The Power-Series Algorithm for Markovian Queueing Networks

W.B. van den Hout *
J.P.C. Blanc
Tilburg University, Faculty of Economics
P.O.Box 90153, 5000 LE Tilburg, The Netherlands

Abstract: A new version of the Power-Series Algorithm is developed to compute the steady-state distribution of a rich class of Markovian queueing networks. The arrival process is a Multi-queue Markovian Arrival Process, which is a multi-queue generalization of the *BMAP*. It includes Poisson, fork and round-robin arrivals. At each queue the service process is a Markovian Service Process, which includes sequences of phase-type distributions, set-up times and multi-server queues. The routing is Markovian. The resulting queueing network model is extremely general, which makes the Power-Series Algorithm a useful tool to study load-balancing, capacity-assignment and sequencing problems.

1 Introduction

Networks of queues without product-form solution are usually difficult to analyze, both analytically and numerically. For Markovian networks, the steady-state distribution is determined by the set of balance equations, but because of the size of the multi-dimensional state space any numerical method to solve these equations is inevitably memory and time consuming. The Power-Series Algorithm (*PSA*) aims to be an efficient way to solve the balance equations. The advantage of the *PSA* over other methods is that techniques like Padé-approximation can be used to extrapolate the power series, and that the behaviour of the power-series can be studied to assess the credibility of the results.

Networks of queues will be considered with unbounded queue sizes. Customers arrive according to a Multi-queue Markovian Arrival Process (*MMAP*), which is a multi-queue generalization of the Batch Markovian Arrival Process (*BMAP*) introduced by Lucantoni [**?**]. On top of the ability of the *BMAP* to model dependencies between interarrival times and batch sizes, the *MMAP* can also model all kinds of dependencies between arrivals at the different queues, like fork and round-robin arrivals. At each queue the service process is a Markovian Service Process (*MSP*).

*The investigations were supported in part by the Netherlands Foundation for Mathematics SMC with financial aid from the Netherlands Organization for the Advancement of Scientific Research (NWO).

This includes for example set-up times, sequences of phase-type distributions and multi-server queues. The routing of customers is Markovian, which includes a large variety of network structures (like the class of Jackson networks, a small subclass of the networks considered here). The extreme generality of the networks contained in this general framework makes the analysis below a useful tool to study load-balancing, capacity-assignment and sequencing problems.

The basic idea of the *PSA* is like a homotopy: the transition rates of the original network are transformed with a parameter γ, such that for $\gamma = 1$ the transformed network is the original network and the asymptotic network for γ in a neighbourhood of $\gamma = 0$ is easy to analyze. Then the information from the problem near $\gamma = 0$ can be used to solve the problem at $\gamma = 1$. The basic idea of the *PSA* stems from Keane (see [**?**]). It has been applied to queueing models with queues in parallel [**?**, **?**], the shortest-queue model [**?**], various polling models [**?**, **?**], and the *BMAP/PH/1* queue [**?**]. For an overview, see [**?**]. For all these models only the arrival process needed to be transformed and the transformation parameter γ could be interpreted as the load of the system. Unfortunately, this procedure is only possible for feedforward networks. For non-feedforward networks, sets of equations would have to be solved with a size that rapidly increases with each step of the algorithm. Koole [**?**] suggests to prevent this by treating the queues asymmetrically. The approach that will be used in the present paper, is to transform the routing process also. In both approaches, the parameter γ no longer has a clear interpretation. This could be overcome by using more than one transformation parameter. For example, a parameter ρ could be used to transform the arrival process, and a parameter σ for the routing process. However, using several parameters leads to power-series expansions in more than one variable. This implies that more coefficients need to be calculated and that multi-dimensional Padé-approximants are required. For this reason, only a single parameter γ will be used here.

In section 2, the network model is introduced. In section 3, the algorithm to calculate the steady-state distribution and moments is described. In section 4, two examples are given. The first considers the optimal order of queues in series. The second shows that for cyclic open networks with symmetric arrivals and equal loads the expected total number of customers is mainly determined by the sum of the second moments of the service-time distributions.

2 The Network Model

The number of queues is S. Unless indicated otherwise, the following notation is used. Vectors are column vectors and written in bold face. The vector $\boldsymbol{e}$ is a vector of ones, $\boldsymbol{0}$ and $\boldsymbol{e}_0$ are vectors of zeros, $\boldsymbol{e}_s$ are the unit vectors of size S for $1 \leq s \leq S$ and $\boldsymbol{e}_{S+1} = \boldsymbol{e}_1$. For any vector $\boldsymbol{n}$, define $|\boldsymbol{n}| = \boldsymbol{e}^T \boldsymbol{n}$. Matrices are written in capitals. The matrix O is a matrix of zeros and I_ℓ a unit matrix of size ℓ. The operator $\otimes$ denotes the Kronecker product of two matrices; the operator $\odot$ the Hadamard (or element-wise) product of two matrices of equal size.

In the examples described below, a phase-type distribution has generator T, as initial distribu-

tion the row vector $\boldsymbol{\alpha}$ and $T^0 = -T\boldsymbol{e}$ (conform Neuts [**?**], but without probability mass at zero). The class of phase-type distributions includes the Erlang and hyperexponential distributions as well as finite mixtures of these.

2.1 Multi-queue Markovian Arrival Process

The arrival process is a Multi-queue Markovian Arrival Process. It has an underlying irreducible Markov process with J_0 states. In this underlying process, a transition $j \to h$ is made with rate α_{jh} $(1 \le j,h \le J_0 < \infty)$. The set of possible batch arrivals is $\{\boldsymbol{b}_m | 0 \le m \le M\}$, with $\boldsymbol{b}_0 = \boldsymbol{0}$ and $\boldsymbol{b}_m \in \mathbb{N}^{\mathrm{S}} \backslash \{\boldsymbol{0}\}$ for $1 \le m \le M \le \infty$. A transition $j \to h$ in the underlying process causes an arrival of batch $\boldsymbol{b}_m$ with probability q_{mjh}.

$$\begin{array}{lcllcl} A & = & \{\alpha_{jh}\}, & \bar{A} & = & \mathrm{diag}(A\boldsymbol{e}), \\ Q_m & = & \{q_{mjh}\}, & \sum_{m=0}^{M} Q_m & = & \boldsymbol{e}\boldsymbol{e}^T, \\ A_m & = & A \odot Q_m, & \sum_{m=0}^{M} A_m & = & A. \end{array}$$

The pure $MMAP$ $\{(\boldsymbol{\mathcal{N}}_t^{MMAP}, \mathcal{J}_t),\ t \ge 0\}$ on state space $\mathbb{N}^{\mathrm{S}} \times \{1, \ldots, \mathrm{J}_0\}$ is identical to the $BMAP$ if $S = 1$ and $\boldsymbol{b}_m = m$ ($0 \le m \le \infty$. It then has generator

$$Q^{MMAP} = \begin{pmatrix} A_0 - \bar{A} & A_1 & A_2 & \cdots \\ O & A_0 - \bar{A} & A_1 & \cdots \\ O & O & A_0 - \bar{A} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Lucantoni [**?**] lists a number of special cases of the $BMAP$, like the Poisson process, Markov-modulated Poisson processes, PH-renewal processes and processes with correlated batch arrivals. If each queue has an independent $BMAP$, this can be modelled as a $MMAP$. Other special cases of $MMAP$s are:

1) Poisson arrivals: independent Poisson arrivals with rate λ_s at queue s:

$$\begin{array}{llll} M = S, & A_0 - \bar{A} = -\sum_{m=1}^{M} \lambda_m, & & \\ \boldsymbol{b}_m = \boldsymbol{e}_m, & A_m = \lambda_m, & & \text{for } 1 \le m \le M. \end{array}$$

2) Round-robin arrivals: an arrival at queue s is followed by an arrival at queue $s+1$ with the interarrival time exponentially distributed with rate λ_s:

$$\begin{array}{llll} M = S, & A_0 - \bar{A} = -\mathrm{diag}(\lambda), & & \\ \boldsymbol{b}_m = \boldsymbol{e}_m, & A_m = \lambda_m \boldsymbol{e}_m \boldsymbol{e}_{m+1}^T, & & \text{for } 1 \le m \le M. \end{array}$$

3) Fork arrivals: simultaneous arrivals at each queue with phase-type interarrival times:

$$\begin{array}{ll} M = 1, & A_0 - \bar{A} = T, \\ \boldsymbol{b}_1 = \boldsymbol{e}, & A_1 = T^0 \boldsymbol{\alpha}. \end{array}$$

2.2 Markovian Service Process

The service processes at all queues are independent Markovian Service Processes. A *MSP* has an underlying Markov process with J states, and the transition rates are allowed to depend on the number of customers n at that queue. A transition $j \to h$ is made with rate $\beta_{jh}(n)$ and such a transition causes a service completion of ℓ customers with probability $r_{\ell jh}(n)$ ($1 \le j,h \le J < \infty$; $0 \le \ell \le n \le \infty$).

$$\begin{array}{lcllcl} B(n) & = & \{\beta_{jh}(n)\}, & \bar{B}(n) & = & \mathrm{diag}(B(n)\boldsymbol{e}), \\ R_\ell(n) & = & \{r_{\ell jh}(n)\}, & \sum_{\ell=0}^{n} R_\ell(n) & = & \boldsymbol{e}\boldsymbol{e}^T, \\ B_\ell(n) & = & B(n) \odot R_\ell(n), & \sum_{\ell=0}^{n} B_\ell(n) & = & B(n). \end{array}$$

A pure Markovian Service Process $\{(\mathcal{N}_t^{MSP}, \mathcal{J}_t),\ t \ge 0\}$ on state space $\mathbb{N} \times \{1, \ldots, \mathrm{J}\}$ has generator

$$Q^{MSP} = \begin{pmatrix} B_0(0) - \bar{B}(0) & O & O & \cdots \\ B_1(1) & B_0(1) - \bar{B}(1) & O & \cdots \\ B_2(2) & B_1(2) & B_0(2) - \bar{B}(2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

In this paper, a number of assumptions is made about this service process. First, it is assumed that all non-empty states are transient, so from any initial state the empty states will eventually be reached. Furthermore, it is assumed that when the *MSP* reaches the empty states, it returns to state j with probability $\boldsymbol{\phi}_j$, where it remains. For this it is sufficient that

$$B_\ell(\ell) = B_\ell(\ell)\boldsymbol{e}\boldsymbol{\phi}^T, \quad B_\ell(0) = O, \quad \text{for } \ell \ge 0.$$

For a queue this implies that at the end of each busy period the *MSP* returns to state j with probability $\boldsymbol{\phi}_j$, where it remains until the next arrival at the queue. Finally, it is assumed that customers are not served in batches:

$$B_\ell(n) = O, \quad \text{for } \ell \ge 2.$$

This assumption is not essential, but is made because otherwise a more complicated routing process needs to be defined and notation would be more involved. In the examples of *MSP*s listed below, the vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are the unit vectors of size 2.

1) Independent phase-type service-time distributions:

$$\begin{array}{ll} B_0(n) - \bar{B}(n) = T, & \text{for } n \ge 1, \\ B_1(n) = T^0\boldsymbol{\alpha}, & \text{for } n \ge 1, \\ \boldsymbol{\phi} = \boldsymbol{\alpha}^T. & \end{array}$$

2) As 1), but with set-up: after each idle period the first service time has initial distribution $\boldsymbol{\alpha}_1$, all other service times have initial distribution $\boldsymbol{\alpha}_2$:

$$\begin{aligned} &B_0(n) - \bar{B}(n) = T, && \text{for } n \geq 1, \\ &B_1(1) = T^0\boldsymbol{\alpha}_1, \\ &B_1(n) = T^0\boldsymbol{\alpha}_2, && \text{for } n \geq 2, \\ &\boldsymbol{\phi} = \boldsymbol{\alpha}_1^T. \end{aligned}$$

Any pair of phase-type distributions $(\tilde{T}_1, \tilde{\boldsymbol{\alpha}}_1)$ and $(\tilde{T}_2, \tilde{\boldsymbol{\alpha}}_2)$ can be modeled by a single generator T with two different initial distributions $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, by taking T block diagonal: $T = \boldsymbol{e}_1\boldsymbol{e}_1^T \otimes \tilde{T}_1 + \boldsymbol{e}_2\boldsymbol{e}_2^T \otimes \tilde{T}_2, \boldsymbol{\alpha}_1 = \boldsymbol{e}_1 \otimes \tilde{\boldsymbol{\alpha}}_1$ and $\boldsymbol{\alpha}_2 = \boldsymbol{e}_2 \otimes \tilde{\boldsymbol{\alpha}}_2$.

Examples 1 and 2 are special cases of sequences of phase-type distributions $\{(T_\ell, \boldsymbol{\alpha}_\ell),\ \ell \geq 1\}$. Because the number of phases J of the *MSP* is finite, such sequences must, after a number of set-up distributions, start repeating itself, either in a deterministic or in a probabilistic sense. Because the *MSP* starts anew at the beginning of each busy period, also mixtures of sequences are possible. This could be used to model for example a situation where at the beginning of each busy period, either a fast or a slow server is chosen. Other examples of *MSP*s are multi-server queues:

3) c identical exponential servers with rate μ:

$$\begin{aligned} &B_0(n) - \bar{B}(n) = -\mu \min\{c, n\}, && \text{for } n \geq 1, \\ &B_1(n) = \mu \min\{c, n\}, && \text{for } n \geq 1, \\ &\boldsymbol{\phi} = 1. \end{aligned}$$

4) c identical phase-type servers:

$$\begin{aligned} &B_0(n) - \bar{B}(n) = \textstyle\sum_{s=1}^{n} I_{s-1} \otimes T \otimes I_{c-s}, && \text{for } 1 \leq n \leq c, \\ &B_0(n) - \bar{B}(n) = \textstyle\sum_{s=1}^{c} I_{s-1} \otimes T \otimes I_{c-s}, && \text{for } c < n, \\ &B_1(n) = [(\textstyle\sum_{s=1}^{n} I_{s-1} \otimes T^0 \otimes I_{n-s})(I_{n-1} \otimes \boldsymbol{\alpha})] \otimes I_{c-n}, && \text{for } 1 \leq n \leq c, \\ &B_1(n) = \textstyle\sum_{s=1}^{c} I_{s-1} \otimes T^0\boldsymbol{\alpha} \otimes I_{c-s}, && \text{for } c < n, \\ &\boldsymbol{\phi} = (\boldsymbol{\alpha} \otimes \ldots \otimes \boldsymbol{\alpha})^T. \end{aligned}$$

Here, I_s is a unit-matrix with size ℓ^s for $0 \leq s \leq c$, where ℓ is the number of phases of the phase-type distribution. The transitions are defined such that if there are no waiting customers in the queue ($n \leq c$), then the first n servers are active and the other $c-n$ servers are idle; when server s completes service, then the customers at servers $s+1, \ldots, n$ move to servers $s, \ldots, n-1$, continuing service in the same phase. Server n becomes idle, with the service phase distributed according to $\boldsymbol{\alpha}$. This way, no variables need to be added to keep track of which servers are active, and when a new customer arrives, service can be started without changing the state of the *MSP*. With non-identical servers this is not possible.

2.3 Markovian Routing Process

The routing is Markovian: after service completion at queue s the customer joins queue t with probability χ_{st} and leaves the network with probability χ_{s0} ($1 \leq s,t \leq S$).

$$X = \{\chi_{st}\}, \quad \boldsymbol{\chi}_0 = \{\chi_{s0}\}, \quad X\boldsymbol{e} + \boldsymbol{\chi}_0 = \boldsymbol{e}.$$

2.4 Markovian Network Process

The above described arrival, service and routing processes determine the network process $\{(\boldsymbol{\mathcal{N}}_t, \boldsymbol{\mathcal{J}}_t),\ t \geq 0\}$ on state space

$$\Omega = \left\{ (\boldsymbol{n}, \boldsymbol{j}) \,\middle|\, \boldsymbol{n} \in \mathbb{N}^{\mathrm{S}}, 1 \leq \mathrm{j_s} \leq \mathrm{J_s} \text{ for } 0 \leq \mathrm{s} \leq \mathrm{S} \right\}.$$

The state $(\boldsymbol{n}, \boldsymbol{j}) \in \Omega$ denotes that there are n_s customers at queue s, the arrival process is in state j_0 and the service process at queue s is in state j_s ($1 \leq s \leq S$). To introduce matrix notation, it is convenient to map the $(2S+1)$-dimensional state space Ω onto the $(S+1)$-dimensional state space

$$\bar{\Omega} = \left\{ (\boldsymbol{n}, i) \,\middle|\, \boldsymbol{n} \in \mathbb{N}^{\mathrm{S}}, 1 \leq \mathrm{i} \leq \mathrm{I} \right\},$$

where $I = J_0 \times \ldots \times J_S$. This can be done 'lexicographically' with the mapping

$$i(\boldsymbol{j}) = 1 + \sum_{s=0}^{S} (j_s - 1)\bar{J}_{s+1},$$

where $\bar{J}_s = J_s \times \ldots \times J_S$ for $0 \leq s \leq S$ and $\bar{J}_S + 1 = 1$. The reverse mapping is

$$j_s(i) = 1 + \left[(i-1) \bmod \bar{J}_s \right] \operatorname{div} \bar{J}_{s+1}, \quad \text{for } 0 \leq s \leq S,$$

This mapping determines the network process $\{(\boldsymbol{\mathcal{N}}_t, \mathcal{I}_t),\ t \geq 0\}$ on state space $\bar{\Omega}$. If the network is stable, the steady-state probabilities of this process

$$P_i(\boldsymbol{n}) = \lim_{t \to \infty} \Pr\left\{ (\boldsymbol{\mathcal{N}}_t, \mathcal{I}_t) = (\boldsymbol{n}, i) \right\}$$

exist for all $(\boldsymbol{n}, i) \in \bar{\Omega}$. They are independent of the initial state $(\boldsymbol{\mathcal{N}}_0, \mathcal{I}_0)$ and uniquely determined by the balance and normalization equations. For any matrix A, let double brackets denote the Kronecker product

$$[\![A]\!]_s = I_{J_0 \times \ldots \times J_{s-1}} \otimes A \otimes I_{J_{s+1} \times \ldots \times J_S}, \quad \text{for } 0 \leq s \leq S.$$

Then the balance equations are

$$\begin{aligned} \left\{ [\![\bar{A} - A_0^T]\!]_0 + \sum_{s=1}^{S} [\![\bar{B}_s(n_s) - B_{s0}^T(n_s) - \chi_{ss} B_{s1}^T(n_s)]\!]_s \right\} P(\boldsymbol{n}) \\ = \sum_{m=1}^{M} [\![A_m^T]\!]_0 \quad P(\boldsymbol{n} - \boldsymbol{b}_m) \\ + \sum_{s=1}^{S} \sum_{\substack{t=0 \\ t \neq s}}^{S} \chi_{st} \quad [\![B_{s1}^T(n_s+1)]\!]_s \quad P(\boldsymbol{n} + \boldsymbol{e}_s - \boldsymbol{e}_t), \end{aligned}$$

for $\boldsymbol{n} \in \mathbb{N}^S$, with $P(\boldsymbol{n}) = \boldsymbol{0}$ for $\boldsymbol{n} \notin \mathbb{N}^S$. The matrices A_0 and $B_{s0}(n_s)$ in the left-hand side correspond to changes in the arrival and service processes without arrival or service completion, and $\chi_{ss}B_{s1}(n_s)$ with the event that a customer joins the same queue again, which does not change the queue lengths. The first expression in the right-hand side corresponds to an arrival and the second with a service completion followed by either a departure from the network ($t = 0$) or a transition to another queue ($t \neq 0, s$).

3 The Power-Series Algorithm

The arrival and routing process of the network described above will be transformed with a transformation parameter γ in [0,1], in such a way that for $\gamma = 1$ the transformed network process is equal to the original network process. This will be done is such a way that the steady-state probabilities as function of γ are analytic at $\gamma = 0$ and the coefficients of the steady-state probabilities can be calculated recursively.

Let $r_m = |\boldsymbol{b}_m|$ denote the number of customers in batch $\boldsymbol{b}_m$, for $1 \leq m \leq M$. Replace the probability matrices Q_m by

$$\begin{aligned} Q_m(\gamma) &= \gamma^{r_m} Q_m, \quad \text{for } 1 \leq m \leq M, \\ Q_0(\gamma) &= Q_0 + \sum_{m=1}^{M} (1 - \gamma^{r_m}) Q_m = \boldsymbol{e}\boldsymbol{e}^T - \sum_{m=1}^{M} \gamma^{r_m} Q_m. \end{aligned}$$

The probability of an arrival of r customers is multiplied by γ^r, and the remaining probability mass is added to the probability of no arrival, so $\sum_{m=0}^{M} Q_m(\gamma) = \boldsymbol{e}\boldsymbol{e}^T$ for $\gamma \in [0, 1]$ and $Q_m(1) = Q_m$ for $1 \leq m \leq M$. For smaller γ less arrivals occur on average and for $\gamma = 0$ no arrivals occur at all.

Let X_d denote the diagonal matrix with the same diagonal as the routing matrix X, and X_o the off-diagonal part of X, so $X_d + X_o = X$. In the transformed network process, the routing probabilities X and $\boldsymbol{\chi}_0$ are replaced by

$$X(\gamma) = X_d + \gamma X_o, \quad \boldsymbol{\chi}_0(\gamma) = \gamma \boldsymbol{\chi}_0 + (1 - \gamma)(I - X_d)\boldsymbol{e}.$$

The probability to go from queue s to queue t, with $t \neq s$, is multiplied by γ, and the remaining probability mass is added to the probability to leave the network, so $X(\gamma)\boldsymbol{e} + \boldsymbol{\chi}_0(\gamma) = \boldsymbol{e}$ for $\gamma \in [0, 1]$ and $X(1) = X$, $\boldsymbol{\chi}_0(1) = \boldsymbol{\chi}_0$. For smaller γ, the customers on average visit less queues, because after each service completion they leave the network with higher probability. For $\gamma = 0$, customers only visit a single station, possibly several times.

The arrival rates at the queues from outside the network are equal to

$$\boldsymbol{\lambda}(\gamma) = \sum_{m=1}^{M} \gamma^{r_m} \boldsymbol{b}_m \boldsymbol{\xi}^T A_m \boldsymbol{e},$$

where $\boldsymbol{\xi}$ is the steady-state distribution of the Markov process underlying the *MMAP*, which can be calculated from $(\bar{A} - A^T)\boldsymbol{\xi} = \mathbf{0}, \boldsymbol{e}^T\boldsymbol{\xi} = 1$. The arrival rates both from outside the network and from the other queues are equal to

$$\boldsymbol{\nu}(\gamma) = \left[I - X^T(\gamma)\right]^{-1} \boldsymbol{\lambda}(\gamma) = \left\{\sum_{r=0}^{\infty} \gamma^r \left[(I - X_d)^{-1} X_o^T\right]^r\right\} (I - X_d)^{-1}\boldsymbol{\lambda}(\gamma).$$

This power series converges and since it has only non-negative coefficients, $\boldsymbol{\nu}(\gamma)$ is increasing in γ: for larger γ there are more arrivals and customers leave the network less often. The service process at each queue does not depend on γ. From this it is easily seen that if the original network is stable, the transformed network is also stable for all γ in [0,1], and the steady-state probabilities are, up to a constant, uniquely determined by the balance equations. These can be rearranged into

$$\begin{aligned}
&\left\{\left[\!\left[\bar{A} - A^T\right]\!\right]_0 + \sum_{s=1}^{S}\left[\!\left[\bar{B}_s(n_s) - B_{s0}^T(n_s) - \chi_{ss}B_{s1}^T(n_s)\right]\!\right]_s\right\} P_\gamma(\boldsymbol{n}) \\
&\quad = \sum_{m=1}^{M} \gamma^{r_m} \left[\!\left[A_m^T\right]\!\right]_0 \{P_\gamma(\boldsymbol{n} - \boldsymbol{b}_m) - P_\gamma(\boldsymbol{n})\} \\
&\quad + \sum_{s=1}^{S}\sum_{\substack{t=1 \\ t\neq s}}^{S} \gamma\chi_{st} \left[\!\left[B_{s1}^T(n_s+1)\right]\!\right]_s \{P_\gamma(\boldsymbol{n} + \boldsymbol{e}_s - \boldsymbol{e}_t) - P_\gamma(\boldsymbol{n} + \boldsymbol{e}_s)\} \\
&\quad + \sum_{s=1}^{S} (1 - \chi_{ss}) \left[\!\left[B_{s1}^T(n_s+1)\right]\!\right]_s P_\gamma(\boldsymbol{n} + \boldsymbol{e}_s),
\end{aligned} \tag{1}$$

for $\boldsymbol{n} \in \mathbb{N}^S$. Clearly, the steady-state probabilities are functions of γ. Because arrivals of batches of size r have a rate that is $O(\gamma^r)$, for $\gamma \downarrow 0$, the steady-state probabilities satisfy

$$P_\gamma(\boldsymbol{n}) = O\left(\gamma^{|\boldsymbol{n}|}\right), \quad \text{for } \gamma \downarrow 0, \boldsymbol{n} \in \mathbb{N}^S. \tag{2}$$

Notice that, for $\gamma = 0$, only the empty states have non-zero probability, because there are departures from the network but no arrivals. In a future paper several statements about convergence and analyticity in the present paper will be proved for a much wider class of Markov processes, among others that the steady-state probabilities are analytic functions of γ, in a neighbourhood of $\gamma = 0$, so they can be represented by their power-series expansions:

$$P_\gamma(\boldsymbol{n}) = \sum_{r=|\boldsymbol{n}|}^{\infty} \gamma^r U_r(\boldsymbol{n}), \quad \text{for } \boldsymbol{n} \in \mathbb{N}^s. \tag{3}$$

The transformed network process is such that the coefficient vectors $U_r(\boldsymbol{n})$ of these power-series expansions can be calculated recursively by the *PSA*. This will be shown first for the empty states, and then for the non-empty states.

The process underlying the *MMAP* is not influenced by the queue-length and the service processes. Summing the steady-state probabilities of the network over all possible queue lengths

and states of the service processes must therefore render the steady-state distribution of the arrival process:

$$\left[I_{J_0} \otimes \boldsymbol{e}^T\right] \sum_{\boldsymbol{n} \in \mathbb{N}^S} P_\gamma(\boldsymbol{n}) = \boldsymbol{\xi},$$

where $\boldsymbol{e}$ is a vector of ones with size $J_1 \times \ldots \times J_S$. When the network is empty, the states of the service processes at the queues are distributed according to the initial distributions $\boldsymbol{\phi}_s$:

$$P_\gamma(\boldsymbol{0}) = \left[I_{J_0} \otimes \boldsymbol{e}^T\right] P_\gamma(\boldsymbol{0}) \otimes \boldsymbol{\phi},$$

where $\boldsymbol{\phi} = \boldsymbol{\phi}_1 \otimes \ldots \otimes \boldsymbol{\phi}_S$. Combining both renders

$$P_\gamma(\boldsymbol{0}) = \left\{ \boldsymbol{\xi} - \left[I_{J_0} \otimes \boldsymbol{e}^T\right] \sum_{\boldsymbol{n} > \boldsymbol{0}} P_\gamma(\boldsymbol{n}) \right\} \otimes \boldsymbol{\phi}.$$

Inserting the power-series expansions (**??**) and equating the coefficients of corresponding powers of γ on either side of the equality sign shows that the coefficients of the expansions of the empty states $P_\gamma(\boldsymbol{0})$ satisfy

$$\begin{aligned} U_0(\boldsymbol{0}) &= \boldsymbol{\xi} \otimes \boldsymbol{\phi}, \\ U_r(\boldsymbol{0}) &= -\left\{ \left[I_{J_0} \otimes \boldsymbol{e}^T\right] \sum_{0 < |\boldsymbol{n}| \leq r} U_r(\boldsymbol{n}) \right\} \otimes \boldsymbol{\phi}, \quad \text{for } r \geq 1. \end{aligned} \tag{4}$$

Notice that $\boldsymbol{e}^T U_0(\boldsymbol{0}) = 1$, so for $\gamma = 0$ all probability mass is at the empty states.

Inserting the power-series expansions (**??**) into the balance equations (**??**) and equating the coefficients of corresponding powers of γ on either side of the equality sign, shows that the coefficients of the power-series expansions of the non-empty states satisfy the following recurrence relations:

$$\begin{aligned} &\left\{ \left[\!\left[\bar{A} - A^T \right]\!\right]_0 + \sum_{s=1}^{S} \left[\!\left[\bar{B}_s(n_s) - B_{s0}^T(n_s) - \chi_{ss} B_{s1}^T(n_s) \right]\!\right]_s \right\} U_r(\boldsymbol{n}) \\ &\quad = \sum_{m=1}^{M} \left[\!\left[A_m^T \right]\!\right]_0 \{U_{r-r_m}(\boldsymbol{n} - \boldsymbol{b}_m) - U_{r-r_m}(\boldsymbol{n})\} \\ &\quad + \sum_{s=1}^{S} \sum_{\substack{t=1 \\ t \neq s}}^{S} \chi_{st} \left[\!\left[B_{s1}^T(n_s + 1) \right]\!\right]_s \{U_{r-1}(\boldsymbol{n} + \boldsymbol{e}_s - \boldsymbol{e}_t) - U_{r-1}(\boldsymbol{n} + \boldsymbol{e}_s)\} \\ &\quad + \sum_{s=1}^{S} (1 - \chi_{ss}) \left[\!\left[B_{s1}^T(n_s + 1) \right]\!\right]_s U_r(\boldsymbol{n} + \boldsymbol{e}_s), \end{aligned} \tag{5}$$

for $\boldsymbol{n} \in \mathbb{N}^S, r \geq |\boldsymbol{n}|$. The matrix in the left-hand side,

$$\left[\!\left[\bar{A} - A^T \right]\!\right]_0 + \sum_{s=1}^{S} \left[\!\left[\bar{B}_s(n_s) - B_{s0}^T(n_s) - \chi_{ss} B_{s1}^T(n_s) \right]\!\right]_s,$$

is invertible for all $\boldsymbol{n} \in \mathbb{N}^S \backslash \{\boldsymbol{0}\}$. The coefficients $U_{\tilde{r}}(\tilde{\boldsymbol{n}})$ in the right-hand side either have $\tilde{r} < r$ or have $\tilde{r} = r$ and $|\tilde{\boldsymbol{n}}| > |\boldsymbol{n}|$. All coefficients $U_{\tilde{r}}(\tilde{\boldsymbol{n}})$ with $|\tilde{\boldsymbol{n}}| > \tilde{r}$ are zero because of the order

property (??). Together, this implies that the coefficients of the expansions of the steady-state probabilities up to the R-th power of γ can be calculated recursively, for increasing values of r and, for each fixed r, for decreasing values of $|\boldsymbol{n}|$, starting with $|\boldsymbol{n}| = r$:

Power-Series Algorithm
Calculate $U_{0,0}$ from (??)
for $r = 1, \ldots, R$ do
  for $N = r, \ldots, 1$ do
    for all $\boldsymbol{n} \in \mathbb{N}^S$ with $|\boldsymbol{n}| = N$ do
      Calculate $U_r(\boldsymbol{n})$ from (??)
  Calculate $U_r(\boldsymbol{0})$ from (??)

Usually one is not so much interested in the steady-state probabilities, but more in moments of the process. The expansions of moments can be obtained from the expansions of the steady-state probabilities:

$$\lim_{t\to\infty} \mathrm{E}_\gamma \{f(\boldsymbol{\mathcal{N}}_t, \mathcal{I}_t)\} = \sum_{(\boldsymbol{n},i)\in\bar{\Omega}} f(\boldsymbol{n},i) P_{\gamma i}(\boldsymbol{n}) = \sum_{r=0}^{\infty} \gamma^r \sum_{(\boldsymbol{n},i)\in\bar{\Omega}:\ \boldsymbol{e}^T\boldsymbol{n}\le r} f(\boldsymbol{n},i) U_{ri}(\boldsymbol{n}),$$

for functions $f : \bar{\Omega} \to \mathbb{R}$. Examples are

$f(\boldsymbol{n},i) = |\boldsymbol{n}|$, the expected total number of customers in the network,
$f(\boldsymbol{n},i) = n_s^t$, the t-th moment of the queue length at queue s,
$f(\boldsymbol{n},i) = n_s n_t$, the cross-product of the queue lengths at queues s and t.

The storage requirements of the algorithm can be substantially reduced if the maximal batch size $\bar{r} = sup_m r_m$ is finite. From (??) it can be seen that in step r of the algorithm, coefficients $U_{\tilde{r}}(\tilde{\boldsymbol{n}})$ with $\tilde{r} < r - \bar{r}$ are then no longer needed to calculated the remaining coefficients.

The *MSP* at queue s depends only on the queue length at queue s. The state dependence could be made more general, not only for the service processes but also for the arrival and routing processes. The state dependence of the service and routing processes must be such that, for $\gamma = 0$, all non-empty states of the transformed network process are transient, so eventually the empty states are reached. Then the service processes must be stopped and the distribution $\boldsymbol{\phi}$ over the service-phases must be known (but $\boldsymbol{\phi}$ need not be the Kronecker product $\boldsymbol{\phi}_1 \otimes \ldots \otimes \boldsymbol{\phi}_S$). The Markov process underlying the *MMAP* must be state independent to calculate $\boldsymbol{\xi}$, but the probability matrices Q_m can be state dependent. This way, the coefficients of the empty states can still be calculated from (??) and the coefficients of the non-empty states can be calculated from (??) if the parameters are replaced by the state dependent parameters.

Suppose that by the algorithm described above, for either probabilities or moments, the coefficients $\{v_r,\ 0 \le r \le R\}$ are obtained. To compute these first R coefficients the number of coefficients of the state probabilities that need to be calculated is

$$\#\{(r,\boldsymbol{n},i) \in \mathbb{N} \times \bar{\Omega} \mid |\boldsymbol{n}| \le \mathrm{r} \le \mathrm{R}\ \} = I \times \binom{R+S+1}{S+1}.$$

Because this number grows fast in R, it will be obvious that it is worthwhile to obtain as much information from the coefficients $\{v_r,\ 0 \le r \le R\}$ as possible. That is why techniques to make series converge or converge faster form an essential part of the *PSA*. The epsilon algorithm and bilinear mapping will shortly be discussed. For a more thorough discussion, see [**?**]. Define the partial and infinite sum

$$V_R(\gamma) = \sum_{r=0}^{R} \gamma^r v_r, \quad V(\gamma) = \lim_{R\to\infty} V_R(\gamma).$$

Since for $\gamma = 1$ the transformed network is equal to the original network, one is interested in $V(1)$. The radius of convergence of $V(\gamma)$ is always strictly positive but can be arbitrarily small, so $V(1)$ need not converge. One way to obtain convergence is by the epsilon algorithm, which is an efficient and stable way to calculate Padé-approximants. Padé-approximants replace the partial sum $V_R(\gamma)$ by a quotient of partial sums $V_S^1(\gamma)$ and $V_{R-S}^2(\gamma)$, in such a way that they coincide in all first R coefficients:

$$V_R(\gamma) = \frac{\sum\limits_{0\le r\le S} \gamma^r v_r^1}{\sum\limits_{0\le r\le R-S} \gamma^r v_r^2} + O(\gamma^{R+1}), \quad \text{for } \gamma \downarrow 0.$$

This way, singularities of $V(\gamma)$ can be included in the denominator. The epsilon algorithm usually improves the speed of convergence considerably, as can be seen from the examples in section 4.

Another way to remedy singularities is by using the bilinear conformal mapping

$$\theta(\gamma) = \frac{(1+G)\gamma}{1+G\gamma}, \quad \gamma(\theta) = \frac{\theta}{1+G(1-\theta)}, \quad \text{for } G \ge 0.$$

This mapping maps the unit interval [0,1] onto itself and for $G \to \infty$ it maps the disk $|\gamma - \frac{1}{2}| \le \frac{1}{2}$ onto the unit disk. If $V(\gamma)$ has no singularities in $|\gamma - \frac{1}{2}| \le \frac{1}{2}$, this mapping can be used to map all singularities outside the unit disk, to make the power-series expansions converge at $\gamma = \theta = 1$. If $V(\gamma)$ is analytic at $\gamma = 0$, the power-series expansion in θ is

$$V\left(\gamma(\theta)\right) = \sum_{r\ge 0} [\gamma(\theta)]^r v_r = \sum_{r\ge 0} \theta^r w_r,$$

where

$$\begin{aligned} w_0 &= v_0, \\ w_r &= \left(\tfrac{G}{1+G}\right)^r \sum_{s=1}^{r} \binom{r-1}{s-1} \tfrac{v_s}{G^s}, \quad \text{for } r \ge 1. \end{aligned}$$

Instead of calculating the coefficients of the expansion in θ from the expansion in γ, they can also be calculated directly. An advantage of direct calculation is that the sequence $\{v_r,\ r \ge 0\}$ grows faster in r than the sequence $\{w_r,\ r \ge 0\}$, so calculation of $\{w_r,\ r \ge 0\}$ will normally be more stable. Because the mapping is conformal and maps 0 onto 0, the steady-state probabilities as a

function of θ also satisfy the order property. Moreover, the $P_\theta(\boldsymbol{n})$ are analytic in θ, so they can be represented by their power-series expansions

$$P_\theta(\boldsymbol{n}) = \sum_{r \geq |\boldsymbol{n}|} \theta^r W_r(\boldsymbol{n}), \quad \text{for } \boldsymbol{n} \in \mathbb{N}^S.$$

As before, the coefficients of the empty states $W_r(\mathbf{0}), r \geq 0$, satisfy (**??**). Because the mapping is a quotient of finite polynomials, the coefficients can still be calculated by a linear recursive algorithm. Assume that $\bar{r} = \sup_m r_m$ is finite. Replacing γ by $\gamma(\theta)$ in the balance equations (**??**), multiplying both sides by $[1 + G(1-\theta)]^{\bar{r}}$, and equating coefficients of corresponding powers of θ renders the new recursive equations for the non-empty states. The mapping was not used in the examples in section 4, because the power series were regular enough to obtain convergence by means of only the epsilon algorithm.

4 Examples

The examples in sections 4.1 and 4.2 consider the optimal order of queues in series and the dependence on higher moments of the service-time distributions of the total number of customers in cyclic networks.

4.1 Optimal Order

An important design problem in queueing theory is how, for a given arrival process and service-time distributions, the queues should be ordered in series, such that the mean sojourn time of customers is minimized, or equivalently the mean queue length. Exact analysis is in general very difficult, even for 2 queues. Whitt [**?**] proposes a heuristic based on the approximation of the departure process of each queue by a renewal process, characterized by the first two moments of the renewal interval. Greenberg and Wolff [**?**] proposed a heuristic based on light traffic behaviour and gave some examples where both heuristics did not give the same solution. They warned that extreme caution must be used in applying approximations to develop design procedures and stated that a heuristic based only on mean and squared coefficient of variation cannot be expected to work well. However, they did not indicate how large the difference in performance of both suggested solutions would be.

Consider the following model. According to a Poisson process with rate λ, customers arrive to obtain service from two servers. Both servers have an Erlang(2) service-time distribution, one with mean 1 and the other with mean 4. Should the customers first visit the fast or the slow server and does the optimal order depend on the arrival rate λ ? According to Whitt the optimal order is to visit the fast server first; Greenberg and Wolff suggest that, in light traffic, the slower server should be visited first. In the table below, the expected total number of customers is shown for both orders and different loads. The indicated load is the load of the slower server and corresponds to arrival rates 0.4, 1.2, 2.0, 2.8 and 3.6.

μ_1, μ_2	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
1, 4	0.1337	0.4745	1.009	2.118	7.231
4, 1	0.1335	0.4734	1.005	2.111	7.218

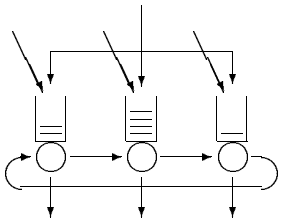
To visit the slower server first is better in all cases, but clearly the difference is negligible. Numerical experiments indicate that visiting the slower server first is still slightly better when the exponential interarrival times are replaced by Erlang or hyperexponential distributions or when the means of the service-time distributions are taken further apart (but with equal coefficient of variation).

Convergence of the power series was slowest for the model with $\rho = 0.9$ and the fast server first. In the table below the original series and the series after applying the epsilon algorithm are shown.

R	1	5	10	20	40	60
$V_R(1)$	0.2250	1.766	3.245	4.877	6.410	6.945
$\epsilon[V_R(1)]$	0.2250	1.766	7.362	7.232	7.231	7.231

The original series seems to converge monotonically, but after applying the epsilon algorithm, convergence is much faster. In general, convergence is slower if the load of the original network is higher and the parameters of the model are more extreme. For example, hyperexponential distributions result in slower convergence than Erlang distributions.

4.2 Insensitivity for Higher Moments



Consider the following model. Customers arrive according to a process that is a mixture of independent identical Poisson arrival processes and fork arrivals. The independent Poisson processes have rate λ_1, the simultaneous fork arrivals have exponential interarrival times with rate λ_2:

$$
\begin{array}{llll}
I = 1, & \alpha_{11} = S\lambda_1 + \lambda_2, & & \\
M = S + 1, & \boldsymbol{b}_m = \boldsymbol{e}_m \ (1 \le m \le S), & \boldsymbol{b}_{S+1} = \boldsymbol{e}, & \\
q_{0,1,1} = 0, & q_{m,1,1} = \frac{\lambda_1}{S\lambda_1+\lambda_2} \ (1 \le m \le S), & q_{S+1,1,1} = \frac{\lambda_2}{S\lambda_1+\lambda_2}. &
\end{array}
$$

The routing is such that, after service completion at a queue, customers either leave the network with probability p, or go to the next queue with probability $1 - p$:

$$
\chi_{st} = \begin{cases} p & t = 0, \\ 1 - p & t = s \bmod S + 1, \\ 0 & \text{otherwise.} \end{cases}
$$

Depending on the coefficient of variation, the service-time distributions at the different queues are either Erlang or hyper-exponential with balanced means. In the table below, the probability of an empty network and the mean queue lengths are given for four different models with 3

queues, $\lambda_1 = \lambda_2 = 0.09, p = 0.2$ and $\mu_1 = \mu_2 = \mu_3 = 1$. This way, all queues have identical load $\rho_1 = \rho_2 = \rho_3 = 0.9$. The difference between the four models is in the variance σ_s^2 of the service-time distributions at the queues.

$\sigma_1^2, \sigma_2^2, \sigma_3^2$	$\Pr\{\boldsymbol{N} = \boldsymbol{0}\}$	E $\{N_1\}$	E $\{N_2\}$	E $\{N_3\}$	E $\{\lvert\boldsymbol{N}\rvert\}$
$1, 1, 1$	0.0033	10.28	10.28	10.28	30.84
$2, \frac{1}{2}, \frac{1}{2}$	0.0038	12.95	9.928	7.760	30.63
$1\frac{1}{2}, 1, \frac{1}{2}$	0.0035	11.39	10.97	8.426	30.78
$\frac{1}{2}, 1, 1\frac{1}{2}$	0.0035	9.141	9.701	11.95	30.79

The convergence of the power series of E $\{\lvert\boldsymbol{N}\rvert\}$ in the second model was poorest:

R	1	5	10	20	40	60
$V_R(1)$	0.2700	2.007	4.326	8.387	14.79	19.37
$\epsilon[V_R(1)]$	0.2700	-0.2523	0.1951	32.41	30.59	30.63

Again, both series seem to converge, but more coefficients need to be calculated to stabilize than in section 4.1.

It can be seen that the mean queue length of each queue is increasing in the variance of the service-time distribution of both the queue itself and the preceding queue. From the last column it can be seen that the expected total number of customers in the network is approximately equal for all four models. For $p = 1$, this follows immediately from the Pollaczek-Khintchine formula, because then all queues are *M/G/1* queues with identical load and mean service time, so:

$$\mathrm{E}\{\lvert\boldsymbol{N}\rvert\} = S\rho_1 + \frac{\rho_1^2}{2(1-\rho_1)}\left(1 + \frac{1}{\mu_1^2}\sum_{s=1}^{S}\sigma_s^2\right).$$

The variances of the service-time distributions are chosen, such that their sum is equal to 3 for all four models. Numerical experiments indicate that the property that the mean total queue length is mainly determined by the sum of the variances also holds for more general models, namely for networks with a symmetric arrival process, cyclic routing and equal loads at the different queues. Here, symmetric means that I and A can be arbitrary, but if $\boldsymbol{b}_m$ is a possible arrival, then each permutation $\boldsymbol{b}_{\tilde{m}}$ of $\boldsymbol{b}_m$ is also a possible arrival and $Q_m = Q_{\tilde{m}}$ (more intuitively, that at each arrival of a batch, an arrival of any permutation of this batch would have been equally likely). A cyclic routing matrix X is a matrix such that

$$\chi_{s,t} = \chi_{s \bmod S+1,\, t \bmod S+1}, \quad \text{for } 1 \le s, t \le S.$$

If the arrival process is symmetric and the routing cyclic, then the loads at the queues are identical if the mean service times are identical. For such networks the following hypothesis can be formulated:

For networks of ./G/1 queues, with a symmetric arrival process, cyclic routing and equal loads at all queues, the expected total number of customers in the network is mainly determined by the sum of the variances of the service-time distributions, and not so much by their shapes.

Of course such a hypothesis could never be proved by the *PSA*, but it can be used to evaluate various 'randomly' chosen models and models that are likely to be counter-examples.

5 Conclusions

A method was proposed to analyze a wide class of Markovian queueing networks. Because of the 'curse of dimensionality', the size of the networks must necessarily be moderate. Networks of up to 4 or 5 queues can be analyzed if the algorithm is programmed carefully, methods to improve the convergence of power series are employed and the parameters of the model are not too extreme. With a good user interface to determine the parameters for a particular model, the Power-Series Algorithm provides a means to easily evaluate many different models. Therefore, it can be an aid for studying the interaction between queues and for testing and developing approximations of performance measures and heuristics.

References

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

[2] Blanc, J.P.C., R.D. van der Mei, Optimization of polling systems by means of gradient methods and the power-series algorithm, *Tilburg University, Report FEW 575* (1992).

[3] Blanc, J.P.C., Performance evaluation of polling systems by means of the power-series algorithm, *Annals of Operations Research* **35** (1992), 155-186.

[4] Blanc, J.P.C., The power-series algorithm applied to the shortest-queue model, *Operations Research* **40** (1992), 157-167.

[5] Blanc, J.P.C., Performance analysis and optimization with the power-series algorithm, in *Performance Evaluation of Computer and Communication Systems*, eds. L. Donatiello, R. Nelson, Springer-Verlag Berlin, 1993, 53-80.

[6] Greenberg, B.S., R.W. Wolff, Optimal order of servers for tandem queues in light traffic, *Management Science* bf 34 (1988), 500-508.

[7] Hooghiemstra, G., M. Keane, S. van de Ree, Power series for stationary distributions of coupled processor models, *SIAM J. Appl. Math.* **48** (1988), 1159-1166.

[8] Hout, W.B. van den, J.P.C. Blanc, The power-series algorithm extended to the BMAP/PH/1 queue. *Tilburg University, Center Discussion Paper 9360* (1993).

[9] Koole, G., On the power series algorithm, *CWI Amsterdam, Report BS-9404* (1994).

[10] Lucantoni, D.M., New results on the single server queue with a batch Markovian arrival process, *Commun. Statist.-Stochastic Models* **7** (1991), 1-46.

[11] Neuts, M.F., *Matrix Geometric Solutions in Stochastic Models: an algorithmic approach.* John Hopkins Univ. Press, Baltimore, 1981.

[12] Whitt, W., The best order for queues in series, *Management Science* **31** (1985), 475-487.