Markov chain, its state space and the transition probability matrix, or for simulation. Furthermore these transition classes are applied to calculate efficiently the aggregate transition probabilities which are essential for our disaggregation-aggregation method.

At that time, a tool is developed which exploits transition classes. This tool generates Markov chains, calculatates the state probabilities, and analyzes the models with simulation. Additionally it supports the approximate disaggregation-aggregation method.

Further research concerns applications of the DA method on other Markov models, time dependent state probabilities, and stiff Markov chains.

References

[1] A. O. Allen. *Probability, Statistics and Queueing Theory.* Academic Press, 1990.

[2] W. J. Stewart (ed.). *Numerical Solution of Markov Chains.* Marcel Dekker, New York, Basel, Hong Kong, 1991.

[3] W. J. Stewart (ed.). *Computation with Markov Chains.* Kluwer, 1995.

[4] E.Gelenbe and R.R.Muntz. Probabilistic models of computer systems. *Acta Informatica*, pages 35–60, 1976.

[5] E. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems.* Academic Press, New York, 1980.

[6] B.R. Haverkort. Approximate analysis of networks of PH/PH/1/K queues: Theory & tool support. In H. Beilner and F. Bause, editors, *Quantitative Evaluation of Computing and Communication Systems*, pages 239–253. Springer, Berlin, 1995.

[7] B.R. Haverkort. Approximate analysis of networks of PH/PH/1/K queues: Test results. *Accepted for publication in Annals of Operations Research*, 1997.

[8] A. M. Kagan, J. V. Linnik, and C. R. Rao. *Characterization Problems in Mathematical Statistics*. Wiley, New York, 1973.

[9] W. Krämer and M. Langenbach-Belz. Approximate formulae for the delay in the queueing system GI/G/1. In *Proceedings ITC-8*, pages 235 1–8, Melbourne, 1976.

[10] P.J. Kühn. Approximate analysis of general queueing networks by decomposition. *IEEE Trans. Comm.*, pages 113–126, 1979.

[11] K.T. Marshall. Some inequalities in queueing. *Operations Research*, 16(3), 1968.

[12] M. F. Neuts. *Structured Stochastic Matrices of M/G/1–Type and their Applications*. Marcel Dekker, New York, 1989.

[13] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Priceton University Press, 1994.

[14] J. Ch. Strelen. Piecewise approximation of densities applying the principle of maximum entropy: Waiting times in G/G/1-systems. In R. Puigjaner and D. Poitier, editors, *Modelling Techniques and Tools for Computer Performance Evaluation*, pages 421–438. Plenum Press, New York, 1989.

[15] J. Ch. Strelen. Approximate product form solutions for Markov chains. *Performance Evaluation*, pages 87–110, 1997.

[16] J. Ch. Strelen, B. Bärk, J. Becker, and V. Jonas. Analysis of queueing networks with blocking using a new aggregation technique. *Accepted for publication in Annals of Operations Research*, 1997.

[17] J.Ch. Strelen. Approximate disaggregation-aggregation solutions for general queueing networks. In A.R. Kaylan and A. Lehmann, editors, *Proc. of the ESM 97 Conference*, pages 773–778. Society for Computer Simulation, 1997.