International Journal of Interdisciplinary Telecommunications and Networking
Volume 12 • Issue 4 • October-December 2020

Analysis of Queueing Networks in Equilibrium:

Numerical Steady-State Solutions of Markov Chains

Izabella V. Lokshina, SUNY Oneonta, USA

Cees J. M. Lanting, DATSA, Belgium

ABSTRACT

Equilibria of queueing networks are a means for performance analysis of real communication networks introduced as Markov chains. In this paper, the authors developed, evaluated, and compared computational procedures to obtain numerical solutions for queueing networks in equilibrium with the use of direct, iterative, and aggregative techniques in steady-state analysis of Markov chains. Advanced computational procedures are developed with the use of Gaussian elimination, power iteration, Courtois' decomposition, and Takahashi's iteration techniques. Numerical examples are provided together with comparative analysis of obtained results. The authors consider these procedures are also applicable to other domains where systems are described with comparable queuing models and stochastic techniques are sufficiently relevant. Several suitable domains of applicability are proposed.

KEYWORDS

Communication Networks, Markov Chains, Numerical Solutions, Performance Analysis, Queueing Networks In Equilibrium, Steady-State Analysis

INTRODUCTION

Queueing networks, which consist of several service stations, are frequently used in modeling to represent particular structures of real systems with large number of resources, such as computer and communication networks. In queueing networks at least two service stations are connected to each other. These service stations, or nodes in a queueing network, represent resources in a real system. Generally, jobs can be transferred between any two nodes of a queueing network; particularly, a job can be directly returned to the node it has just left.

A queueing network is *open* when jobs can enter the network from and leave the network to the outside. A queueing network is *closed* when jobs can neither enter nor leave the network. The number of jobs in a closed network is constant. Performance measure of a queueing network could be obtained while computing the steady-state probabilities (Hassin and Haviv, 2003; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Lokshina, Zhong and Lanting, 2020; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007).

DOI: 10.4018/IJITN.2020100101

Developing computational procedures to obtain the steady-state probabilities for all possible states in a queueing network is a central issue in the queueing theory. The means for many other important network performance measures can be easily obtained after having these algorithms developed (Dugas, Chapados, Ducharme, Saint-Mleux and Vincent; 2011; Hassin and Haviv, 2003; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007).

Markov processes provide flexible, powerful, and efficient means for evaluation and analysis of dynamic systems. Performance and reliability measures of a queueing network can be derived and evaluated with the steady-state analysis of Discrete-Time Markov Chains (DTMC) and Continuous-Time Markov Chains (CTMC). Generally, each queueing system can be considered as a specific case of Markov process, and then mathematically evaluated in terms of the process (Hassin and Haviv, 2003; Liu, Ma and Li, 2012; Lokshina, 2016; Lokshina and Bartolacci, 2012; Zhang, 2009). Direct and iterative methods can be used to obtain numerical solutions in the steady-state analysis of Markov chains (Bhalai, 2002; Liu, Ma and Zhang, 2015; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007).

Direct methods, as they are applied, can modify the parameter matrix. They can also require fixed amount of computational time, independently to parameter values. At the same time, direct methods are subject to accumulation of round-off errors and create significant difficulties with sparse storage (Hassin and Haviv, 2003; Mehdi, 2003; Lokshina, 2016; Radev, Lokshina and Denchev, 2007).

Iterative methods are based on a property of successive convergence to desired solutions, and evaluation process can be terminated as soon as iterates are sufficiently close to each other, i.e. final (or, exact) values. The main advantage of iterative methods, compared with direct methods, is because iterative methods maintain the parameter matrix (Stewart, 1994; Lokhina, 2016; Radev, Lokshina and Denchev, 2007), since efficient sparse storage schemes and efficient sparsity-preserving algorithms are applied. On the other hand, the main disadvantage of iterative methods is that convergence is not always guaranteed (Rubinstein and Melamed, 1998; Lokshina, 2016), and it is highly dependent on the applied method (Hassin and Haviv, 2003; Krieger, Muller-Clostermann and Sczittnick, 1990; Lokshina, 2016; Lokshina and Bartolacci, 2012). The rate of convergence is very sensitive for entry values in the parameter matrix (Trivedi, 2001; Lokshina, 2016; Radev, Lokshina and Denchev, 2007).

The important task here is to obtain exact or approximate numerical solutions for global and local balance equations of queueing systems with Markov chains (Hassin and Haviv, 2003; Hayes and Ganesh Babu, 2004; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007). This is a focal point for modeling closed and open queueing networks, especially with heavy tailed traffic.

Therefore, in this paper the authors developed and compared computational procedures to obtain numerical solutions for queueing networks in equilibrium with the use of direct, iterative and aggregative techniques in steady-state analysis of Markov chains. These procedures were developed primarily in the context of real communication networks; however, they are applicable also to other domains where systems are described with comparable queuing models and stochastic methods are sufficiently relevant. Some suitable domains of applicability are recommended.

This paper is comprised of ten sections and organized as follows. Section two defines global balance equations and marginal probabilities. Section three outlines algorithms for steady-state solutions of Markov chains. Sections four, five and six consider direct, iterative and approximation methods for numerical solutions, respectively. Section seven provides comparison of numerical results. Section eight proposes several additional domains of applicability for developed computational procedures. Section nine offers conclusions followed by acknowledgement and references.

GLOBAL BALANCE EQUATIONS AND MARGINAL PROBABILITIES

Global balance equations are a set of equations that define the equilibrium distribution of a queueing network represented as Markov chain. Therefore, obtaining steady-state probabilities $\pi(k_1, k_2,\ldots, k_N)$ for all possible states in a queueing network is very important as then the mean values of other performance measures can be easily calculated (Hassin and Haviv, 2003; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007). Multi-dimensional state spaces are considered, and $\pi(k_1, k_2,\ldots, k_N)$ denotes steady-state probabilities of states $(k_1, k_2,\ldots, k_N)$.

The most important performance measures of a queueing network are marginal probabilities $\pi_i(k)$. For a closed queueing network, marginal probabilities $\pi_i(k)$ indicate that the i-th node contains exactly $k_i = k$ jobs which can be calculated according to equation (1):

$$\pi_i\left(k\right) = \sum_{\sum_{j=1}^{N} k_j = K}^{k_i = k} \pi\left(k_1,\ldots,k_N\right) \tag{1}$$

Marginal probabilities $\pi_i(k)$ are joint probabilities for all possible states $(k_1, k_2,\ldots,k_N)$, 0 £ k_i £ K that satisfy the condition (2),

$$\sum_{j=1}^{N} k_j = K \tag{2}$$

while a fixed number of jobs, k, is specified for the i-th node. Normalization condition for joint probabilities is shown in equation (3).

$$\sum_{\sum_{j=1}^{N} k_j = K} \pi\left(k_1,\ldots,k_N\right) = 1 \tag{3}$$

For an open network, marginal probabilities can be calculated according to (4), with the use of normalization condition (5).

$$\pi_i\left(k\right) = \sum_{k_i = k} \pi\left(k_1,\ldots,k_N\right) \tag{4}$$

$$\sum \pi\left(k_1,\ldots,k_N\right) = 1 \tag{5}$$

Using marginal probabilities, the other important performance measures for open and closed networks can be obtained, such as utilization, throughput, mean number of jobs, mean queue length, mean response time and mean waiting time.

ALGORITHMS FOR STEADY-STATE SOLUTIONS OF MARKOV CHAINS

Based on fundamental equations $\nu = \nu P$ and $0 = \pi Q$, the model (6) can be used to obtain the steady-state probability vector in ergodic Markov chains:

$$0 = \nu \left(P - I\right) \tag{6}$$

Accordingly, for both discrete-time and continuous-time Markov chains, linear system (7) can be solved:

$$0 = xA \tag{7}$$

Based on entries that represent parameters of Markov chain, matrix A is singular, and it has rank of n-1 for any Markov chain of size $\left|S\right| = n$. In order to obtain a unique positive solution, normalization condition can be directly imposed into equation (7) with the use of equation (8).

$$x1 = 1 \tag{8}$$

This can be defined as substitution of one column (let's say, the last column) of matrix A with a unit vector. The resulting linear system of non-homogeneous equations can be written according to equation (9):

$$b = xA, \qquad b = [0, 0, \ldots, 0, 1] \tag{9}$$

For any given ergodic continuous-time Markov chain, discrete-time Markov chain can be developed, which obtains identical steady-state probability vectors like in case of CTMC. Let's consider generator matrix $Q - [q_{ij}]$ for continuous-time Markov chain, as shown in equation (10):

$$P = Q \,/\, q + I \tag{10}$$

where $q > \max_{i,j \in S} \left|q_{ij}\right|$.

The situation $q = \max_{i,j \in S} \left|q_{ij}\right|$ must be avoided, in order to assure aperiodicity of the resulting discrete-time Markov chain (Mehdi, 2003; Lokshina, 2016; Radev, Lokshina and Denchev, 2007). The obtained matrix P can be used to determine the steady-state probability vector $\pi = \nu$, with solving $\nu = \nu P$ and $\nu 1 = 1$.

The technique that is frequently applied in order to reduce CTMC to DTMC is called randomization (Hassin and Haviv, 2003; Hayes and Ganesh Babu, 2004), and sometimes, uniformization (Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007).

On the other hand, the transition probability matrix P for ergodic discrete-time Markov chain is given, and generator matrix Q of CTMC can be defined according to equation (11):

$$Q = P - I \tag{11}$$

After resolving $0 = \pi Q$ by considering condition $\pi 1 = 1$, a desired steady-state probability vector $\pi = \nu$ can be obtained.

DIRECT METHODS FOR NUMERICAL SOLUTIONS

Closed-form methods are applicable for solutions when Markov chains possess particular structures. For Markov chains with more general structures, using numerical methods must be avoided (Bhalai, 2002; Hassin and Haviv, 2003; Lokshina, 2016; Radev, Lokshina and Denchev, 2007).

Many direct methods are known to solve system of linear equations (Stewart, 1994). Some of them are restricted to certain regular structures of parameter matrices that are of less importance for Markov chains, since these structures generally cannot be applied with Markov chains. Among the most frequently applied techniques are Gaussian elimination procedure, as well as Grassmann, Taksar, and Heyman techniques, which are based on renewal arguments of Gaussian elimination (Hassin and Haviv, 2003; Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007).

Gaussian elimination procedure suffers sometimes from numerical difficulties created by subtractions of nearly equal numbers. These particular properties, as well as their variants through reformulations on the base of regenerative properties of Markov chains, can be avoided when applying Grassmann, Taksar, and Heyman algorithms. The basic idea of Gaussian elimination is transformation of the original system (12) into another equivalent system with the use of elementary operations on the parameter matrix, which preserve rank of matrix.

$$\begin{aligned}
&a_{0,0}x_0 + a_{1,0}x_1 + \cdots + a_{n-1,0}x_{n-1} = b_0,\\
&a_{0,1}x_0 + a_{1,1}x_1 + \cdots + a_{n-1,1}x_{n-1} = b_1,\\
&\quad\vdots\\
&a_{0,n-1}x_0 + a_{1,n-1}x_1 + \cdots + a_{n-1,n-1}x_{n-1} = b_{n-1}.
\end{aligned} \tag{12}$$

For obtaining a desired solution **x**, which is identical to the solution of original system, an equivalent system of linear equations with triangular matrix structure can be derived as equation (13).

$$\begin{aligned}
&a_{0,0}^{(n-1)}x_0 = b_0^{(n-1)},\\
&a_{0,1}^{(n-2)}x_0 + a_{1,1}^{(n-2)}x_1 = b_1^{(n-2)},\\
&\quad\vdots\\
&a_{0,n-1}^{(0)}x_0 + a_{1,n-1}^{(0)}x_1 + \cdots + a_{n-1,n-1}^{(0)}x_{n-1} = b_{n-1}^{0}.
\end{aligned} \tag{13}$$

As soon as the original system (12) is transformed into a triangular structure, the final results can be obtained by applying straightforward substitution process. In order to obtain (13), Gaussian elimination procedure needs to be performed on the original system (Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007).

Approximately, this algorithm can be described as follows: first, the *n*-th equation in (12) is solved for x_{n-1}, and then x_{n-1} is eliminated from all other *n*-1 equations. Next, the (*n*-1)-th equation is solved for x_{n-2}, and, again, x_{n-2} is eliminated from the remaining *n*-2 equations; and so forth. Finally, system (13) is resolved, where $a_{i,j}^{(k)}$ denotes coefficient of x_i in the (*j*+1)-th equation, which is obtained after the *k*-th elimination step.

Gaussian elimination is based on elementary matrix operations that preserve rank of matrix. Such elementary matrix operations correspond to interchange and multiplication of equations by real-value constant. In matrix terms, an essential part of Gaussian elimination procedure is provided by factorization of parameter matrix A into elements of upper triangular matrix U and lower triangular matrix L.

Finally, computation of consequential vector x can be made by applying two simple steps according to equation (14).

$$b = xA = xUL = yL \tag{14}$$

At first, equation yL = b is resolved for a vector of unknowns y, and then equation xU =y is resolved for the vector of unknowns x.

Advanced computational procedure, developed according to Gaussian elimination, is demonstrated as Algorithm 1.

Algorithm 1:

Step 1: Construct parameter matrix A and right-hand-side vector b according to equation: $b = xA, \quad b = [0,0,...,0,1]$.

Step 2: Carry out elimination steps or apply standard algorithm to split parameter matrix A into upper triangular matrix U and lower triangular matrix L such that $A = UL$ holds. Note that parameters of U can be computed with formula:

$$a_{ij}^{(k)} = \begin{cases} 0, & \begin{matrix} j = n-k-1, n-k-2, ..., 0 \\ i = n-1, n-2, ..., n-k, \end{matrix} \\ a_{ij}^{(k-1)} - a_{n-k}^{(k-1)} \dfrac{a_{n-k,j}^{(k-1)}}{a_{n-k,n-k}^{(k-1)}}, & otherwise \end{cases},$$

as computation of L can be deliberately avoided.

Step 3: Compute intermediate values y according to equation: $yL = b$**,** or compute intermediate values according to formula:

$$b_j^{(k)} = b_j^{(k-1)} - b_{n-k}^{(k-1)} \frac{a_{n-k,j}^{(k-1)}}{a_{n-k,n-k}^{(k-1)}},$$
$$where \qquad j = n-k-1, n-k, ..., 0.$$

Step 4: Perform substitution to obtain finally **x** according to equation: $xU = y$ by applying formula:

$$x_0 = \frac{b_0^{(n-1)}}{a_{0,0}^{(n-1)}}, x_j = \frac{b_j^{(n-j)}}{a_{j,j}^{(n-j)}} - \sum_{k=0}^{j-1} \frac{a_{k,j}^{(n-j)}}{a_{j,j}^{(n-j)}} x_k, \quad j = 1, 2, ..., n-1.$$

ITERATIVE METHODS FOR NUMERICAL SOLUTIONS

Convergence is important problem for iterative methods (Hassin and Haviv, 2003; Radev, Lokshina and Denchev, 2007). A Heuristic approach can be applied to select appropriate techniques for decision-making on convergence, but there are no general algorithms for the selection of such techniques.

An Estimate of error must be made to determine convergence because the desired solution vector is not known. Tolerance level ε must be specified in order to provide a measure of how close the current iteration vector $x^{(k)}$ is to the desired solution vector **x**.

Some distance measures are often used to evaluate current iteration vector $x^{(k)}$ in relation to some previous iteration vectors $x^{(l)}, l < k$. If the current iteration vector is "close enough" to previous iteration vectors with respect to *ε,* then this condition is used as indicator of convergence to the final result. If *ε* is too small, convergence can become very slow or not happen at all. If *ε* is too large, accuracy requirements can be violated, or worse, convergence can be wrongly assumed.

Some appropriate norm functions must be applied in order to compare different iteration vectors. Size and type of the parameter matrix must be considered for the right choice of such a norm function. Concerning the right choice of *ε* and norm function, the authors assume that elements x_i of solution vector can differ significantly from each other.

The power iteration procedure (Lokshina, 2016; Lokshina and Bartolacci, 2012; Lokshina and Bartolacci, 2007; Radev, Lokshina and Denchev, 2007) is a reliable iterative technique to compute steady-state probability vector of finite ergodic Markov chains. It sometimes tends to converge slowly, and the only required condition for convergence is aperiodic transition probability matrix P, and then the irreducibility condition is unnecessary.

The power iteration procedure follows transient behavior of underlying discrete-time Markov chains until some stationary, not necessarily steady-state, convergence is obtained. Therefore, it can also be used as technique to compute the transient state probability vector $\nu(n)$ of DTMC.

Equation $\nu = \nu P$ recommends to start with the initial guess of some probability vector $\nu^{(0)}$, and repeatedly multiply it by transition probability matrix P until convergence to ***ν*** is obtained, with $\lim_{i \to \infty} \nu^{(i)} = \nu$. Since ergodicity or, at least, aperiodicity of the underlying Markov chain is assumed, this procedure is guaranteed to converge to a desired fixed point of a unique steady-state probability vector. A single iteration step is the following according to equation (15):

$$\nu^{(i+1)} = \nu^{(i)} P, \qquad i \geq 0 \tag{15}$$

The relation between iteration at step *i* and initial probability vectors can be presented as equation (16):

$$\nu^{(i)} = \nu^{(0)} P^i, \qquad i \geq 0 \tag{16}$$

To obtain the final result of steady-state probability vector ***ν*,** only renormalization remains to be executed. The speed of convergence of power iteration procedure depends on relative sizes of eigenvalues. The closest non-dominant eigenvalue is equal to 1, which slows up convergence.

Advanced computational procedure, developed according to power iteration, is demonstrated as Algorithm 2.

Algorithm 2:

Step 1: Select appropriate q for formula

$$A = \begin{cases} P, \\ Q / q + I; \end{cases} \qquad \nu^{(0)} = \left(\nu_0^{(0)}, \nu_1^{(0)}, \ldots, \nu_{n-1}^{(0)}\right).$$

Select convergence criterion ε , and set $n = 0$ .Define appropriate vector norm function $f\left(\left\|\nu^{(n)}, \nu^{(l)}\right\|\right), n \geq l$.

Set convergence = false.

Step 2: Repeat until convergence:
Step 2.1: $\nu^{(n+1)} = \nu^{(n)} A$;
Step 2.2: If $f\left(\left\|\nu^{(n+1)}, \nu^{(l+1)}\right\|\right) < \varepsilon, l \leq n$, Then convergence = true.
Step 2.3: $n = n + 1, \qquad l = l + 1$.
Step 3: Obtain finally: $\left.\begin{matrix} \pi \\ \nu \end{matrix}\right\} \approx \nu^{(n)}$

APPROXIMATION METHODS FOR NUMERICAL SOLUTIONS

Two basic approximation methods for steady-state analysis of Markov chains (Bhalai, 2002; Hassin and Haviv, 2003; Lokshina, 2016; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007) are frequently used to analyze communication systems with Markov chains. The Courtois's non-iterative approximation procedure is usually applied in order to estimate computing $\nu^{\approx}$ of a desired state probability vector ν .

Courtois's non-iterative approximation is highly efficient because instead of solving one linear system of equations with size of state space *S,* several smaller linear systems are solved independently. One system is solved for each subset S_j of partitioned state space *S,* and another system is solved for aggregate chain (Lokshina, 2016; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007).

Takahashi's iterative approximation procedure differs substantially from Courtois's non-iterative approximation with respect to both conditions of useful methodology and applicability. While Courtois's non-iterative approximation is applied to estimate computations of steady-state probability vector $\nu^{\approx}$ for assumed ergodic DTMC or $\pi^{\approx}$ in the case of CTMC, Takahashi's iterative approximation procedure allows computations of the exact state probability vector. To perform a straightforward comparison of the two techniques, any given an ergodic CTMC can easily be transformed into ergodic DTMC.

Courtois's non-iterative approximation is based on decomposability properties of models considered by the authors; when Takahashi's iterative aggregation is executed by using independently computed sub-results as elements to compose the final results. Applicability of techniques must be confirmed in each case. If a Markov chain has tightly coupled subsets of states, and states within each subset are tightly coupled to each other and weakly coupled to states outside this subset, it indicates that the technique is applicable (Lokshina, 2016; Radev, Lokshina and Denchev, 2007; Radev, Lokshina and Radeva, 2007).

Then such subsets of states must be aggregated to form macro-states as basis for further analysis. Macro-state probabilities, together with conditional micro-state probabilities within the subsets, can be composed to obtain micro-state probabilities of the initial model (Lokshina, 2016; Radev, Lokshina and Denchev, 2007).

Let P be the transition probability matrix. State space, associated with P, is partitioned into groups, often referred to as lumps, such that any state within a group contains the same number of customers in specific portion of state descriptor. Transition probability matrix P, obtained according to these groups, consists of blocks of transition probabilities along diagonal, and of other blocks along off-diagonals above and below diagonal. P is assumed to be almost completely decomposable if the sum of non-zero transition probabilities from each row positioned within diagonal block is close to 1.

That means that the sum of off-diagonal probabilities along a row is extremely small. It occurs when majority of transitions are between states of the same group, and with very few transitions between states of different groups. Generally, the error induced by the technique could be bounded, but formal error bounding is frequently omitted (Lokshina, 2016; Radev, Lokshina and Denchev, 2007).

Advanced computational procedure, developed according to Courtois's non-iterative approximation, is demonstrated as Algorithm 3.

Algorithm 3:

Step 1: Create the state space and arrange it correctly according to a pattern of decomposition.

Step 2: Generate the transition probability matrix *P* with the use of randomization technique *P* = *Q/q* + *I* if the starting point is CTMC, and partition P into $M \times M$ number of sub-matrices $P^*_{IJ}, 0 \le I, J \le M-1$.

Step 3: Verify the nearly complete decomposability of P with the chosen value of ε according to formula:

$$\varepsilon < \frac{1 - \max_{\mathrm{I}} \left|\lambda^*_{\mathrm{I}}(2)\right|}{2}.$$

Step 4: Decompose *P* in a way that $P = P^* + \varepsilon \cdot \tilde{C}$. Matrix P^* contains only stochastic diagonal sub-matrices P^*_{II}, and ε is a precision measure of Courtois's approximation. It is defined as the maximum sum for entries of non-diagonal sub-matrices $P^*_{IJ}, I \neq J$ of *P*.

Step 5: For each $I, 0 \le I, J \le M-1$, solve equation $v^*_I \cdot P^*_{II}, = v^*_I$, with $v^*_I \cdot 1 = 1$ to obtain the conditional state probability vectors v^*_I.

Step 6: Compute the appropriate coupling between decomposed macro states:

Step 6.1: Generate the transition probability matrix $\Gamma = \left[\Gamma_{IJ}\right]$ according to formula: $\Gamma_{IJ} = \sum_{i \in S_I} \left(v^*_{Ii} \sum_{j \in S_j} p_{IJij} \right);$

Step 6.2: Solve equation $\gamma \Gamma = \gamma$, to obtain the macro steady-state probability vector γ.

Step 7: Compute the approximate steady-state probability vector $v^{\approx}$ for appropriate micro states with the use of conditional state probability vectors $v^*_I, 0 \le I \le M-1$, according to equation:

$v^{\approx}_{Ii} = \gamma_I v^*_{Ii},\ 0 \le I \le M-1$ and $\forall i \in S_I$

Step 8: With obtained $\nu^{\approx}$ compute finally the steady-state performance and dependability measures along the lines of specified reward structure.

Compared to Courtois's non-iterative approximation technique, the Takahashi's iterative approximation procedure partitions state spaces S into M separate subsets of states $S_I \subset S, \quad 0 \le I \le M-1$, such that each subset S_{I} is aggregated into macro-state I. However, criteria applied to cluster states are different in these two techniques. Calculations of transition probabilities Γ_{IJ} among macro-states I and J are performed with the use of conditional micro-state probability vector ν_I^* and initially assumed transition probabilities p_{ij}, or $p_{\mathrm{IJ}ij}$.

With Courtois's non-iterative approximation, the conditional probability vectors ν_I^* for a given partition element I can be obtained separately for each subset of micro-states $S_I \subset S, 0 \le I \le M-1$, if the initial model is almost completely decomposable. In Takahashi's iterative approximation, partitioning of state space is performed in a different way, with the use of approximate lump-ability presented in terms of DTMC.

However, this choice implies no limitations since ergodic DTMC and CTMC are equivalent with respect to their steady-state probability computations when applying transformations. Takahashi's iterative approximation is more convenient being used directly on generator matrix $Q = \left[q_{ij} \right]$ of CTMC; additionally, it allows to obtain an exact state probability vector.

Advanced computational procedure, produced according to Takahashi's iterative approximation, is shown as Algorithm 4.

Algorithm 4:

Step 1: Create the state space S and arrange it suitably according to a pattern of decomposition along the lines of approximate lump-ability into transition probability matrix.

Step 2: Initialization:

Step 2.1: n:= 0.

Step 2.2: Estimate ν^0.

Step 2.3: Choose ε and 0< c < 1.

Step 2.4: Choose appropriate vector norm function f(||…||).

Step 3: $f\left(\left\|\nu^{(n)} - \nu^{(n)}P\right\|\right) \ge \varepsilon$.

Step 4: *Geometric convergence from Y to N: n ≥ 1 residual error* $\left(\nu^{(n)}\right)$>c* residual error $\left(\nu^{(n-1)}\right)$

$\nu' = \nu^{(n)}; \nu^{(n)} = \nu' P$, n:=n+1.

Step 5 Aggregation for all: 0 ≤ I, J ≤ M -1.

Step 5.1: Generate the system of equations according to formula:

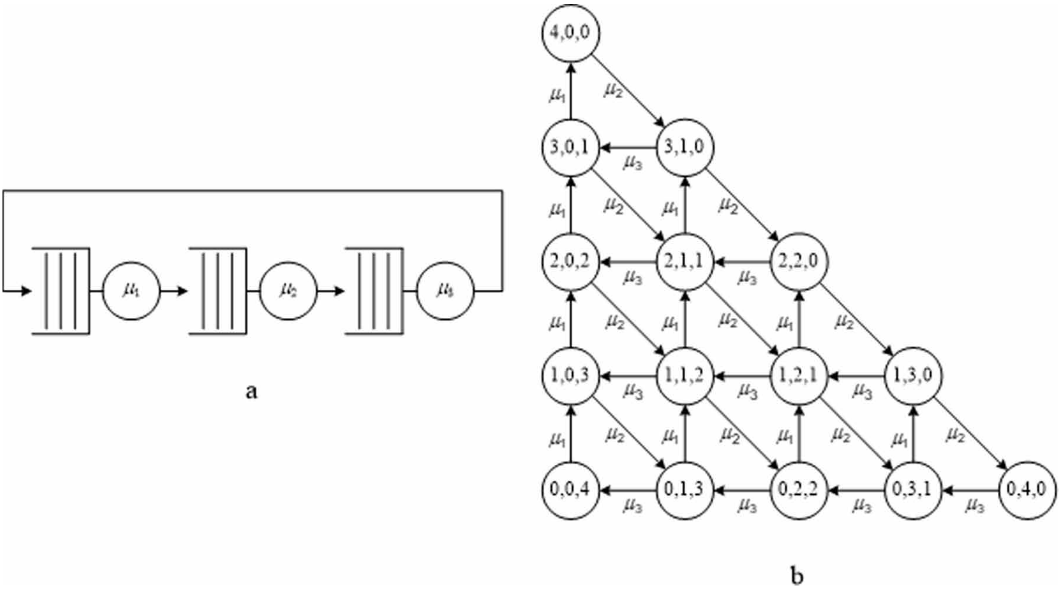
$$\Gamma^{(n)} = \left[\Gamma_{IJ}^{(n)}\right] = \left[\sum_{i \in S_I} \sum_{j \in S_J} \frac{\nu_i^{(n-1)} p_{ij}}{\sum_{k \in S_I} \nu_k^{(n-1)}}\right].$$

Step 5.2: *Solve* $\gamma^{(n)} = \gamma^{(n)} \cdot \Gamma^{(n)}$ according to: $\gamma = \gamma \cdot \Gamma, \qquad \gamma \cdot 1 = 1$.

Step 6: Disaggregation for all: $0 \le I \le M-1$.

Step 6.1: Calculate appropriately:

Figure 1. Queueing model (a) and transition rate diagram (b)



$$\Gamma_{Ij}^{(n)} = \sum_{i \in S_I} \frac{v_i^{(n-1)} p_{ij}}{\sum_{k \in S_I} v_k^{(n-1)}} \qquad \forall j \in S \text{ according to } \Gamma_{Ij} = \sum_{i \in S_I} \frac{v_i^{\approx} p_{ij}}{\sum_{k \in S_I} v_k^{\approx}}$$

Step 6.2: *Calculate* $v_I^{(n)} = \left[v_{Ij} \right]$ by solving appropriate system of equations:

$$v_i^{(n)} = \sum_{j \in S_I} v_j^{(n)} p_{ji} + \sum_{K=0, K \neq I}^{M-1} \gamma_K^{(n)} \Gamma_{K\ i}^{(n)}, \qquad \forall i \in S_I ;$$

Step 7: *With* $v^{(n)} = \left[v_{\mathrm{I}}^{(n)} \right]$ obtained in the previous step, solve finally $v^{(n)} \cdot \mathbf{1} = 1$.

COMPARISON OF NUMERICAL RESULTS

In this paper, the authors considered implementations of algorithms together with illustrations. To begin with, the authors studied an example of a closed queueing network with three nodes (*N*=3) and four customers (*K*=4), as shown in Figure 1(a).

The number of evident states of the system is equal to possible variation and can be calculated as binomial coefficient according to formula (17).

$$\binom{N+K-1}{N-1} = \binom{6}{2} = \frac{6!}{2!\ \ 4!} = 15 \tag{17}$$

State transition rate diagram for this queueing model is presented with a Markov chain that is shown in Figure 1(b). This diagram demonstrates possible transitions between Markov chain nodes. Transition rates between states are equal to $\mu_1 = 0.5; \mu_2 = 0.4; \mu_3 = 0.1$.

Table 1. Numerical results for steady-state probabilities computed with proposed algorithms for example 1

State	Exact Value	Algorithm 1		Algorithm 2		Algorithm 3	
		Computed	Error $\times 10^{-6}$	$\nu^{(90)}$	Error $\times 10^{-6}$	$\nu^{\approx}$	Error
(0,0,4)	0.000962837	0.000962837	0.252	0.000962837	0.287	0.044678333	0.9784
(0,1,3)	0.004814184	0.004814184	0.037	0.004814184	0.065	0.0340617	0.8587
(0,2,2)	0.024070919	0.024070919	0.01	0.024070919	0.014	0.029638102	0.1878
(0,3,1)	0.120354595	0.120354595	0.002	0.120354594	0.011	0.065026881	0.8508
(0,4,0)	0.601772974	0.601772974	0	0.601772969	0.009	0.218083349	1.7594
(1,3,0)	0.150446243	0.150446243	0	0.150443245	0.011	0.125959943	0.1944
(2,2,0)	0.037510811	0.037510811	0.006	0.037610813	0.057	0.142925268	0.7368
(3,1,0)	0.009402703	0.009402703	0.052	0.009402704	0.126	0.115046814	0.9183
(4,0,0)	0.002350674	0.002350674	0.186	0.002350676	0.971	0.062154431	0.9622
(3,0,1)	0.00188054	0.00188054	0.069	0.001880541	0.415	0.021921065	0.9142
(2,0,2)	0.001504433	0.001504433	0.166	0.001504432	0.352	0.023430384	0.9358
(1,0,3)	0.001203546	0.001203546	0.174	0.001203546	0.067	0.020034416	0.9399
(1,1,2)	0.00601773	0.00601773	0.049	0.00601773	0.048	0.006432901	0.0645
(1,2,1)	0.030088649	0.030088649	0.015	0.030088649	0.005	0.06432901	0.5323
(2,1,1)	0.007522162	0.007522162	0.02	0.007522163	0.075	0.026277403	0.7137

Numeration of states represents a total number of customers in each node. To define a system, the authors make fifteen global-balance equations, and for each of them the left-hand side represents the flow out of state and the right-hand side represents the flow into state.

First, the authors wrote down local balance equations and searched the solutions applying substitution. This way the authors obtained exact steady-state probabilities, as shown in Table 1.

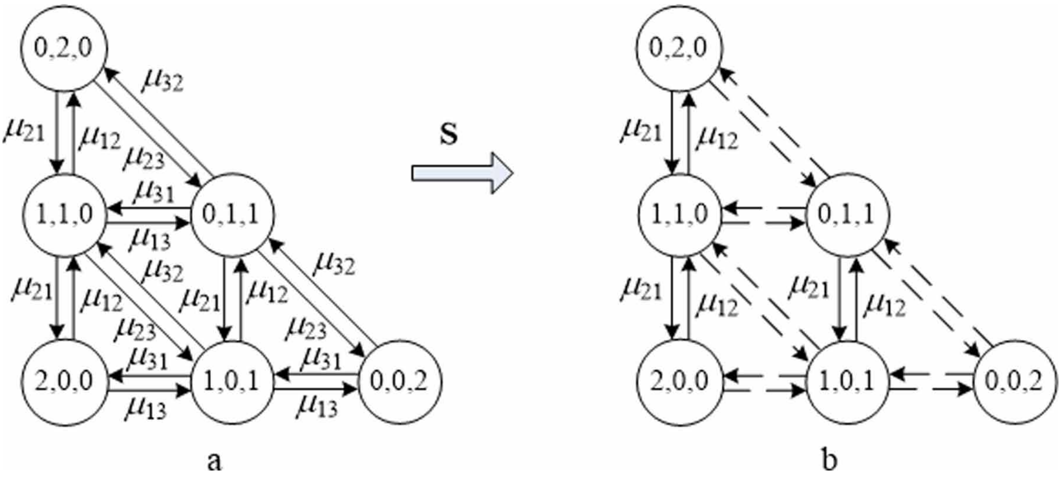
Next, the authors obtained results for steady-state probabilities with the use of Gaussian elimination technique - by applying Algorithm 1; iterative power procedure - by applying Algorithm 2; and Courtois's approximation technique – by applying Algorithm 3. A desired probability vector was not known in advance; therefore, an estimate of relative error was used.

Good convergence for iterative power procedure was obtained for $\boldsymbol{\nu}$=90 iterations. The worst results were obtained for Courtois's approximation technique because the involved aggregation procedure has been dependent on the matrix type. Comparing obtained and exact results, the authors demonstrated that advanced algorithms based on direct and iterative methods provided excellent results for closed networks with relative error $<10^{-6}$.

An approximation algorithm, based on aggregation and desegregation, was not suitable for the demonstrated example because the sum of elements around the main diagonal was close to 1, and elements positioned far from the main diagonal were not close to 0. This confirmed that aggregation and desegregation could be only applied to computations with limited number of queueing networks.

Then, the authors demonstrated basic principles of decomposition implemented in Algorithms 3 and 4 in an example with continuous-time Markov chain. The authors studied a closed queueing network with two customers (K=2) and three server stations (N=3) as shown in Figure 2(a). Each arbitrary pattern of customers was distributed among stations and was represented by a state. Consequently, in this example there were 6 states according to formula (17).

Figure 2. CTMC of queueing network (a) and its decomposition regarding service station three (b)



For instance, in state (2,0,0) two customers were at service station one, while service stations two and three were both empty. After an exponentially distributed time period, one customer travelled to service station two. Transition behavior was controlled using the transition rate μ_{12} .

Transition behavior between other states could be explained similarly. The lines pointed in the direction for possible transitions because only the counterclockwise direction was allowed in the queueing network.

At that point, the authors assumed that customers preferably stayed at two service stations and rarely relocated to the third service station, which was utmost isolated.

Decomposition is shown in Figure 2(b). Solid lines emphasize the tightly coupled states, while dotted lines represent the loose coupling. There were parameters with strong interaction between state 1 and 2, while state 3 interacted less with the others and their transition rates were $\mu_{12} = 4$; $\mu_{21} = 0$; $\mu_{13} = 0$; $\mu_{31} = 0.2$; $\mu_{23} = 0.4$; $\mu_{32} = 0$.

Corresponding infinitesimal generator matrix Q follows:

$$Q = \begin{pmatrix} -4 & 4 & 0 & 0 & 0 & 0 \\ 0 & -4.4 & 4 & 0.4 & 0 & 0 \\ 0 & 0 & -0.4 & 0 & 0.4 & 0 \\ 0.2 & 0 & 0 & -4.2 & 4 & 0 \\ 0 & 0.2 & 0 & 0 & -0.6 & 0.4 \\ 0 & 0 & 0 & 0.2 & 0 & -0.2 \end{pmatrix}$$

The decomposition and values for generator matrix Q are provided symbolically in the following configuration, as shown in Table 2.

This decomposition was used in a numerical example when computing the steady-state probability vector by applying Algorithms 3 and 4. The relative error between obtained and exact values was used to estimate precise solutions.

Additional comparative analysis of Algorithms 2, 3 and 4 was also made. Good convergence of numerical results was obtained after ν =200 iterations as shown in Table 3. Better results could be obtained after using more levels of decomposition as state probabilities may vary significantly.

Table 2. Decomposition and values for infinitesimal generator matrix Q

State	(2,0,0)	(1,1,0)	(0,1,0)	(1,0,1)	(0,1,1)	(0,0,2)
(2,0,0)	$-\Sigma$	μ_{12}	0	μ_{13}	0	0
(1,1,0)	μ_{21}	$-\Sigma$	μ_{12}	μ_{23}	μ_{13}	0
(0,2,0)	0	μ_{21}	$-\Sigma$	0	μ_{23}	0
(1,0,1)	μ_{31}	μ_{32}	0	$-\Sigma$	μ_{12}	μ_{13}
(0,1,1)	0	μ_{31}	μ_{32}	μ_{21}	$-\Sigma$	μ_{23}
(0,0,2)	0	0	0	μ_{31}	μ_{32}	$-\Sigma$

Table 3. Numerical results for steady-state probabilities computed with proposed algorithms for example 2

State	Exact Value	Algorithm 3		Algorithm 4			Algorithm 2	
		$\nu^{\approx}$	Error	$\nu^{(0)}$	$\nu^{(2)}$	Error	$\nu^{(200)}$	Error $\times 10^{-5}$
(2,0,0)	0.00136	0.0049	2.56	$0.16\bar{6}$	0.0015	0.0714	0.0013679	0.731
(1,1,0)	0.01368	0.0139	0.017	$0.16\bar{6}$	0.0137	0	0.013679	0
(0,2,0)	0.13679	0.1391	0.017	$0.16\bar{6}$	0.1378	0.0073	0.1367993	0.241
(1,0,1)	0.02736	0.0382	0.398	$0.16\bar{6}$	0.0273	0.0036	0.0273597	0.183
(0,1,1)	0.27359	0.2679	0.02	$0.16\bar{6}$	0.2739	0.0011	0.273598	0
(0,0,2)	0.54719	0.5358	0.02	$0.16\bar{6}$	0.5457	0.0026	0.547195	0.007

Lastly, the authors confirmed that Takahashi's iterative approximation with the use of Algorithm 4 provided very good results. After the first iteration step $\nu^{(1)}$, the results were not any close to exact values, but after the second iteration step $\nu^{(2)}$ they resembled exact values up to the third decimal digit. Comparison of running times of the decomposition and iterative algorithms could be obtained only in a case of significant number of steady states, i.e. not less than a hundred.

PROPOSED DOMAINS OF APPLICABILITY

The computational procedures for equilibria of queueing networks have been developed primarily for real systems with large number of resources, such as computer and communication networks, where these procedures are suitable for network dimensioning, management and traffic engineering. The increasing use of streaming services creates environments where these procedures are not only suitable but also of immediate interest, particularly, when based on high-speed internet, 4G and 5G wireless and mobile networks.

However, the authors consider the computational procedures for equilibria of queueing networks are also applicable to other domains where systems are described with comparable queuing models and stochastic methods are sufficiently relevant.

Several suitable domains of applicability are library (applied to modeling of the overall items and customer flows); hospital (applied to modeling of the overall patient flow); healthcare (applied to modeling of the human cardiovascular system); finance (applied to modeling of the equity/stock prices in a general equilibrium setting); blockchain systems (applied to modeling of the blockchain queue); basic and intelligent transportation systems (applied to modeling of the overall passenger flow); road traffic (applied to modeling of the overall flow of vehicles in a metropolitan area); call centers (applied to modeling of the overall customer calls); etc. Some of these examples have been discussed in the literature (Abel, 1994; De Bruin, Van Rossum, Visser and Koole, 2007; Li, Ma, Chang, Ma and Yu, 2019; Lokshina and Bartolacci, 2012; Lokshina, Gregus and Thomas, 2019; Lokshina, Zhong and Lanting, 2020; Shanmugasundaram and Umarani, 2015; Ukil, Jara and Mari, 2019; Zhu, Hu, Jiang and Khattak, 2017).

CONCLUSION

In this paper, the authors considered new computational procedures to obtain numerical steady-state solutions for queueing networks in equilibrium based on computing stochastic probability vectors of Markov chains. Advanced computational procedures for steady-state analysis of ergodic Markov chains with the use of the Gaussian elimination technique, power iteration procedure, Courtois's non-iterative approximation technique and Takahashi's iterative approximation procedure were developed and analyzed. Numerical examples for closed queueing networks with three nodes and exponentially distributed service rates were demonstrated. The obtained numerical results were analyzed and compared with exact results.

The obtained outcome can be summarized as follows:

- The algorithms based on direct techniques are effective; they modify parametric matrix and require fixed amounts of computational time. Computations with direct techniques provide excellent results with relative error less than 10^{-6} for a limited number of queueing network states. With increasing number of steady states of Markov chains, the convergence of algorithms based on direct techniques is expected to decrease.
- The convergence of iterative algorithms based on a power iteration procedure is more dependent on network performance parameters, such as throughput, response service time, and queueing length, and less dependent on the structure of queueing networks.
- The computational precision and convergence of algorithms, based on the aggregation and desegregation of Markov chains, are considerably dependent on selected decomposition and values of infinitesimal generator matrix. Therefore, implementation of these algorithms is limited to so-called small-scale queueing networks, with a small queue length and a small number of steady states.
- The computational procedures for equilibria of queueing networks are developed in the context of real communication networks; however, they are applicable also to other domains where systems are described with comparable queuing models and stochastic methods are sufficiently relevant. Some suitable domains of applicability are proposed.

ACKNOWLEDGMENT

The authors wish to extend their gratitude to the anonymous reviewers for their time, assistance, and patience.

REFERENCES

Abel, A. (1994). Exact solutions for expected rates of return under Markov regime switching: Implications for the equity premium puzzle. *Journal of Money, Credit and Banking*, *26*(3), 345–361. doi:10.2307/2078006

Bhalai, S. (2002). *Markov Decision Processes: The Control of High-Dimensional Systems*. Amsterdam, The Netherlands: Universal Press.

De Bruin, A., Van Rossum, A., Visser, M., & Koole, G. (2007). Modeling the emergency cardiac in-patient flow: An application of queuing theory. *Health Care Management Science*, *10*(: 2), 125–137. doi:10.1007/s10729-007-9009-8 PMID:17608054

Dugas, C., Chapados, N., Ducharme, R., Saint-Mleux, X., & Vincent, P. (2011). A high-order feature synthesis and selection algorithm applied to insurance risk modelling. *International Journal of Business Intelligence and Data Mining*, *6*(:3), 237–258. doi:10.1504/IJBIDM.2011.041957

Hassin, R., & Haviv, M. (2003). *To queue or not to queue: equilibrium behavior in queueing systems*. Boston, MA: Kluwer Academic Publishers. doi:10.1007/978-1-4615-0359-0

Hayes, J., & Ganesh Babu, T. (2004). *Modeling and Analysis of Telecommunications Networks*. New York: John Wiley and Sons. doi:10.1002/0471643505

Krieger, U., Muller-Clostermann, B., & Sczittnick, A. (1990). Modeling and analysis of communication systems based on computational methods for Markov chain. *IEEE Journal on Selected Areas in Communications*, *8*(: 9), 630–1648.

Li, Q., Ma, J., Chang, X., Ma, F., & Yu, H. (2019). Markov processes in Blockchain systems. *Computational Social Networks*, *6*(: 5). doi:10.1186/s40649-019-0066-1

Liu, W., Ma, Y., & Li, J. (2012). Equilibrium threshold strategies in observable queueing systems under single vacation policy. *Applied Mathematical Modelling*, *36*(1), 6186–6202. doi:10.1016/j.apm.2012.02.003

Liu, Z., Ma, Y., & Zhang, Z. (2015). Equilibrium mixed strategies in a discrete-time Markovian queue under multiple and single vacation policies. *Quality Technology & Quantitative Management*, *12*(3), 369–382. doi:10.1080/16843703.2015.11673387

Lokshina, I. (2016). Study on estimating probabilities of buffer overflow in high-speed communication networks. *Telecommunication Systems Journal*, *62*(2), 289–302. doi:10.1007/s11235-015-0055-0

Lokshina, I., & Bartolacci, M. (2007). Neural modelling and performance analysis of the link occupancy distribution for wireless broadband transmissions. *International Journal of Mobile Network Design and Innovation*, *2*(2), 129–133. doi:10.1504/IJMNDI.2007.015063

Lokshina, I., & Bartolacci, M. (2012). Accelerated rare event simulation with Markov chain modelling in wireless communication networks. *International Journal of Mobile Network Design and Innovation*, *4*(4), 185–191. doi:10.1504/IJMNDI.2012.054449

Lokshina, I., Gregus, M., Thomas, W. (2019). Application of Integrated Building Information Modeling, IoT and Blockchain Technologies in System Design of a Smart Building. *Procedia Computer Science*, *160*, 497-502. https://doi.org/10.1016/j.procs.2019.11.058

Lokshina, I., Zhong, H., & Lanting, C. (2020). Self-similar Teletraffic in a Smart World. In N. Kryvinska & M. Greguš (Eds.), *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies* (Vol. 30). Cham: Springer. doi:10.1007/978-3-030-19069-9_5

Mehdi, J. (2003). *Stochastic Models in Queueing Theory*. Academic Press.

Radev, D., Lokshina, I., & Denchev, V. (2007). Queueing networks in equilibrium and Markov chains: Numerical solution methods. *World Review of Entrepreneurship, Management and Sustainable Development*, *3*(: 3/4), 302–316. doi:10.1504/WREMSD.2007.014048

Radev, D., Lokshina, I., & Radeva, S. (2007). Evaluation of the queueing network equilibrium based on clustering analysis and self-organizing map. *World Review of Science, Technology and Sustainable Development*, *4*(2/3), 150–160. doi:10.1504/WRSTSD.2007.013580

Rubinstein, R., & Melamed, B. (1998). *Modern Simulation and Modeling*. New York: John Wiley & Sons.

Shanmugasundaram, S., & Umarani, P. (2015). Queueing theory applied in our day to day life. *International Journal of Scientific & Engineering Research, 6*(4). http://www.ijser.org

Stewart, W. (1994). *Introduction to Numerical Solution of Markov Chains*. Princeton University Press.

Trivedi, K. (2001). *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*. New York: John Willey & Sons.

Ukil, A., Jara, A., & Mari, L. (2019). Data-Driven Automated Cardiac Health Management with Robust Edge Analytics and De-Risking. *Sensors (Basel)*, *19*(12), 2733. doi:10.3390/s19122733 PMID:31216659

Zhang, Z. (2009). Performance analysis of a queue with congestion-based staffing policy. *Management Science*, *55*(2), 240–251. doi:10.1287/mnsc.1080.0914

Zhu, J., Hu, L., Jiang, Y., & Khattak, A. (2017). Circulation network design for urban rail transit station using a PH(n)/PH(n)/CIC queuing network model. *European Journal of Operational Research*, *260*(3), 1043–1068. doi:10.1016/j.ejor.2017.01.030

Izabella V. Lokshina, PhD, is Professor of MIS and chair of the Management, Marketing and Information Systems Department at SUNY Oneonta, USA. Her main research interests are intelligent information systems and communication networks, and complex system modeling and simulation.

Cees J. M. Lanting, PhD, is Senior Consultant at DATSA Belgium, Leuven, Belgium. His main research interests are smart communications and IoT.