Analysis of Queueing Networks with Blocking using a New Aggregation Technique

Johann Christoph Strelen, Berthold Bärk, Jürgen Becker, Volker Jonas
Rheinische Friedrich–Wilhelms–Universität Bonn

Abstract

Queueing networks with repetitive-service blocking are investigated. The state probabilities are calculated applying a new aggregation technique, the *disaggregation-aggregation (DA) iteration.* Thus one obtains close approximate solutions with product form.

Introduction

Queueing networks have been successfully used in modeling a variety of computer systems, communication networks, and manufacturing systems. They can efficiently be analyzed if they have a product form solution. In queueing networks with finite capacities, blocking occurs ([11, 12]). Only in special cases they have a product form solution ([1]), otherwise the analysis is difficult due to complexity.

We investigate queueing network models with repetitive-service blocking: If a job tries to enter a node, and there is no buffer space available, the job goes back to its former node and instantaneously receives another service there. Afterwards, a new target node is chosen. In this paper, the steady state probabilities are approximated applying disaggregation-aggregation iteration ([13, 14]).

Disaggregation aggregation (DA) iteration is a new general approximate aggregation technique for finite Markov chains with discrete time. The calculated state probabilities are in product form. Here, neither the complete state space, nor all 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 treated, for example billions of states. It is also possible to calculate time-dependent state probabilities.

Time and space demands are higher than for many known, special purpose heuristics, but are by orders of magnitude lower than for exact solutions of Markov chains with many states. On the other hand, the results of DA iteration are usually more accurate than the heuristic solutions. Moreover, the accuracy can be influenced: The more aggregates are

This work was supported in part by the Deutsche Forschungsgemeinschaft
September 8, 1997

considered, the more accuracy can be obtained, if the aggregation is chosen suitably. If many aggregates are used, the aggregation is called *fine*.

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

We analyzed quite a lot of Markov chain models applying DA iteration, and we compared the obtained results with the exact solutions and with results which were calculated using heuristic methods known from the literature. The accuracy of DA solutions compares favourably.

We discuss DA iteration in section 1, its application to blocking networks in section 2, and numerical examples in section 3.

1 Disaggregation-aggregation iteration

Irreducible aperiodic Markov chains $(Z(t),\ t \in I\!N)$, discrete and homogeneous in time, with finite state space $\mathcal{Z} = [1 : n],\ n \in I\!N$, are considered. They stem from stochastic models of computer, communication, or manufacturing systems. These systems consist of interacting subsystems, their *components*. At every moment, each component is in one of its possible *component states*. This is why we denote the whole system's states also by *state tuples* $\mathbf{z} = (z_1, \ldots, z_K)$, where z_k indicates the state of component k. The states of component k are numbered $z_k = 0, \ldots, N_k$ for all components $k = 1, \ldots, K$.

Aggregates are sets of system states. An aggregate can be defined by the state of a certain 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. All aggregates $\mathcal{Z}_{k,0}, \ldots, \mathcal{Z}_{k,N_k}$ form a partition of the state space, for k fixed.

In the sequel, we will use the notions "partition" and "aggregate" slightly more generally. Therefore we introduce them more formally: A class $\{\mathcal{Z}_{k,0}, \ldots, \mathcal{Z}_{k,N_k}\},\ N_k \in I\!N,\ k \in I\!N$, of disjoint non-empty subsets of $\mathcal{Z}$ whose union is the full set $\mathcal{Z}$ itself is called *partition* $\mathcal{Z}_k$ of $\mathcal{Z}$. A class $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_C\},\ C \in I\!N$, of partitions is called an *aggregation* AGG of $\mathcal{Z}$, and each of the subsets $\mathcal{Z}_{k,j}$ of system states is an *aggregate*.

State tuples $\mathbf{z} = (z_1, \ldots, z_C) \in\ [0 : N_1] \times \ldots \times [0 : N_C]$ indicate the aggregates $\mathcal{Z}_{k,z_k}$ of each partition $\mathcal{Z}_1, \ldots, \mathcal{Z}_C$ to which a system state belongs. If for all state tuples the intersections $\mathcal{Z}_{1,z_1} \cap \ldots \cap \mathcal{Z}_{C,z_C}$ contain at most one element, each state tuple defines at most one system state $z \in \mathcal{Z}$, and the aggregation is said to *completely identify the states*. If the intersection is empty, there is no according state, and the state tuple $\mathbf{z}$ is termed *infeasible*, otherwise $\mathbf{z}$ describes a state and is *feasible*. In the sequel we consider only aggregations which completely identify the states. Thus, for each state $z \in \mathcal{Z}$ there is one and only one such state tuple $\mathbf{z}$. The set $\mathcal{Z}^T$ of all these feasible state tuples is another representation of the state space $\mathcal{Z}$. There is a one-to-one mapping $\mathcal{I} :\ \mathcal{Z}^T$ onto $\mathcal{Z}$.

A special aggregation is defined by the structure of the model as follows. Consider a system which consists of K subsystems, for example a queueing network with K nodes. If each aggregate is defined by the state z_k of a single subsystem, $\mathcal{Z}_{k,j} = \{\mathcal{I}(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}^T, z_k = j\},\ j = 0, \ldots, N_k,\ k = 1, \ldots, K$, where $[0 : N_k]$ is the set of all states of subsystem k, then we call the aggregation *canonical*. If the whole system's state is uniquely defined by the states of all subsystems, this canonical aggregation completely identifies the states. Consider for example a closed queueing network with N jobs circulating in it.

z_k denotes the number of jobs in node k. We assume that these numbers z_k describe the states of the nodes completely. Under these circumstances, any state tuple $(z_1, \ldots, z_K)$ where $z_1 + \ldots + z_K = N$ is feasible and indicates one and only one system state. If $z_1 + \ldots + z_K \neq N$, the tuple is infeasible – the state $(z_1, \ldots, z_K)$ cannot occur.

$p^{(t)}(z) \triangleq \mathrm{P}\{Z(t) = z\}, \;\; z \in \mathcal{Z}$, denotes the state probabilities of the Markov chain at time t. For convenience, we use also $p^{(t)}(\mathbf{z}), \; \mathbf{z} \in \mathcal{Z}^T$, where $p^{(t)}(\mathbf{z}) = p^{(t)}(z)$ holds if z and $\mathbf{z}$ denote the same state, i.e. $z = \mathcal{I}(\mathbf{z})$ holds. In the sequel, if z and $\mathbf{z}$ occur in the same context, they denote the same state.

The *probability vector* $\mathbf{p}^{(t)} = [p^{(t)}(1), \ldots, p^{(t)}(n)]$ contains all state probabilities at time t, and $\mathbf{P} \triangleq [p(x, y)]_{x,y=1}^{n}$ denotes the matrix of the transition probabilities. For convenience, we use also $p(\mathbf{x}, \mathbf{y}), \; \mathbf{x}, \mathbf{y} \in \mathcal{Z}^T$, where $p(\mathbf{x}, \mathbf{y}) = p(x, y)$ for $x = \mathcal{I}(\mathbf{x}), \; y = \mathcal{I}(\mathbf{y})$ holds.

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

Each aggregate $\mathcal{Z}_{k,j}$ defines a *macro probability* or *aggregate probability* which is a marginal probability as follows:

$$r_k(j) \triangleq \mathrm{P}\{Z \in \mathcal{Z}_{k,j}\} = \sum_{z \in \mathcal{Z}_{k,j}} p(z), \;\; j = 0, \ldots, N_k, \;\; k = 1, \ldots, C. \tag{1}$$

We collect the marginal probabilities into tuples $\mathbf{r}_k \triangleq \big(r_k(0), \ldots, r_k(N_k)\big), k = 1, \ldots, C$, and, in turn, these into $\mathbf{R} \triangleq (\mathbf{r}_1, \ldots, \mathbf{r}_C)$.

By (1), the *aggregation function*

$$\mathcal{A} : [0, 1]^n \to [0, 1]^{N_1+1} \times \ldots \times [0, 1]^{N_C+1}, \;\; \mathbf{p} \mapsto \mathbf{R},$$

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

Sometimes *joint marginal probabilities*

$$r_{k,k'}(j, j') \triangleq \mathrm{P}\{Z \in \mathcal{Z}_{k,j}, Z \in \mathcal{Z}_{k',j'}\} = \sum_{z \in \mathcal{Z}_{k,j} \cap \mathcal{Z}_{k',j'}} p(z),$$

$$j = 0, \ldots, N_k, \;\; j' = 0, \ldots, N_{k'}, \;\; \{k, k'\} \subseteq [1 : C],$$

are used. Again we collect them into tuples $\mathbf{r}_{k,k'}$.

Each partition $\mathcal{Z}_k, \; k \in [1 : C]$, defines a random variable $Z_k : \mathcal{Z}^T \to [0 : N_k], \; \mathbf{z} \mapsto z_k$. The marginal distribution $\mathbf{r}_k$ is the probability distribution of Z_k. Furthermore, sometimes it is more convenient to consider the random vector $\mathbf{Z} = (Z_1, \ldots, Z_C)$ for which $\mathrm{P}\{\mathbf{Z} = \mathbf{z}\} = p(\mathbf{z}), \; \mathbf{z} \in \mathcal{Z}^T$, holds, instead of the random variable Z.

Our proposed solution technique needs *disaggregation*: Given positive macro probabilities $\mathbf{R}$, a distribution $\boldsymbol{\pi} = [\pi(1), \ldots, \pi(n)]$ over $\mathcal{Z}$ is defined as follows. First, it satisfies the macro probabilities, $\mathbf{R} = \mathcal{A}(\boldsymbol{\pi})$, and secondly, $\boldsymbol{\pi}$ maximizes the entropy. Thus, according to the *product form theorem*, see Kagan, Linnik, Rao [7, theorem 13.2.1, page 409], this maximum entropy distribution (MED) $\boldsymbol{\pi}$ is uniquely determined and has a product form,

$$\pi(\mathbf{z}) = \frac{1}{G} \prod_{k=1}^{C} \rho_{k,z_k}, \tag{2}$$

where the $\rho_{k,z_k} > 0$ are some real numbers,

$$r_k(j) = \frac{1}{G} \sum_{\substack{\mathbf{z}\in\mathcal{Z}^T \\ z_k=j}} \prod_{i=1}^{C} \rho_{i,z_i},\ j = 0,\ldots,N_k,\ k = 1,\ldots,C, \tag{3}$$

holds, and

$$G = \sum_{\mathbf{z}\in\mathcal{Z}^T} \prod_{k=1}^{C} \rho_{k,z_k} \tag{4}$$

is a normalizing constant. In fact, the MED is completely determined by all *MED parameters* $\rho_{k,j}$, $\boldsymbol{\rho}$ for short.

By (2), (3), and (4), the *disaggregation function* is defined:

$$\mathcal{D} : [0,1]^{N_1+1} \times \ldots \times [0,1]^{N_C+1} \to [0,1]^n,\ \mathbf{R} \mapsto \boldsymbol{\pi}.$$

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

In general, the nonlinear equations (3) and (4) must be solved in order to calculate all MED parameters $\rho_{k,j}$ and the normalizing constant G of the MED $\boldsymbol{\pi}$, given the marginal probabilities $\mathbf{R}$. This is easily accomplished in some special cases. For example, if a canonical aggregation completely identifies the states, and if the state space is *rectangular*, i.e. if $\mathcal{Z}^T$ is a cartesian product, $\mathcal{Z}^T = [0 : N_1] \times \ldots \times [0 : N_C]$, the MED is given by

$$p(\mathbf{z}) = r_1(z_1) \cdot \ \ldots\ \cdot r_C(z_C),\ \mathbf{z} \in \mathcal{Z}^T, \tag{5}$$

which is easily verified (see [13]). Here the MED parameters are $\rho_{k,j} = r_k(j)$, $j = 0,\ldots,N_k$, $k = 1,\ldots,C$, and $G = 1$ holds. This can be generalized for the case where $\mathcal{Z}^T$ consists of the union of cartesian products, [14, section 3]. Another simple case where the state space is not necessarily rectangular are the state probabilities in product form queueing networks.

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:

$$\mathcal{T} : [0,1]^{N_1+1} \times \ \ldots\ \times [0,1]^{N_C+1} \to [0,1]^{N_1+1} \times \ \ldots\ \times [0,1]^{N_C+1},\ \mathbf{R} \mapsto \mathcal{A}\Big(\mathcal{D}(\mathbf{R})\mathbf{P}\Big).$$

This approximation $\mathbf{p}$ and its marginal probabilities $\mathbf{R}$ are referred to as *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*,

$$\mathbf{R}^{(t+1)} = \mathcal{T}(\mathbf{R}^{(t)}),\ t = 1,2,\ldots \tag{6}$$

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

This DA iteration resembles vector iteration, $\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)}\mathbf{P}$, $t = 1,2,\ldots$, beginning with any distribution $\mathbf{p}^{(1)}$. $\mathbf{p}^{(t)}$ are the time-dependent state probabilities and $\mathcal{A}(\mathbf{p}^{(t)})$

the time-dependent macro probabilities. $\mathbf{p}^{(t)}$ converges to the stationary probability vector and $\mathcal{A}(\mathbf{p}^{(t)})$ to the stationary macro 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 macro 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 stationary macro probabilities, and $\mathcal{D}(\mathbf{R}^{(t)})$ an approximation of the stationary state probabilities.

It is important to note that the function $\mathcal{T}$ must be evaluated without handling either all transition probabilities or all probabilities $\boldsymbol{\pi}^{(t)}$. Whether this is possible, depends on the transition probabilities and on the aggregation. According to our experience, such conditions are fulfilled very often in Markov chains originating from performance or reliability models. We have not yet found any model which could not be treated with DA iteration although we tried many different models. In [14, section 3], conditions are given under which this is possible.

More precisely, the conditions are fulfilled if the following expressions for the joint probabilities $\varphi_k(j,l)$ of the random vectors $(Z_k^{(t)}, Z_k^{(t+1)})$,

$$\varphi_k(j,l) = \sum_{\substack{\mathbf{y}\in\mathcal{Z}^T \\ y_k=j}} \sum_{\substack{\mathbf{z}\in\mathcal{Z}^T \\ z_k=l}} p(\mathbf{y},\mathbf{z})p^{(t)}(\mathbf{y}),\ j,l = 0,\ldots,N_k,\ k = 1,\ldots,C, \tag{7}$$

can be simplified substantially by collecting many summands into only a few expressions.

Observing

$$r_k^{(t+1)}(l) = \sum_{j=0}^{N_k} \varphi_k(j,l),\ l = 0,\ldots,N_k, \tag{8}$$

$$\varphi_k(j,j) = r_k^{(t)}(j) - \sum_{\substack{l=0 \\ l\neq j}}^{N_k} \varphi_k(j,l),\ j = 0,\ldots,N_k,\ k = 1,\ldots,C, \tag{9}$$

follows. (9) is already simple in the sense mentioned above.

The conditional probabilities

$$\pi_k(j,l) \triangleq \mathrm{P}\{Z_k^{(t+1)} = l \mid Z_k^{(t)} = j\} = \frac{\varphi_k(j,l)}{r_k^{(t)}(j)},\ \ j,l = 0,\ldots,N_k,\ k = 1,\ldots,C, \tag{10}$$

are *aggregate transition probabilities* as the following *aggregate transition equations*

$$r_k^{(t+1)}(l) = \sum_{j=0}^{N_k} \pi_k(j,l) r_k^{(t)}(j),\ l = 0,\ldots,N_k,\ k = 1,\ldots,C, \tag{11}$$

hold. The aggregate transition probabilities are not homogeneous in time, in general. According to (9), the *aggregate non-transition probabilities* are

$$\pi_k(j,j) = 1 - \sum_{\substack{l=0 \\ l\neq j}}^{N_k} \pi_k(j,l),\ j = 0,\ldots,N_k,\ k = 1,\ldots,C. \tag{12}$$

(8) and (9) or (11) and (12) are used to efficiently calculate the new macro probabilities $\mathbf{R}^{(t+1)}$ from the old ones, $\mathbf{R}^{(t)}$, thus evaluating the function $\mathcal{T}$.

If no closed form solution is available, the nonlinear equations (3) and (4) can be solved iteratively as follows ([6]) for the MED parameters $\boldsymbol{\rho}$, given the macro probabilities $\mathbf{R}$:

$$\rho_{k,j}^{[i+1]} = \Big(1 - \omega_i + \omega_i \frac{r_k(j)}{r_k^{[i]}(j)}\Big) \rho_{k,j}^{[i]},\ j = 0, \ldots, N_k,\ k = 1, \ldots, K,\ i = 1, 2, \ldots, \tag{13}$$

where the damping factor is chosen as $\omega_i = \max\{1,\ 1/2,\ 1/4, \ldots\}$ so that $\boldsymbol{\rho}^{[i+1]}$ fulfils the equations (3) and (4) more accurate than $\boldsymbol{\rho}^{[i]}$. Here $\boldsymbol{\rho}^{[1]}$ are initial approximations of the unknown parameters $\boldsymbol{\rho}$, and $r_k^{[i]}(j)$ are the macro probabilities according to the MED parameters $\boldsymbol{\rho}^{[i]}$, see (3). But in fact, the iteration (13) can be omitted when steady state solutions are desired. It is sufficient to apply just one iteration step,

$$\rho_{k,j}^{(t+1)} = \Big(1 - \omega_t + \omega_t \frac{r_k^{(t+1)}(j)}{r_k^{(t)}(j)}\Big) \rho_{k,j}^{(t)} \tag{14}$$

within each DA iteration step (6). This iteration did never diverge, using a damping factor $\omega_i = 1$ or $\omega_i = 1/2$.

We summarize, stating the

DA Iteration Algorithm

Input: Reasonable MED parameters $\boldsymbol{\rho}^{(1)}$ – for example all $\rho_{k,j}^{(1)} = 1$ – and a small bound $\epsilon > 0$

0) Calculate the normalization constant $G^{(1)}$ and the marginal probabilities $\mathbf{R}^{(1)}$. Let $t = 1$

1) Calculate the joint distribution functions $\varphi_k(j, l)$ for all j, l, k, or the aggregate transition probabilities $\pi_k(j, l)$ for all j, l, k

2) From these, calculate the new macro probabilities $\mathbf{R}^{(t+1)}$

3) Calculate the MED parameters $\boldsymbol{\rho}^{(t+1)}$ either directly, for example using (5), or approximately using (14). In the latter case, calculate again the macro probabilities $\mathbf{R}^{(t+1)}$ from the new MED parameters $\boldsymbol{\rho}^{(t+1)}$. Calculate the normalizing constant $G^{(t+1)}$

4) If $\max_{j,k} |r_k^{(t+1)}(j) - r_k^{(t)}(j)| < \epsilon$ holds, the iteration is completed. Otherwise let $t := t + 1$ and goto step 1)

Results: The macro probabilities $\mathbf{R}^{(t+1)}$, the MED parameters $\boldsymbol{\rho}^{(t+1)}$, and the normalizing constant $G^{(t+1)}$. $\square$

In [14] the DA product form theorem is proven: The equation $\mathbf{R} = \mathcal{T}(\mathbf{R})$ has a solution.

Unfortunately, we have not yet found error estimates for DA solutions or a proof of convergence of the iteration. But in [14] is proven that there is always an aggregation with only a small number of aggregates for which an approximating product form (2) exists which is as accurate as desired. And we solved many models using DA iteration, among others polling, fork join, stochastic Petri net, and blocking models, and we found very accurate results; the iteration did never diverge.

Moreover, the accuracy can be substantially ameliorated if more and suitably chosen aggregates are considered. We tried this with a lot of models. An aggregation with many aggregates is called a *fine grain aggregation*.

But how should the aggregates be chosen? Two observations are important for this question. If a system consists of subsystems which behave stochastically independent, and if a Markov model for this system is solved using DA iteration with canonical aggregation, canonical with respect to these components, the DA solution is exact. On the other hand, if a system consists of highly dependent components, the canonical approximation is expected to be inaccurate. Consequently, accurate approximations can be expected if the components are nearly independent. This observation serves as a guide for aggregation: Look for independent components, and design the aggregation accordingly, or combine the clusters of dependent components, each cluster into a single new conmponent, in order to get nearly independent components. In general, this produces more aggregates and requires more computing time. These matters will be treated in [2].

In section 2, fine grain aggregations for queueing networks with blocking are presented, and in section 3, numerical results are presented which verify that accurate approximations can be obtained using fine grain aggregation.

With DA iteration, time-dependent state probabilities can be calculated as follows: If the Markov chain is discrete in time, the DA solutions $\mathbf{p}^{(t)},\ t = 1, 2, \ldots$, are approximate time-dependent state probabilities, and $\mathbf{R}^{(t)}$ the according time-dependent macro probabilities. If the Markov chain is continuous in time, randomization ([5]) can be applied onto the DA solutions in order to obtain time-dependent solutions.

2 Queueing networks with repetitive-service blocking

We consider queueing networks with repetitive service blocking. A network 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. When a service of a job in node k is finished, the job tries to go to node l with transfer 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's state is given by the $K-$tuple $\mathbf{z} = (z_1, \ldots, z_K)$ where z_k denotes the number of jobs in node k. In an open queueing network, the state space

$$\mathcal{Z}^T = \mathcal{S}_1 \times \ldots \times \mathcal{S}_K$$

where $\mathcal{S}_k = [0 : \nu_k]$, and in a closed queueing network

$$\mathcal{Z}^T = \{\mathbf{z} \in \mathcal{S}_1 \times \ldots \times \mathcal{S}_K \mid z_1 + \ldots + z_K = N\}, \tag{15}$$

where

$$\mathcal{S}_k = [\tau_k : \nu_k], \quad \tau_k = \max\Big(0, N - \sum_{j \neq k} \nu_j\Big) :$$

A state $\tau_k > 0$ in node k occurs if no buffer space is left in any other node. □

The probabilities B_k that a job is blocked in node k, the *blocking probabilities*, are given by

$$B_k = \sum_{\substack{j=1 \\ j \neq k}}^{K} t_{k,j} \Big(\mathrm{P}\{Z_j = \nu_j\} - \mathrm{P}\{Z_k = 0,\ Z_j = \nu_j\}\Big), \quad k = 1, \ldots, K. \tag{16}$$

2.1 Open queueing networks

Here we treat open queueing networks with repetitive-service blocking. First, we give the Markov chain model. Secondly, DA iteration is presented applying canonical aggregation. Thirdly, fine grain solutions with more aggregates are described; in this manner even more accurate approximations are obtained.

The transitions from state $\mathbf{y} \in \mathcal{Z}^T$ into state $\mathbf{z} \in \mathcal{Z}^T$ occur as follows ($\mathbf{1}_k$ denotes the K-tuple with all elements 0 but 1 in position k):

1. arrival of a job at node k from outside, $\mathbf{z} = \mathbf{y} + \mathbf{1}_k$, if $y_k < \nu_k$, rate λ_k
2. service at node k is completed:
   (a) the job leaves the queueing network, $\mathbf{z} = \mathbf{y} - \mathbf{1}_k$, if $y_k > 0$, rate $\mu_k t_{k,0}$
   (b) the job goes from node k to node l, $l \neq k$, $\mathbf{z} = \mathbf{y} - \mathbf{1}_k + \mathbf{1}_l$, if $y_k > 0$ and $y_l < \nu_l$, rate $\mu_k t_{k,l}$.

Thus the sparse generator matrix $\mathbf{Q} = [q_{y,z}]_{y,z=1}^{n}$ is given by

$$q_{y,z} = \begin{cases} \mu_k t_{k,0} & \text{if } y_k > 0,\ z = \mathcal{I}(\mathbf{y} - \mathbf{1}_k), \\ \mu_k t_{k,l} & \text{if } y_k > 0,\ \ y_l < \nu_l,\ z = \mathcal{I}(\mathbf{y} - \mathbf{1}_k + \mathbf{1}_l), \\ \lambda_k & \text{if } y_k < \nu_k,\ z = \mathcal{I}(\mathbf{y} + \mathbf{1}_k), \\ -\sum_{\substack{x=1 \\ x \neq y}}^{n} q_{y,x} & \text{if } y = z, \\ 0 & \text{otherwise}, \end{cases} \tag{17}$$

$$\mathbf{y} = \mathcal{I}^{-1}(y),\ y, z = 1, \ldots, n.$$

In the sequel, we consider the Markov chain with the transition probability matrix $\mathbf{P} = \mathbf{Q}\Delta + \mathbf{I}$ where $\Delta = 0.99/\max_x |q_{x,x}|$. By the factor 0.99, numerical problems stemming from rounding errors are avoided. This Markov chain is discrete in time and has the same steady state probabilities.

Canonical aggregation

Here the number C of partitions equals the number K of nodes in the queueing network. Aggregate $\mathcal{Z}_{k,j}$ is the set of all states with j jobs in node k,

$$\mathcal{Z}_{k,j} = \{\mathcal{I}(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}^T,\ \ z_k = j\}, \quad k = 1, \ldots, K,\ j = 0, \ldots, \nu_k. \tag{18}$$

Disaggregation is effectuated according to (5). The aggregate transition probabilities are *

$$\pi_k(j,l) = \begin{cases} \alpha_k & \text{if } l = j-1 \geq 0, \\ 1-\beta_k & \text{if } l = j = 0, \\ 1-\alpha_k-\beta_k & \text{if } 0 < l = j < \nu_k, \\ 1-\alpha_k & \text{if } l = j = \nu_k, \\ \beta_k & \text{if } l = j+1 \leq \nu_k, \\ 0 & \text{otherwise,} \end{cases} \qquad j,l = 0,\ldots,\nu_k, \tag{19}$$

with

$$\alpha_k = \mu_k \Delta t_{k,0} + \sum_{\substack{c=1 \\ c\neq k}}^{K} \mu_k \Delta t_{k,c}\Big(1 - r_c(\nu_c)\Big), \quad \beta_k = \lambda_k \Delta + \sum_{\substack{c=1 \\ c\neq k}}^{K} \mu_c \Delta t_{c,k}\Big(1 - r_c(0)\Big),$$

$$k = 1,\ldots,K.$$

Observing the product form probabilities (5), the blocking probabilities are given by

$$B_k = \Big(1 - r_k(0)\Big)\Big(\sum_{\substack{j=1 \\ j\neq k}}^{K} t_{k,j} r_j(\nu_j)\Big), \quad k = 1,\ldots,K.$$

The number of aggregates is $\sum_{k=1}^{K}(\nu_k + 1)$.

Fine grain aggregation

Here we use more aggregates, and thus we may expect – and in fact we obtain – more accurate solutions. More specific, we define aggregates according to the state of *two* nodes. Thus the dependencies between two such nodes are completely taken into account. The partitions are denoted by $\mathcal{Z}_{(k,k')}$, $k,k' \in [1:K]$, $k \neq k'$, and the aggregates are

$$\mathcal{Z}_{(k,k'),(l,l')} = \{\mathcal{I}(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}^T,\ z_k = l,\ z_{k'} = l'\}, \quad l = 0,\ldots,\nu_k,\ l' = 0,\ldots,\nu_{k'}.$$

Not all partitions $\mathcal{Z}_{(k,k')}$, $k \neq k'$, are considered, but only for a subset $\mathcal{K}_2 \subset [1:K]^2 \setminus \{(1,1),\ldots,(K,K)\}$ of node pairs.

Here, the macro probabilities are the joint marginal probabilities $r_{k,k'}(j,j')$. They are calculated corresponding to (11) by the aggregate transition equations

$$r_{k,k'}^{(t+1)}(l,l') = \sum_{j=0}^{\nu_k} \sum_{j'=0}^{\nu_{k'}} \pi_{k,k'}\Big((j,j'),(l,l')\Big) r_{k,k'}^{(t)}(j,j'), \quad (l,l') \in \mathcal{S}_k \times \mathcal{S}_{k'}, \quad (k,k') \in \mathcal{K}_2. \tag{20}$$

In general, the aggregate transition probabilities $\pi_{k,k'}\Big((j,j'),(l,l')\Big)$ differ for different choices of the set $\mathcal{K}_2$.

Sometimes the dependencies between a singular node $k^* \in [1:K]$ and the other nodes are of particular importance, for example in a star topology. Then we try *center aggregation* with the partitions

$$\mathcal{Z}_{(k^*,k)},\ k = 1,\ldots,K,\ k \neq k^*, \quad \mathcal{K}_2 = \Big\{(k^*,k)\Big| k = 1,\ldots,K,\ k \neq k^*\Big\},$$

*Proofs are omitted in this paper. They are straightforward and sometimes lengthy.

and the partition $\mathcal{Z}_{k^*}$. Node k^* is called the *center*. The number of aggregates is $(\nu_{k^*}+1)\left(1+\sum_{\substack{c=1\\c\neq k^*}}^{K}(\nu_c+1)\right)$, and the MED calculated from given marginal probabilities is

$$p(\mathbf{z}) = r_{k^*}(z_{k^*}) \prod_{\substack{c=1\\c\neq k^*}}^{K} \frac{r_{k^*,c}(z_{k^*}, z_c)}{r_{k^*}(z_{k^*})}, \quad \mathbf{z} \in \mathcal{Z}^T. \tag{21}$$

The aggregate transition probabilities are

$$\pi_{k,k'}\Big((j,j'),(l,l')\Big) =$$
$$\begin{cases} \mu_{k^*}\Delta t_{k^*,0} + \sum\limits_{\substack{c=1\\c\neq k^*,k'}}^{C} \mu_{k^*}\Delta t_{k^*,c}(1-\dfrac{r_{k^*c}(j,\nu_c)}{r_{k^*}(j)}) & \text{if } l=j-1\geq 0, l'=j', \\ \mu_{k'}\Delta t_{k',0} + \sum\limits_{\substack{c=1\\c\neq k^*,k'}}^{C} \mu_{k'}\Delta t_{k',c}(1-\dfrac{r_{k^*c}(j,\nu_c)}{r_{k^*}(j)}) & \text{if } l=j, l'=j'-1\geq 0, \\ \lambda_{k^*}\Delta + \sum\limits_{\substack{c=1\\c\neq k^*,k'}}^{C} \mu_c\Delta t_{c,k^*}(1-\dfrac{r_{k^*c}(j,0)}{r_{k^*}(j)}) & \text{if } l=j+1\leq \nu_k, l'=j', \\ \lambda_{k'}\Delta + \sum\limits_{\substack{c=1\\c\neq k^*,k'}}^{C} \mu_c\Delta t_{c,k'}(1-\dfrac{r_{k^*c}(j,0)}{r_{k^*}(j)}) & \text{if } l=j, l'=j'+1\leq \nu_{k'}, \\ \mu_k\Delta t_{k,k'} & \text{if } l=j-1\geq 0, l'=j'+1\leq \nu_{k'}, \\ \mu_{k'}\Delta t_{k',k} & \text{if } l=j+1\leq \nu_k, l'=j'-1\geq 0, \\ 0 & \text{if } |l+l'-j-j'|>1, \end{cases}$$
$$j,l=0,\ldots,\nu_k,\quad j',l'=0,\ldots,\nu_{k'},\quad (j,j')\neq(l,l'),\quad (k,k')\in\mathcal{K}_2.$$

Here and in the sequel, the aggregate non-transition probabilities $\pi_{k,k'}\Big((j,j'),(j,j')\Big)$ are calculated according to (12). Aggregate transition probabilities for the partition $\mathcal{Z}_{k^*}$ are not needed, the marginal probabilities $r_{k^*}(j),\ j\in\mathcal{S}_{k^*}$, are obtained by (22).

The blocking probabilities are obtained using (16) where the marginal probabilities are given by

$$\begin{aligned} r_k(j) &= \begin{cases} \sum\limits_{j'=0}^{\nu_{k*}} r_{k^*k}(j',j) & \text{if } k\neq k^*, \\ \sum\limits_{j'=0}^{\nu_{k'}} r_{k^*k'}(j,j') & \text{if } k=k^*, \end{cases} \qquad k=1,\ldots,K,\ k'\in[1:K]\setminus\{k^*\}, \\ r_{k,k'}(j,j') &= \begin{cases} \sum\limits_{j''=0}^{\nu_{k*}} \dfrac{r_{k^*,k}(j'',j)r_{k^*,k'}(j'',j')}{r_{k^*}(j'')} & \text{if } k\neq k^*,\ k'\neq k^*, \\ r_{k^*k'}(j,j') & \text{if } k=k^*,\ k'\neq k^*, \\ r_{k^*k}(j',j) & \text{if } k\neq k^*,\ k'=k^*, \end{cases} \\ j &= 0,\ldots,\nu_k,\ j'=0,\ldots,\nu_{k'},\ k,k'=1,\ldots,K,\ k\neq k'. \end{aligned} \tag{22}$$

Sometimes the states of separate pairs (k,k') of nodes depend strongly. Under such circumstances one can put these pairs into $\mathcal{K}_2$ and consider the according partitions $\mathcal{Z}_{(k,k')},\ (k,k')\in\mathcal{K}_2$. If K is even, and all nodes of the network are combined into such pairs,

$$\mathcal{K}_2 = \Big\{ \Big(\mathrm{per}(1), \mathrm{per}(2)\Big),\ \Big(\mathrm{per}(3), \mathrm{per}(4)\Big), \ldots, \Big(\mathrm{per}(K-1), \mathrm{per}(K)\Big) \Big\}$$

holds, where $\Big(\mathrm{per}(1), \ldots, \mathrm{per}(K)\Big)$ is a permutation of $(1, \ldots, K)$. We call this *paired aggregation.*

The number of aggregates is $\sum_{(k,k')\in\mathcal{K}_2} (\nu_k+1)(\nu_{k'}+1)$, and the MED is

$$p(\mathbf{z}) = \prod_{(k,k')\in\mathcal{K}_2} r_{k,k'}(z_k, z_{k'}), \quad \mathbf{z} \in \mathcal{Z}^T.$$

Here the aggregate transition probabilities are

$$\pi_{k,k'}\Big((j,j'),(l,l')\Big) = \begin{cases} \mu_k \Delta t_{k,0} + \sum\limits_{\substack{c=1\\ c\neq k,k'}}^{C} \mu_k \Delta t_{k,c}\Big(1 - r_c(\nu_c)\Big) & \text{if } l = j-1 \geq 0, l' = j', \\ \mu_{k'} \Delta t_{k',0} + \sum\limits_{\substack{c=1\\ c\neq k,k'}}^{C} \mu_{k'} \Delta t_{k',c}\Big(1 - r_c(\nu_c)\Big) & \text{if } l = j, l' = j'-1 \geq 0, \\ \lambda_k \Delta + \sum\limits_{\substack{c=1\\ c\neq k,k'}}^{C} \mu_c \Delta t_{c,k}\Big(1 - r_c(0)\Big) & \text{if } l = j+1 \leq \nu_k, l' = j', \\ \lambda_{k'} \Delta + \sum\limits_{\substack{c=1\\ c\neq k,k'}}^{C} \mu_c \Delta t_{c,k'}\Big(1 - r_c(0)\Big) & \text{if } l = j, l' = j'+1 \leq \nu_{k'}, \\ \mu_k \Delta t_{k,k'} & \text{if } l = j-1 \geq 0, l' = j'+1 \leq \nu_{k'}, \\ \mu_{k'} \Delta t_{k',k} & \text{if } l = j+1 \leq \nu_k, l' = j'-1 \geq 0, \\ 0 & \text{if } |l + l' - j - j'| > 1, \end{cases}$$

$$j, l = 0, \ldots, \nu_k, \quad j', l' = 0, \ldots, \nu_{k'}, \quad (j,j') \neq (l,l'), \quad (k,k') \in \mathcal{K}_2.$$

The blocking probabilities are obtained using (16) where the required marginal probabilities are given by

$$\begin{aligned} r_k(j) &= \begin{cases} \sum\limits_{j''=0}^{\nu_k} r_{k'',k}(j'', j) & \text{if } (k'',k) \in \mathcal{K}_2, \\ \sum\limits_{j''=0}^{\nu_{k''}} r_{k,k''}(j, j'') & \text{if } (k,k'') \in \mathcal{K}_2, \end{cases} \qquad j = 0, \ldots, \nu_k,\ k = 1, \ldots, K, \\ r_{k,k'}(j,j') &= \begin{cases} r_{k,k'}(j,j') & \text{if } (k,k') \in \mathcal{K}_2, \\ r_k(j) r_{k'}(j') & \text{otherwise}, \end{cases} \\ j &= 0, \ldots, \nu_k,\ j' = 0, \ldots, \nu_{k'},\ k, k' = 1, \ldots, K. \end{aligned}$$

Clearly, the states of two nodes k and k' are dependent if the nodes are *connected*, i.e. if jobs travel from node k to node k' or vice versa. Therefore it seems reasonable to consider all these pairs:

$$\mathcal{K}_2 = \Big\{ (k,k') \Big| k < k',\ t_{k,k'} > 0 \text{ or } t_{k',k} > 0 \Big\}.$$

We call this the *all connected pairs (ACP) aggregation*.

First, we restrict ourselves to tandem queues where jobs arrive from outside at node 1 with rate λ_1, travel from node k to node $k+1$, $k = 1, \ldots, K-1$, and leave the network at node K. Thus, $\mathcal{K}_2 = \{(1,2), (2,3), \ldots, (K-1, K)\}$ is the set of considered pairs for ACP aggregation.

There are $C = K-1$ partitions of the state space $\mathcal{Z}^T$, and the aggregate probabilities are joint marginal probabilities, namely

$$r_k(j,j') \triangleq \mathrm{P}\{Z_k = j,\ Z_{k+1} = j'\},\ j \in \mathcal{S}_k,\ j' \in \mathcal{S}_{k+1},\ k = 1, \ldots, C.$$

The number of aggregates is $\sum_{c=1}^{C}(\nu_c + 1)(\nu_{c+1} + 1)$. Disaggregation is obtained by

$$p(\mathbf{z}) = \frac{r_1(z_1, z_2) \cdot \ \ldots \ \cdot r_C(z_C, z_{C+1})}{r_2(z_2) \cdot \ \ldots \ \cdot r_C(z_C)},\ \mathbf{z} \in \mathcal{Z}^T.$$

From this, one obtains

$$r_k(j) = \sum_{j'=0}^{\nu_{k+1}} r_k(j,j'),\ k = 1, \ldots, C, \quad \text{and} \quad r_k(j) = \sum_{j'=0}^{\nu_{k-1}} r_{k-1}(j',j),\ k = 2, \ldots, K.$$

The aggregate transition probabilities are

$$\pi_{k,k+1}\Big((j,j'),(l,l')\Big) =$$
$$\begin{cases} \lambda_1 \Delta & \text{if } k = 1,\ j+1 = l,\ j' = l',\ j < \nu_1, \\ \mu_{k-1}\Delta\Big(1 - r_{k-1}(0,j)/r_k(j)\Big) & \text{if } 1 < k,\ j+1 = l,\ j' = l',\ j < \nu_k, \\ \mu_k \Delta & \text{if } j-1 = l,\ j'+1 = l',\ j > 0,\ j' < \nu_{k+1}, \\ \mu_{k+1}\Delta\Big(1 - r_{k+1}(j',\nu_{k+2})/r_{k+1}(j')\Big) & \text{if } k < C,\ j = l,\ j'-1 = l',\ j' > 0, \\ \mu_K \Delta & \text{if } k = C,\ j = l,\ j'-1 = l',\ j' > 0, \\ 0 & \text{otherwise if } \ (j,j') \neq (l,l'), \end{cases}$$
$$j,l = 0, \ldots, \nu_k, \quad j',l' = 0, \ldots, \nu_{k+1}, \quad k = 1, \ldots, C.$$

The blocking probabilities are obtained using (16), $B_k = r_{k+1}(\nu_{k+1}) - r_k(0, \nu_{k+1}),\ k = 1, \ldots, K-1$.

Now we discuss an aggregation which is even finer than the all connected pairs of nodes aggregation, the *all connected triplets (ACT) aggregation*. Here we restrict ourselves to open tandem queues as above. The triplets of nodes are $(k, k+1, k+2),\ k = 1, \ldots, K-2$. There are $C = K-2$ partitions $\mathcal{Z}_k$ of the state space with the aggregates

$$\begin{aligned} \mathcal{Z}_{k,(j,j',j'')} &= \{\mathcal{I}(\mathbf{z}) \,|\, \mathbf{z} \in \mathcal{Z}^T,\ z_k = j,\ z_{k+1} = j',\ z_{k+2} = j''\}, \\ & \quad\ (j,j',j'') \in \mathcal{S}_k \times \mathcal{S}_{k+1} \times \mathcal{S}_{k+2},\ k = 1, \ldots, C. \end{aligned}$$

We will use the marginal probabilities $r_k(j)$, the joint marginal probabilities $r_k(j,j')$, and the aggregate probabilities are the *type-3 marginal probabilities*

$$r_k(j,j',j'') \triangleq \mathrm{P}\{Z_k = j,\ Z_{k+1} = j',\ Z_{k+2} = j''\},\ (j,j',j'') \in \mathcal{S}_k \times \mathcal{S}_{k+1} \times \mathcal{S}_{k+2},\ k = 1, \ldots, C.$$

The number of aggregates is $\sum_{k=1}^{C}(\nu_k+1)(\nu_{k+1}+1)(\nu_{k+2}+1)$, and disaggregation is obtained by

$$p(\mathbf{z}) = \frac{r_1(z_1,z_2,z_3)\cdot\ \ldots\ \cdot r_C(z_C,z_{C+1},z_{C+2})}{r_2(z_2,z_3)\cdot\ \ldots\ \cdot r_C(z_C,z_{C+1})},\quad \mathbf{z}\in\mathcal{Z}^T.$$

From this one obtains the *type-4 marginal probabilities*

$$\begin{aligned} r_k(j,j',j'',j''') &\triangleq \mathrm{P}\{Z_k=j,\ Z_{k+1}=j',\ Z_{k+2}=j'',\ Z_{k+3}=j'''\} \\ &= \frac{r_k(j,j',j'')r_{k+1}(j',j'',j''')}{r_{k+1}(j',j'')},\quad k=1,\ldots,C-1, \end{aligned}$$

which are needed for the aggregate transition probabilities

$$\pi_k\Big((j,j',j''),(l,l',l'')\Big) = \begin{cases} \lambda_1\Delta & \text{if } k=1,\ j+1=l,\ j<\nu_k, \\ \mu_{k-1}\Delta\Big(1-\frac{r_{k-1}(0,j',j'',j''')}{r_k(j,j',j'')}\Big) & \text{if } k>1,\ j+1=l,\ j<\nu_k, \\ \mu_k\Delta & \text{if } j-1=l,\ j'+1=l',\ j>0,\ j'<\nu_{k+1}, \\ \mu_{k+1}\Delta & \text{if } j'-1=l',\ j''+1=l'',\ j'>0,\ j''<\nu_{k+2}, \\ \mu_{k+2}\Delta\Big(1-\frac{r_k(j,j',j'',\nu_{k+3})}{r_k(j,j',j'')}\Big) & \text{if } k<C,\ j''-1=l'',\ j''>0, \\ \mu_{k+2}\Delta & \text{if } k=C,\ j''-1=l'',\ j''>0, \\ 0 & \text{otherwise if } (j,j',j'')\neq(l,l',l''), \end{cases}$$

$$(j,j',j''),\ (l,l',l'')\in\mathcal{S}_k\times\mathcal{S}_{k+1}\times\mathcal{S}_{k+2},\ k=1,\ldots,C.$$

Clearly, the joint marginal probabilities are

$$r_k(j,j') = \begin{cases} \sum_{j''\in\mathcal{S}_{k+2}} r_k(j,j',j'') & \text{if } k\le K-2, \\ \sum_{j''\in\mathcal{S}_k} r_k(j'',j,j') & \text{if } k>1, \end{cases}\qquad j\in\mathcal{S}_k,\ j'\in\mathcal{S}_{k+1},\ k=1,\ldots,K.$$

The blocking probabilities are obtained using (16).

2.2 Closed queueing networks

Again, we first give a Markov chain model, and then we present DA iteration with canonical and fine grain aggregation. Here the solution of nonlinear equations is needed for disaggregation.

A state $\mathbf{z}=(z_1,\ldots,z_K)$ of the Markov chain gives the numbers z_k of jobs in node k, $k=1,\ldots,K$, and the state space is given in (15). The only transitions from state $\mathbf{y}\in\mathcal{Z}^T$ into state $\mathbf{z}\in\mathcal{Z}^T$ occur when a job's service in node k is finished and this job goes to node l: $\mathbf{z}=\mathbf{y}-\mathbf{1}_k+\mathbf{1}_l$ if $y_l<\nu_l$, $y_k>\tau_k$, and $k\neq l$, with transition rate $\mu_k t_{k,l}$. Thus the generator matrix $\mathbf{Q}$ has the elements

$$\begin{aligned} q_{y,z} &= \begin{cases} \mu_k t_{k,l} & \text{if } y_k>\tau_k,\ y_l<\nu_l,\ z=\mathcal{I}(\mathbf{y}-\mathbf{1}_k+\mathbf{1}_l), \\ -\sum_{\substack{x=1\\x\neq y}}^{n} q_{y,x} & \text{if } y=z, \\ 0 & \text{otherwise}, \end{cases} \\ \mathbf{y} &= \mathcal{I}^{-1}(y),\quad y,z=1,\ldots,n, \end{aligned}$$

and $\mathbf{P}=\mathbf{Q}\Delta+\mathbf{I}$ with $\Delta=0.99/\max_x|q_{x,x}|$ is the transition probability matrix of a Markov chain with discrete time, which has the same steady state probabilities.

Canonical aggregation

Again the number C of partitions equals the number K of nodes in the queueing network. N is the population. Aggregate $\mathcal{Z}_{k,j}$ is the set of all states with j jobs in node k.

The new macro probabilities $\mathbf{R}^{(t+1)}$ are obtained according to (8) and (9) using

$$\varphi_k(j,l) = \begin{cases} \alpha_k^j & \text{if } l = j-1 \geq \tau_k, \\ r_k(j) - \beta_k^j & \text{if } l = j = \tau_k, \\ r_k(j) - \alpha_k^j - \beta_k^j & \text{if } \tau_k < l = j < \nu_k, \\ r_k(j) - \alpha_k^j & \text{if } l = j = \nu_k, \\ \beta_k^j & \text{if } l = j+1 \leq \nu_k, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\alpha_k^j = \sum_{\substack{c=1 \\ c\neq k}}^{K} \mu_k \Delta t_{k,c} \Big(r_k(j) - r_{k,c}(j,\nu_c) \Big), \quad \beta_k^j = \sum_{\substack{c=1 \\ c\neq k}}^{K} \mu_c \Delta t_{c,k} \Big(r_k(j) - r_{k,c}(j,\tau_c) \Big),$$

$$j, l = \tau_k, \ldots, \nu_k, \quad k = 1, \ldots, K.$$

The marginal probabilities and the normalizing constant G are efficiently calculated from the MED parameters $\boldsymbol{\rho}$ by convolution:

$$G = (\boldsymbol{\rho}_1 \otimes \ldots \otimes \boldsymbol{\rho}_K)_N,$$

$$r_k(j) = \frac{\rho_{k,j}}{G} \left(\bigotimes_{\substack{c=1 \\ c\neq k}}^{K} \boldsymbol{\rho}_c \right)_{N-j}, \quad j = \tau_k, \ldots, \nu_k, \; k = 1, \ldots, K,$$

$$r_{k,k'}(j,j') = \frac{\rho_{k,j}\rho_{k',j'}}{G} \left(\bigotimes_{\substack{c=1 \\ c\neq k,k'}}^{K} \boldsymbol{\rho}_c \right)_{N-j-j'}, \quad (j,j') \in \mathcal{S}_k \times \mathcal{S}_{k'}, \quad \{k,k'\} \subseteq [1:K],$$

where $\boldsymbol{\rho}_c \triangleq (\rho_{c,\tau_c}, \ldots, \rho_{c,\nu_c})$, $c = 1, \ldots, K$, and $\otimes$ is the convolution operator ($\rho_{k,j} = 0$ for all $j \notin [\tau_k : \nu_k]$). Approximate disaggregation is done according to (14).

Fine grain aggregation

Now we consider ACP aggregation for closed queueing networks with cyclic topology: The N jobs travel from node k to node $k \oplus 1$ where $k \oplus 1$ equals 1 if $k = K$ and $k+1$ otherwise.

Two subsequent nodes k, $k \oplus 1$ may assume the states

$$\mathcal{S}_k^{(2)} \triangleq \Big\{ (z_k, z_{k\oplus 1}) \,\Big|\, \mathbf{z} \in \mathcal{Z}^T \Big\}$$
$$= \Big\{ (j,j') \,\Big|\, \max\Big(\tau_{k\oplus 1},\, N - \sum_{\substack{i\neq k, \\ i\neq k\oplus 1}} \nu_i - j\Big) \leq j' \leq \min\Big(\nu_{k\oplus 1},\, N - \sum_{\substack{i\neq k, \\ i\neq k\oplus 1}} \tau_i - j\Big),\; j \in \mathcal{S}_k \Big\}.$$

There are $C = K$ partitions of the state space $\mathcal{Z}^T$. The aggregate probabilities are the joint marginal probabilities $r_k(j,j') \triangleq \mathrm{P}\{Z_k = j, Z_{k\oplus 1} = j'\}$, $(j,j') \in \mathcal{S}_k^{(2)}$, $k = 1, \ldots, K$.

The state probabilities are

$$p(\mathbf{z}) = \rho_{1,z_1,z_2} \cdot \ldots \cdot \rho_{K,z_K,z_1}/G, \;\; \mathbf{z} \in \mathcal{Z}^T,$$

where the MED parameters $\rho_{k,j,j'}$, $(j,j') \in \mathcal{S}_k^{(2)}$, $k = 1,\ldots,K$, and the normalization constant G are solutions of the nonlinear equations (3) and (4).

The normalization constant G is efficiently calculated as follows, given the MED parameters. Let

$$\begin{aligned}
\sigma_k(z_1, z_{k-1}, j_k) &\triangleq \begin{cases} \rho_{K-1,z_{k-1},j_k}\rho_{K,j_k,z_1} & \text{if } \; k = K, \\ \displaystyle\sum_{z_k=\tau_k}^{\min(\nu_k,j_k)} \rho_{k-1,z_{k-1},z_k}\sigma_{k+1}(z_1,z_k,j_k-z_k) & \text{if } \; k < K, \end{cases} \\
&\quad z_1 \in \mathcal{S}_1, \;\; j_k = N - z_1 \;\text{ if }\; k = 2, \\
&\quad z_{k-1} = \tau_{k-1},\ldots,\min(\nu_{k-1}, N - z_1) \;\text{ and} \\
&\quad j_k = \max(0,\, N - z_1 - \nu_2 - \ldots - \nu_{k-2} - z_{k-1},\, \tau_k + \ldots + \tau_K) \\
&\quad\quad\quad ,\ldots,\min(N - z_1 - \tau_2 - \ldots - \tau_{k-2} - z_{k-1},\, \nu_k + \ldots + \nu_K) \;\text{ if }\; k > 2, \\
&\quad k = 2,\ldots,K, \\
\sigma_1 &\triangleq \sum_{z_1=\tau_1}^{\nu_1} \sigma_2(z_1, z_1, N - z_1).
\end{aligned} \tag{23}$$

Then $G = \sigma_1$ holds.

The *type-3 marginal probabilities*,

$$r_k(j,j',j'') \triangleq \mathrm{P}\{Z_k = j, Z_{k\oplus 1} = j', Z_{k\oplus 2} = j''\},$$

$(j,j',j'') \in \{(z_k, z_{k\oplus 1}, z_{k\oplus 2}) | \mathbf{z} \in \mathcal{Z}^T\}$, where $k\oplus 2 \triangleq (k\oplus 1)\oplus 1$, are needed for the aggregate transition probabilities. They are calculated similarly: Replace in (23) τ_k and ν_k by j, $\tau_{k\oplus 1}$ and $\nu_{k\oplus 1}$ by j', $\tau_{k\oplus 2}$ and $\nu_{k\oplus 2}$ by j'' , and calculate σ_1; then $r_k(j,j',j'') = \sigma_1/G$ holds.

The aggregate transition probabilities are

$$\begin{aligned}
&\pi_{k,k\oplus 1}\Big((j,j'),(l,l')\Big) = \\
&\quad \begin{cases} \mu_{k\ominus 1}\Delta\Big(1 - r_{k\ominus 1}(\tau_{k\ominus 1}, j, j')/r_k(j,j')\Big) & \text{if } j+1 = l,\; j' = l',\; j < \nu_k, \\ \mu_k\Delta & \text{if } j-1 = l,\; j'+1 = l',\; j > \tau_k,\; j' < \nu_{k\oplus 1}, \\ \mu_{k\oplus 1}\Delta\Big(1 - r_{k\oplus 1}(j, j', \nu_{k\oplus 2})/r_k(j,j')\Big) & \text{if } j = l,\; j'-1 = l',\; j' > \tau_{k\oplus 1}, \\ 0 & \text{otherwise if }\; (j,j') \neq (l,l'), \end{cases} \\
&(j,j'),\,(l,l') \in \mathcal{S}_k^{(2)}, \;\; k = 1,\ldots,K,
\end{aligned}$$

where $k \ominus 1$ equals K if $k = 1$ and $k - 1$ otherwise.

When DA iteration for stationary probabilities is to be done, we omit as usual the solution of the nonlinear equations (3) and (4) and apply iteration (14) instead.

The blocking probabilities are obtained using (16).

3 Numerical experiences

In order to find out what the merits of DA iteration are with respect to accuracy and computing time, we analyzed many Markov models applying this method, and compared the obtained approximations with the exact figures from the Markov chain solution. Besides we considered heuristic techniques from the literature, calculated solutions with them, and compared the accuracy of these results with ours.

Typically, an open or closed queueing network is analyzed for a variety of system parameters like number of nodes, number of jobs in the network, buffer capacity of a node, arrival rates, or service rates, using some different solution techniques like vector iteration for exact results, DA iteration with canonical aggregation and with fine grain aggregation, or methods from the literature. For each approximation technique, we report the largest observed approximation error. Furthermore, for a quite large model, we report the computing times which were consumed by vector iteration and by DA iteration with canonical and with fine grain aggregation. For a very large model which cannot be analyzed exactly, we present the computing time for DA iteration with canonical aggregation.

We present an outline of our results analyzing the following queueing network models with repetitive-service blocking:

1. Open tandem networks, arrivals at the first node. Comparison with results according to Caseau and Pujolle [3]. Their technique relies on a heuristic: the sevice rates μ_k, $k = 1, \ldots, K$, are augmented in order to take into account repeated services which are due to blocking.

2. Open tandem networks. Comparison with results according to Labetoulle and Pujolle [10]. They use diffusion approximation for the solution. The blocking networks are open, interarrival times and service times are generally distributed. Here, the blocking policy is a bit different, a rejected job clings to the next node (*repetitive-service with fixed destination*). In a tandem network both blocking policies are equivalent.

3. Open networks with various topologies. Comparison with results according to Kouvatsos, Xenios and Denazis [9, 8]. Their algorithm is quite general, suited for open and closed queueing networks with repetitive-service blocking. Interarrival times and service times have general exponential distributions. The state probabilities are obtained by maximizing the entropy observing some reasonable constraints.

4. Comparison of DA iteration with coarse, fine, and very fine aggregation for open networks with tandem, split, merge, and symmetrical topology.

5. Closed, cyclic networks. Comparison with results obtained by the technique of Frein and Dallery [4]. Their decomposition method can analyze closed cyclic networks with blocking-before-service policy which is equivalent to the repetitive-service policy in these networks.

6. Comparison of the computing times for vector iteration and DA iteration, applied to a symmetrical topology with a large number of states.

7. A tandem topology with 10^9 system states. DA iteration consumes only a moderate amount of computing time for this large model.

The accuracy is measured using absolute errors for probabilities and utilizations and tolerances for mean job numbers and throughputs. The absolute error of any performance measure in node k is $|e_k - a_k|$ where e_k is the true value of this measure for node k, and a_k the calculated approximation of it. The tolerance is defined as $|e_k - a_k| / \sum_{l=1}^{K} |e_l|,\ k = 1, \ldots, K$.

1. The considered queueing networks have a tandem topology with K nodes, where jobs from node k go to node $k+1,\ k = 1, \ldots, K-1$. Arrivals occur only in node 1, and the jobs which leave node K leave the network.

We report the maximum absolute errors and the maximum tolerances, maximum over all nodes and all examples 1.1 and 1.2, respectively, achieved by the heuristic technique from the literature (L) and by DA iteration with canonical aggregation (DAI-canonical) and center aggregation (DAI-center):

Examples 1.1 (Tandem network) K identical nodes: $\lambda_1 = 1,\ \mu_k = 1,\ \nu_k = 3,$ $k = 1, \ldots, K,\ K = 2, \ldots, 7$. The maximum absolute errors of the probabilities of the job numbers in the nodes (macro probabilities) are 0.14 (L) and 0.007 (DAI-canonical), and for of the utilization of the servers 0.08 (L) and 0.005 (DAI-canonical), respectively. The maximum tolerances of the mean job numbers in the nodes are 0.1 (L) and 0.003 (DAI-canonical).

Examples 1.2 (Tandem network) $K = 4,\ \nu_k = 3,\ k = 1, \ldots, K,\ \lambda_1 = 1,$ $\mu_1 = \mu_2 = \mu_4 = 1,\ \mu_3 = 2, 3, \ldots, 10$. Here we applied also DA iteration with center aggregation - center is node 3 (DAI-center). Maximum absolute errors of the macro probabilities: 0.6 (L), 0.12 (DAI-canonical), 0.02 (DAI-center), and of the utilization of the servers: 0.09 (L), 0.01 (DAI-canonical), 0.01 (DAI-center). Maximum tolerances of the mean job numbers: 0.25 (L), 0.05 (DAI-canonical), 0.01 (DAI-center).

2. Examples 2.1 The same tandem network as in examples 1.1 is investigated. The number of nodes and the capacities vary as follows: $K = 1, \ldots, 7$, capacity $\nu = 2, 3, 4$ in all nodes. The macro probabilities $r_k(0)$ and $r_k(\nu),\ k = 1, \ldots, K$, are calculated. The maximum absolute errors, maximum over the nodes, is fairly varying among the different models (different values K) when the diffusion approximation from the literature is applied, the overall maximum is 0.57. Applying DA iteration with canonical aggregation, it is 0.013.

3. We solved some queueing netwoks with the maximum entropy algorithm from the literature (L) and with DA iteration (DAI) applying canonical or center aggregation as appropriate.

Examples 3.1 The tandem network, $K = 2, \ldots, 7$ nodes, capacity $\nu = 3$ in all nodes, and also the service rates are equal in all nodes, varying between 2 and 5.

Examples 3.2 The tandem network, $K = 4,\ \nu = 3,\ \mu_1 = \mu_2 = \mu_4 = 1,\ \mu_3 = 2, 3, \ldots, 10$.

Examples 3.3 A merge topology: $K = 4$ nodes, $\nu_1 = 5,\ \nu_2 = 3,\ \nu_3 = 2,\ \nu_4 = 3,$
arrivals at nodes 2,3, and 4: $\lambda_2 = 1,\ \lambda_3 = 0.5,\ \lambda_4 = 0.5,$
transfer probabilities $t_{1,0} = t_{2,1} = t_{3,1} = t_{4,1} = 1,$
service rates $\mu_2 = \mu_3 = \mu_4 = 1,\ \mu_1 = 1.1, 1.2, \ldots, 2.5$.

Examples 3.4 A split topology. $K = 4$ nodes, $\nu_1 = 5,\ \nu_2 = 3,\ \nu_3 = 2,\ \nu_4 = 3,$
arrivals at node 1, $\lambda_1 = 0.1, 0.2, \ldots, 1.9,$
transfer probabilities $t_{1,2} = 0.2,\ t_{1,3} = 0.3,\ t_{1,4} = 0.5,\ t_{2,0} = t_{3,0} = t_{4,0} = 1,$

service rates $\mu_1 = 1,\ \mu_2 = 0.6,\ \mu_3 = 0.4,\ \mu_4 = 0.3$.

Again we give the maximum absolute errors of the macro probabilities and utilizations and the maximum tolerances of the mean job numbers, maximum over the nodes and over all examples 3: Probabilities of the job numbers in the nodes: 0.23 (L), 0.02 (DAI), utilization of the servers: 0.28 (L), 0.02 (DAI), mean job numbers in the nodes: 0.21 (L), 0.02 (DAI).

All examples were solved using DA iteration with canonical aggregation except examples 3.2. Here DA iteration with canonical aggregation gives the errors 0.12, 0.02, 0.046, respectively, so center aggregation was applied in order to gain the same order of accuracy as in the other examples.

4. These numerical examples are analyzed using DA iteration with canonical and center, paired, ACP, or ACT aggregation in order to demonstrate the merits of fine grain aggregation. We report the maximum absolute errors of the macro probabilities.

Examples 4.1 The tandem network: $K = 4$ nodes, capacity $\nu = 3$ in all nodes, arrivals at node 1, $\lambda_1 = 1$, service rates $\mu_1 = \mu_2 = \mu_4 = 1,\ \mu_3 = 1, 2, \ldots, 20$, transfer probabilities $t_{1,2} = t_{2,3} = t_{3,4} = t_{4,0} = 1$.

The maximum absolute error occurs when $\mu_3 = 20$. It is 0.13 using canonical aggregation, 0.02 using center aggregation, 0.022 using ACP aggregation, and if ACT aggregation is applied, which is even finer, it is 0.0023.

Examples 4.2 In the split network, examples 3.4, the maximum absolute error is 0.022 when canonical aggregation is used. It occurs for $\lambda_1 = 0.5$. When center aggregation is used, it is 0.002 and occurs for $\lambda_1 = 0.6$.

Examples 4.3 In the merge network from example 3.3, but with $\mu_2 = \mu_3 = 0.5$, the maximum absolute error is 0.016 when canonical aggregation is used. It occurs for $\mu_1 = 2.0$. When center aggregation is used, it is 0.0015 and occurs for $\mu_1 = 2.2$.

Examples 4.4 A symmetrical network: $K = 4$ nodes, capacities $\nu_1 = \nu_3 = 3,\ \nu_2 = \nu_4 = 2$,
arrivals at nodes 1 and 3, $\lambda_1 = \lambda_3 = 0.01, 0.02, \ldots, 0.2$,
transfer probabilities $t_{1,2} = t_{3,4} = 0.8,\ t_{1,3} = t_{3,1} = 0.2,\ t_{3,0} = t_{4,0} = 1$,
service rates $\mu_1 = \mu_3 = 0.5,\ \mu_2 = \mu_4 = 0.1$.

The maximum absolute error is 0.11 when canonical aggregation is used. It occurs for $\lambda_1 = \lambda_3 = 0.08$. When paired aggregation is used, $\mathcal{K}_2 = \{\,(1,2),\ (3,4)\,\}$, it is 0.011 and occurs also for $\lambda_1 = \lambda_3 = 0.08$.

5. We analyzed the following closed cyclic networks with the decomposition method from the literature (L) and using DA iteration with canonical aggregation (DAI-canonical) and with ACP aggregation (DAI-ACP):

Examples 5.1 $K = 4$ nodes, capacities $\nu_1 = 5,\ \nu_2 = 4,\ \nu_3 = 5,\ \nu_4 = 4$,
service rates $\mu_1 = 3,\ \mu_2 = 4,\ \mu_3 = 5,\ \mu_4 = 2$, N=5,...,9 jobs.

Examples 5.2 $K = 4$ nodes, capacities $\nu_1 = 2,\ \nu_2 = 4,\ \nu_3 = 3,\ \nu_4 = 4$,
service rates $\mu_1 = 2,\ \mu_2 = 1,\ \mu_3 = 3,\ \mu_4 = 4$, N=3,...,7 jobs.

Examples 5.3 $K = 5$ nodes, capacities $\nu_1 = 2,\ \nu_2 = 4,\ \nu_3 = 3,\ \nu_4 = 3,\ \nu_5 = 4$,
service rates $\mu_1 = 2,\ \mu_2 = 2,\ \mu_3 = 2,\ \mu_4 = 2,\ \mu_5 = 2$, N=3,...,8 jobs.

Examples 5.4 $K = 5$ nodes, capacities $\nu_1 = 2,\ \nu_2 = 4,\ \nu_3 = 3,\ \nu_4 = 3,\ \nu_5 = 4$,
service rates $\mu_1 = 5,\ \mu_2 = 3,\ \mu_3 = 4,\ \mu_4 = 6,\ \mu_5 = 2$, N=3,...,8 jobs.

The throughputs, the mean numbers of jobs in the nodes, and the macro probabilities were calculated. We give maximum tolerances and absolute errors, respectivly, maximum with respect to all nodes and all examples. Maximum tolerances of the throughputs: 0.015 (L), 0.0095 (DAI-canonical), 0.00033 (DAI-ACP), and of the mean job numbers: 0.158 (L), 0.043 (DAI-canonical), 0.0005 (DAI-ACP). Maximum absolute errors of the macro probabilities: 0.475 (L), 0.087 (DAI-canonical), 0.004 (DAI-ACP).

Example 6 This example gives an impression of the consumed computing time of DA iteration vs. vector iteration which calculates "exact" solutions. The symmetrical network of example 4.4, but with capacities $\nu_1 = \nu_3 = 12$, $\nu_2 = \nu_4 = 8$, has 13 689 states. The parameters $\lambda_1 = \lambda_3 = 0.32$, $\mu_1 = \mu_3 = 2$, $\mu_2 = \mu_4 = 0.4$ were used. On a SUN-4/330, vector iteration took 2 752 seconds, DA iteration with canonical aggregation 7 seconds, and with paired aggregation 99 seconds.

Example 7 The tandem topology, example 4.1, but with 9 nodes, $\mu_3 = 1$, and the capacity $\nu = 9$ in all nodes, which has 10^9 system states, was analyzed. On an Atari 1040ST, the computing time for DA iteration with canonical aggregation was 238 seconds. This demonstrates that Markov models with a really large state space can be treated with the DA iteration technique.

Conclusion

The DA iteration technique is well suited for the analysis of queueing networks with repetitive-service blocking, but, in fact, it is much more general, applicable to many Markov models. Its time and space demands are higher than for many special purpose heuristics from the literature, but are by orders of magnitude lower than for exact solutions of Markov chains. Hence Markov models can be treated with DA iteration whose exact analysis is impossible due to time and space complexity. Further work has to be done about the approximation error estimates.

Acknowledgements

The authors would like to express their gratitude to the unknown referees for their valuable suggestions.

References

[1] Ian F. Akyildiz and Horst von Brand. Exact solutions for open, closed, and mixed queueing networks with rejection blocking. *Theoretical Computer Science*, 64:203–219, 1989.

[2] B. Bärk. Aggregationsmethoden für Markovketten. 1997.

[3] P. Caseau and G. Pujolle. Throughput capacity of a sequence of queues with blocking due to finite waiting room. *IEEE Transactions on Software Engineering*, 5(6):631–642, 1979.

[4] Y. Frein and Y. Dallery. Analysis of Cyclic Queueing Networks with Finite Buffers and Blocking Before Service. *Performance Evaluation*, 10:197–210, 1989.

[5] W. K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In *Numerical Solution of Markov Chains*, pages 357–372. Marcel Dekker, New York, Basel, Hong Kong, 1991.

[6] V. Jonas. Iterative Analyse von Blockiernetzen mit alternierender Aggregation und Disaggregation. Master's thesis, Universität Bonn, 1992.

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

[8] D. Kouvatsos and S. Denazis. Comments on and tuning to: Mem for arbitrary queueing networks with repetitive-service blocking under random routing. Research Report CS-18-91, Bradfod, 1991.

[9] D. Kouvatsos and N. P. Xenios. Mem for arbitrary queueing networks with multiple general servers and repetitive-service blocking. *Performance Evaluation*, 10:169–195, 1989.

[10] J. Labetoulle and G. Pujolle. Isolation method in a network of queues. *IEEE Transactions on Software Engineering*, 6(6):373–380, 1980.

[11] R. O. Onvural. Survey of closed queueing networks with blocking. *ACM Computing Surveys*, 22(2):83–121, 1990.

[12] H. G. Perros. Approximation algorithms for open queueing networks with blocking. In Takagi, editor, *Stochastic Analysis of Computer and Communication*, pages 451–498. North-Holland, 1990.

[13] J. Ch. Strelen. Iterative Analyse von Markov-Modellen mit alternierender Aggregation und Disaggregation. In A. Lehmann and F. Lehmann, editors, *Messung, Modellierung und Bewertung von Rechensystemen IFB 286*, pages 320–336. Springer, Berlin, 1991.

[14] J. Ch. Strelen. Approximate product form solutions for Markov chains. *Performance Evaluation*, 30:87–110, 1997.