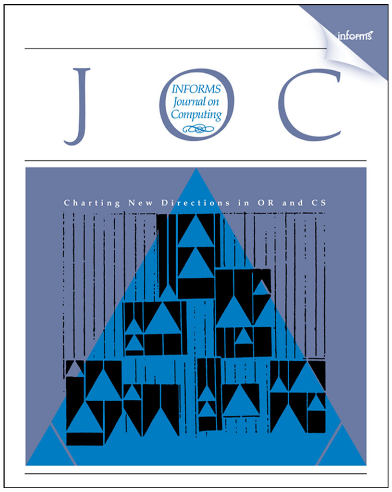
informs
J
INFORMS Journal on Computing
C
Charting New Directions in OR and CS

INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Space Reduction for a Class of Multidimensional Markov Chains: A Summary and Some Applications

Qi-Ming He, Attahiru Sule Alfa

To cite this article:
Qi-Ming He, Attahiru Sule Alfa (2018) Space Reduction for a Class of Multidimensional Markov Chains: A Summary and Some Applications. INFORMS Journal on Computing 30(1):1-10. https://doi.org/10.1287/ijoc.2017.0759

http://pubsonline.informs.org/journal/ijoc/

INFORMS JOURNAL ON COMPUTING
Vol. 30, No. 1, Winter 2018, pp. 1–10
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Space Reduction for a Class of Multidimensional Markov Chains: A Summary and Some Applications

Qi-Ming He,[a] **Attahiru Sule Alfa**[b, c]

[a] Department of Management Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada; [b] Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada; [c] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa

Contact: q7he@uwaterloo.ca, http://orcid.org/0000-0003-2381-3242 (Q-MH); attahiru.alfa@umanitoba.ca (ASA)

Received: July 7, 2016
Revised: December 1, 2016
Accepted: March 6, 2017
Published Online: November 1, 2017

https://doi.org/10.1287/ijoc.2017.0759

Copyright: © 2017 INFORMS

Abstract. In this paper, we present examples of a class of Markov chains that occur frequently, but whose associated matrices are a challenge to construct efficiently. These are Markov chains that arise as a result of several identical Markov chains running in parallel. Specifically for the cases considered, both the infinitesimal generator matrix for the continuous case, and more so the transition probability matrix for the discrete equivalent, are complex to construct effectively and efficiently. We summarize the algorithms for constructing the associated matrices and present examples of applications, ranging from special queueing problems to reliability issues and order statistics. MATLAB subroutines are provided in an online supplement for the implementation of the algorithms.

History: Accepted by Winfried Grassmann, Area Editor for Computational Probability and Analysis.
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2017.0759.

Keywords: Markov chains • order statistics • queues with multiple servers • reliability • phase-type distribution • Markovian arrival process

1. Introduction

Markov chains have been an indispensable tool in the analysis of many stochastic systems, such as queueing models and reliability models. While the theory on Markov chains has matured and has found broad applications, there are still some very important issues that are overlooked. The construction of the transition probability matrix or the infinitesimal generator matrix for independent identical Markov chains in parallel is one of them. On the one hand, the issue is well-known for years due to many applications. On the other hand, it is challenging to actually do the construction for even the moderate size of problems. Since the Markov chains being combined are independent and identical, researchers simply use the Kronecker product form and of course end up with a Markov chain with a huge state space; hence leading to a dimensionality problem. This approach leads to what we call TPFS (track-phase-for-server).

With the increasing computing power gained in the past decades, it turns out that one well-known approach, which we call CSFP (count-server-for-phase), for the problem leads to Markov chains that are manageable computationally. This idea is based on counting the number of Markov chains that are in each state, rather than keeping track of the state each Markov chain is as in the case of TPFS. However, constructing the CSFP Markov chain is a bit more challenging and has thus not received much attention from researchers. Thus we feel the need to put together a paper to summarize the algorithms developed for the approach.

Some examples of the types of problems we are focusing on are as follows. Consider the following:

1. A *MAP*/*PH*/*K* queueing system in which all the servers are identical: If the *MAP* is of dimension m_a and the *PH*-distribution is of dimension m_s, it is straightforward to study the queueing system by utilizing the phase of each server using the TPFS approach. However, that could lead to a huge state space of up to $m_a m_s^K$ states for the interior blocks of the resulting Markov chain. If we choose to use the CSFP approach, constructing the resulting Markov chain could be complex, but then the resulting state space will be not larger than $m_a(K+m_s-1)!/(K!(m_s-1)!)$, which is significantly smaller than $m_a m_s^K$. This is a huge difference in the state space required, especially as m_s and K increase. We note that similar queueing models such as the *MAP*/*PH*/*K*/*K* queue have the same issue that can be addressed in the same way.

2. A queueing system in which arrivals are from N identical sources, each generating traffic according to a *MAP* with dimension m_a: Suppose the resulting traffic is multiplexed (superimposed) to feed into a multiserver queueing system with K identical servers with *PH* service time distribution of dimension m_s. Again, we can use the TPFS to construct the resulting Markov chain, which would result in a huge state space; or we can use the CSFP that will result in a much smaller state space but more complex to construct.

3. A reliability problem in which we are studying the behavior of K identical components: Let us assume that the life of each component follows a PH distribution with dimension m_s. The study of the resulting Markov chain is the same as the idea in the first example given earlier.

4. The order statistics in which K independent identical PH random variables are of interest; let them be of order m_s: The resulting order statistics are PH distributions that can also be studied as the case in the first example.

These are just some examples in which this type of problem occurs, i.e., that of constructing a resulting Markov chain of a combination of several independent ones. The common thread with all these examples is that they are a result of several identical and independent Markov chains that are running in parallel. We will elaborate further on these examples later in the paper.

Meanwhile we present a brief summary of the state space explosion associated with the first example. Consider the case where $m_a = 1$ and $m_s = 2$; we show in Table 1 how the state space increases with K.

Example 1 (He and Alfa 2015)**.** If $m_s = 2$, the numbers of states of the two Markov Chains to be constructed by using the TPFS and CSFP approaches are given in Table 1.

With this type of fast growth in the state space for the TPFS approach, the idea of developing an efficient algorithm for constructing the CSFP Markov chain becomes important.

Focusing mainly on algorithms for the CSFP approach, we separate the continuous time case from the discrete time case, since the latter is significantly more challenging due to the occurrence of simultaneous events. We introduce basic algorithms and subroutines for the case with independent identical Markov chains in parallel. The construction process involved only matrix operations without any preprocessing of the state space. We then apply the basic subroutines to queueing models, superposition of Markov arrival processes, and order statistics. In addition, we add some simple state space complexity analysis and time complexity analysis. We also collect a few simple properties that can be used in debugging programs.

The remainder of the paper is organized as follows. In Section 2, we briefly review the existing literature. Algorithms for the continuous time case are summarized in Section 3, and the discrete time case is taken care of in Section 4. Section 5 concludes the paper.

2. Literature Review

We consider a Markov chain that consists of K independent identical Markov chains running in parallel. The problem of interest is to construct the infinitesimal generator (i.e., Q-matrix) or transition probability matrix (i.e., P-matrix) of the Markov chain. The straightforward approach TPFS can and has been used for that purpose. Unfortunately, the state space of the resulting Markov chain is so large that it becomes useless for real applications. Meanwhile, the CSFP approach has been known and used. The corresponding Markov chain has a significantly smaller state space. However, the construction of the transition blocks is challenging for the CSFP approach, especially for the discrete time case. Yet the use of CSFP depends largely on constructing the Q-matrix or P-matrix efficiently.

It is somehow surprising to see that the first formal algorithm on the problem is only dated back to Ramaswami (1985), even though the problem has been known long before that. Ramaswami (1985) introduced subroutines for the continuous time case. An algorithm for the discrete time case was introduced only recently in He and Alfa (2015). For the continuous case, He et al. (2017) developed an algorithm that is essentially equivalent to Ramaswami (1985).

Although the literature on CSFP is limited, applications of the Markov chain consisting of independent identical Markov chains running in parallel are enormous. For example, the distributions of the order statistics of independent identically distributed phase-type random variables can be constructed by using the CSFP approach, which is addressed in detail in Bladt and Nielsen (2017). However, the construction of the matrices for CSFP depends on that of TPFS through a transformation matrix that has to be constructed manually. The study of queueing systems with multiple identical servers is closely related to the CSFP approach. See examples in (i) Ramaswami and Lucantoni (1985), Asmussen and O'Cinneide (1998), Asmussen and Möller (2001), and Breuer et al. (2002) for some standard queueing models with multiple servers; (ii) Wagner (1997) for queues with multiple servers and service priority; (iii) Kim et al. (2012) for queues with multiple servers and retrials; and (iv) He et al. (2017) for queues with multiple servers and impatient customers. Finally, the idea of modeling communication traffic resulting from several sources multiplexed has been dealt with by different researchers in different ways, but mostly by approximations because of the huge Markov chains that could result. An

Table 1. Comparison of the Numbers of States for TPFS and CSFP

K	2	4	6	8	10	15	20	30
TPFS	4	16	64	256	1,024	32,768	1,048,576	1,073,741,824
CSFP	3	5	7	9	11	16	21	31

example of such problems was discussed by Thompson et al. (2001) and in several references therein. Most researchers at best resorted to using two-dimensional Markov chains as approximations for each source to avoid the dimensionality explosion that could result after multiplexing.

In the rest of this paper, we summarize a few basic algorithms introduced in Ramaswami (1985), He et al. (2017), and He and Alfa (2015). In addition, we use the basic algorithms to construct Markov chains associated with the *MAP*/*PH*/*K* queue (He and Alfa 2015), the superposition of independent identical Markovian arrival processes, and order statistics of phase-type distributions. We also add some simple analysis on state space and time complexity.

3. Continuous Time Markov Chains

The algorithm to be introduced in this section was first introduced in Ramaswami (1985). The version presented in this section is from He et al. (2017), which is essentially equivalent to that in Ramaswami (1985). An advantage of the construction process in He et al. (2017) is that it is self-proving. In this section, the basic algorithm is introduced in Section 3.1 and three applications are presented in Sections 3.2–3.4.

3.1. Independent Identical Continuous Time Markov Chains in Parallel

We consider K independent identical continuous time Markov chains (CTMCs), denoted as $\{X_k(t), t \geq 0\}$, for $k = 1, 2, \ldots, K$. Assume that these Markov chains have m_s states and the same Q-matrix S. Putting the K Markov chains together, we obtain a new Markov chain $\{(X_1(t), \ldots, X_K(t)), t \geq 0\}$. It is easy to see that the new Markov chain has m_s^K states and Q-matrix

$$Q^{(\mathrm{TPFS})}_{(1,\ldots,K)} = S \oplus S \oplus \cdots \oplus S, \tag{1}$$

where $\oplus$ is for the Kronecker sum of matrices. The Markov chain is formulated by the TPFS approach. The state space of the Markov chain becomes too large even for moderate K and m_s (see Table 1). Thus, the other well-known approach CSFP is usually used.

Define $N_j(t)$ the number of the K original CTMCs in phase j at time t. Then it is easy to see that $\{(N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$ is also a CTMC. This new Markov chain contains the same information as the other one, if it is unnecessary to distinguish the K original Markov chains.

Proposition 3.1. *Random variables* $\{N_1(t), \ldots, N_{m_s}(t)\}$ *satisfy* $N_1(t) + \cdots + N_{m_s}(t) = K$, *for* $t \geq 0$. *The number of states of CTMC* $\{(N_1(t), \ldots, N_{m_s}(t)),\ t \geq 0\}$ *is* $(K + m_s - 1)!/(K!(m_s - 1)!)$.

Next, we introduce an algorithm for the construction of the Q-matrix for $\{(N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$. The idea is to divide the Q-matrix into smaller blocks and build those smaller blocks, which can be realized by decomposing the state space of the CTMC. Specifically, we decompose the state space according to $N_{m_s}(t)$, which takes values $\{0, 1, \ldots, K\}$. Since the original Markov chains are continuous in time, it is easy to see that $N_{m_s}(t)$ can increase or decrease at most by one at each transition. This leads to a quasi birth-and-death (QBD) structure in the Q-matrix of $\{(N_1(t), \ldots, N_{m_s-1}(t), N_{m_s}(t)), t \geq 0\}$, where the level variable $N_{m_s}(t)$ records the number of Markov chains in phase m_s and the phase variable $(N_1(t), \ldots, N_{m_s-1}(t))$ records the numbers of Markov chains in phases $\{1, 2, \ldots, m_s - 1\}$ at time t. Define, for $1 \leq k \leq K$ and $1 \leq m \leq m_s$,

$$\Omega(k, m) = \left\{ (n_1, \ldots, n_m) \colon n_i \geq 0, i = 1, 2, \ldots, m, \sum_{i=1}^{m} n_i = k \right\}. \tag{2}$$

The state space of $\{(N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$ is $\Omega(K, m_s)$, which can be decomposed as

$$\begin{aligned} \Omega(K, m_s) = &(\Omega(K, m_s - 1) \times \{0\}) \\ &\cup (\Omega(K - 1, m_s - 1) \times \{1\}) \\ &\cup \cdots \cup (\Omega(0, m_s - 1) \times \{K\}). \end{aligned} \tag{3}$$

Based on the decomposition of the state space, transitions of the Markov chain can be classified as follows.

1. *In-coming* transitions from m_s to $\{1, 2, \ldots, m_s - 1\}$: $N_{m_s}(t)$ is decreased by one;
2. Transitions within $\{1, 2, \ldots, m_s - 1\}$: $N_{m_s}(t)$ remains the same;
3. *Out-going* transitions from $\{1, 2, \ldots, m_s - 1\}$ to m_s: $N_{m_s}(t)$ is increased by one.

Based on the above decomposition and classification, the Q-matrix of $\{(N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$ can be written as Equation (4) in Figure 1 where $\mathbf{u} = S[m_s, 1{:}m_s - 1]$ and $\mathbf{v} = S[1{:}m_s - 1,\ m_s]$. It is clear from Equation (4) in Figure 1 that, to obtain $Q(K, m_s)$, we need to construct three sets of matrices:

(i) $\{Q_u^+(k, m_s - 1), k = 0, 1, \ldots, K - 1\}$: The transition blocks for in-coming transitions;

(ii) $\{Q_v^-(k, m_s - 1), k = 1, \ldots, K\}$: The transition blocks for out-going transitions;

(iii) $\{Q(k, m_s - 1), k = 0, 1, \ldots, K\}$: The transition blocks for transitions within $\{1, 2, \ldots, m_s - 1\}$.

Next, three subroutines are developed for that purpose.

Subroutine QPlus( ): Construction of $Q_u^+(k, m)$ with given k, m, and row vector $\mathbf{u}$, which is for the transition rates that the number of processes in phases $\{1, 2, \ldots, m\}$ *increases* by one (i.e., in-coming transitions). Similar to Equation (4), by decomposing the state set $\Omega(k, m)$ according to the value of $N_m(t)$, we obtain Equation (5) in Figure 2 where (i) $Q_u^+(0, m) = \mathbf{u}[1{:}m]$, $m = 1, 2, \ldots, m_s$, and (ii) $Q_u^+(k, 1) = u_1, k = 0, 1, \ldots, K - 1$. It is clear from (5) in Figure 2 that, to

Figure 1. Matrix $Q(k,m_s)$

$$Q(K,m_s)=\left(\begin{array}{cccccc}
 & \Omega(K,m_s-1)\times\{0\} & \Omega(K-1,m_s-1)\times\{1\} & \cdots & \Omega(1,m_s-1)\times\{K-1\} & \Omega(0,m_s-1)\times\{K\} \\
\Omega(K,m_s-1)\times\{0\} & Q(K,m_s-1) & Q_v^-(K,m_s-1) & & & \\
\Omega(K-1,m_s-1)\times\{1\} & Q_u^+(K-1,m_s-1) & Q(K-1,m_s-1) & Q_v^-(K-1,m_s-1) & & \\
\vdots & & \ddots & \ddots & \ddots & \\
\Omega(1,m_s-1)\times\{K-1\} & & & (K-1)Q_u^+(1,m_s-1) & Q(1,m_s-1) & Q_v^-(1,m_s-1) \\
\Omega(0,m_s-1)\times\{K\} & & & & KQ_u^+(0,m_s-1) & Q(0,m_s-1)
\end{array}\right)$$

$$+\left(\begin{array}{cccccc}
 & \Omega(K,m_s-1)\times\{0\} & \Omega(K-1,m_s-1)\times\{1\} & \cdots & \Omega(1,m_s-1)\times\{K-1\} & \Omega(0,m_s-1)\times\{K\} \\
\Omega(K,m_s-1)\times\{0\} & 0\times I & & & & \\
\Omega(K-1,m_s-1)\times\{1\} & & S(m_s,m_s)I & & & \\
\vdots & & & \ddots & & \\
\Omega(1,m_s-1)\times\{K-1\} & & & & (K-1)S(m_s,m_s)I & \\
\Omega(0,m_s-1)\times\{K\} & & & & & KS(m_s,m_s)
\end{array}\right) \tag{4}$$

Figure 2. Matrix $Q_u^+(k,m)$

$$Q_u^+(k,m)=\left(\begin{array}{cccccc}
 & \Omega(k+1,m-1)\times\{0\} & \Omega(k,m-1)\times\{1\} & \cdots & \Omega(1,m-1)\times\{k\} & \Omega(0,m-1)\times\{k+1\} \\
\Omega(k,m-1)\times\{0\} & Q_u^+(k,m-1) & u_mI & & & \\
\Omega(k-1,m-1)\times\{1\} & & Q_u^+(k-1,m-1) & u_mI & & \\
\vdots & & \ddots & \ddots & \ddots & \\
\Omega(1,m-1)\times\{k-1\} & & & Q_u^+(1,m-1) & u_mI & \\
\Omega(0,m-1)\times\{k\} & & & & Q_u^+(0,m-1) & u_m
\end{array}\right) \tag{5}$$

Figure 3. Matrix $Q_v^-(k,m)$

$$Q_v^-(k,m)=\left(\begin{array}{cccccc}
 & \Omega(k-1,m-1)\times\{0\} & \Omega(k-2,m-1)\times\{1\} & \cdots & \Omega(1,m-1)\times\{k-2\} & \Omega(0,m-1)\times\{k-1\} \\
\Omega(k,m-1)\times\{0\} & Q_v^-(k,m-1) & & & & \\
\Omega(k-1,m-1)\times\{1\} & v_mI & Q_v^-(k-1,m-1) & & & \\
\vdots & & \ddots & \ddots & & \\
\vdots & & & \ddots & \ddots & \\
\Omega(1,m-1)\times\{k-1\} & & & & (k-1)v_mI & Q_v^-(1,m-1) \\
\Omega(0,m-1)\times\{k\} & & & & & kv_m
\end{array}\right) \tag{6}$$

obtain $Q_u^+(k,m)$, we need to first find similar matrices with smaller k and/or m. A recursive procedure follows from the observation immediately, which leads to subroutine QPlus( ) for $Q_u^+(k,m)$. MATLAB-codes for QPlus( ) are given in Table A.1 in the online supplement.

Subroutine QMinus( ): Construction of $Q_v^-(k,m)$ for given k, m, and column vector v, which is for the transition rates that the number of processes in phases $\{1,2,\dots,m\}$ decreases by one (i.e., out-going transitions). Similar to $Q_u^+(k,m)$, $Q_v^-(k,m)$ can be written as (6) in Figure 3 where (i) $Q_v^-(1,m)=\mathbf{v}[1{:}m]$, $m=1,2,\dots,m_s$, and (ii) $Q_v^-(k,1)=kv_1$, $k=1,\dots,K$. Subroutine QMinus( ) for $Q_v^-(k,m)$ is based on the structure of the matrix given in Equation (6) in Figure 3. MATLAB-codes for QMinus( ) are given in Table A.2 in the online supplement.

Subroutine Qkm( ): Construction of $Q(k,m)$ with given k, m, and matrix $S[1{:}m,1{:}m]$. We note that $Q(0,m)=0$ and $Q(k,1)=kS(1,1)$. The subroutine Qkm( ) for $Q(k,m)$ is based on the structure of the matrix given in Equation (4) (Note: use $\{k,m,S[1{:}m,1{:}m]\}$ to replace $\{K,m_s,S\}$ in Equation (4).) Similar to $Q_u^+(k,m)$ and $Q_v^-(k,m)$, to find $Q(k,m)$, we need to construct all such matrices with smaller k and m. As shown in (4), subroutines QPlus( ) and QMinus( ) are to be called in Qkm( ). MatLab-codes for Qkm( ) are given in Table A.3 in the online supplement.

Finally, we call Qkm( ) with $\{K,m_s,S\}$ to obtain $Q(K,m_s)$ for $\{(N_1(t),\dots,N_{m_s}(t)),\ t\ge 0\}$.

Subroutine SPhi( ): Assume that the original CTMCs are ergodic with stationary distribution $\theta=(\theta_1,\theta_2,\dots,\theta_{m_s})$, i.e., $\boldsymbol{\theta}S=0$ and $\boldsymbol{\theta}\mathbf{e}=1$. Since the K original CTMCs are independent and parallel, the stationary distribution $\boldsymbol{\phi}$ of $\{(N_1(t),\dots,N_{m_s}(t)),\ t\ge 0\}$ has a multinomial distribution:

$$\phi(\mathbf{n})=\frac{K!}{n_1!\dots n_{m_s}!}\prod_{j=1}^{m_s}\theta_j^{n_j},\quad \text{for } \mathbf{n}=(n_1,\dots,n_{m_s})\in\Omega(K,m_s). \tag{7}$$

A simple procedure can be developed for computing $\boldsymbol{\phi}$. MATLAB-codes for SPhi( ) are given in Table A.4 in the online supplement.

Verification. The following relationships are useful for debugging subroutines and for checking the correctness of the algorithm: (i) $Q(K, m_s)\mathbf{e} = 0$; and (ii) $\boldsymbol{\phi} Q(K, m_s) = 0$ and $\boldsymbol{\phi}\mathbf{e} = 1$.

3.2. Continuous Time *MAP/PH/K* Queue

The queueing model has K identical servers and a single queue. The customer arrival process and service time are defined explicitly as follows.

- Arrivals: Continuous time Markovian arrival process (D_0, D_1), where D_0 and D_1, are square matrices of order m_a. Intuitively, D_0 contains the transition rates without an arrival and D_1 contains the transition rates with one arrival. The stationary distribution $\boldsymbol{\theta}_a$ of the underlying Markov chain of the arrival process satisfies $\boldsymbol{\theta}_a(D_0 + D_1) = 0$ and $\boldsymbol{\theta}_a\mathbf{e} = 1$. (See Neuts 1979 for more detail.)
- Service: Continuous time phase-type distribution with *PH*-representation $(\boldsymbol{\beta}, S)$ of order m_s. Intuitively, $\boldsymbol{\beta}$ is the initial distribution of the underlying CTMC of the phase-type distribution, and S is the subgenerator of transitions between nonabsorbing phases. The stationary distribution θ_s of the underlying CTMC satisfies $\boldsymbol{\theta}_s(S + \mathbf{S}^0\boldsymbol{\beta}) = 0$ and $\boldsymbol{\theta}_s\mathbf{e} = 1$. (See Neuts 1981 for more detail.)

Define $q(t)$ the queue length at time t. We introduce two CTMCs that can be used to analyze $q(t)$. Let $I_a(t)$ be the phase of the underlying Markov chain of the Markovian arrival process, $I_{s,k}(t)$ the phase of the underlying Markov chain associated with the service time of the k-th server (which is working), at time t. It is easy to see that $\{(q(t), I_a(t), I_{s,1}(t), \ldots, I_{s,\min\{q(t),K\}}(t)), t \geq 0\}$ is a CTMC. This is called the TPFS approach since the process is defined by *Tracking the Phase For the underlying Markov chain of individual Servers*. This CTMC is of the *QBD*-type, for which $q(t)$ is the level variable and $(I_a(t), I_{s,1}(t), \ldots, I_{s,\min\{q(t),K\}}(t))$ the phase variable of the QBD process.

Let $N_j(t)$ be the number of servers whose service phase is j at time t. That $\{N_1(t), \ldots, N_{m_s}(t)\}$ are similar to that in Section 3.1, except that $N_1(t) + \cdots + N_{m_s}(t)$ is the total number of servers in service at time t and it takes values $\{0, 1, \ldots, K\}$. It is also easy to see that $\{(q(t), I_a(t), N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$ is a CTMC. This is called the CSFP approach since it *Counts the numbers of Servers For individual Phases*. This CTMC is also of the *QBD*-type, for which $q(t)$ is the level variable and $(I_a(t), N_1(t), \ldots, N_{m_s}(t))$ the phase variable.

For both the TPFS and CSFP approaches, the Q-matrix of the associated CTMC has the QBD structure and can be rewritten as follows:

$$Q = \begin{pmatrix} A_{0,0} & A_{0,1} & & & & & & \\ A_{1,0} & A_{1,1} & A_{1,2} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & A_{K-1,K-2} & A_{K-1,K-1} & A_{K-1,K} & & & \\ & & & A_{K,K-1} & A_1 & A_0 & & \\ & & & & A_2 & A_1 & A_0 & \\ & & & & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{8}$$

For the CSFP approach, the transition blocks can be obtained as follows:

$$\begin{aligned} A_{0,0} &= D_0, \quad A_{0,1} = D_1 \otimes P^+(0, m_s); \\ A_{k,k-1} &= I \otimes Q^-(k, m_s), \quad A_{k,k} = D_0 \oplus Q(k, m_s), \\ A_{k,k+1} &= D_1 \otimes P^+(k, m_s); \\ A_2 &= I \otimes (Q^-(K, m_s)P^+(K-1, m_s)), \\ A_1 &= D_0 \oplus Q(K, m_s), \quad A_0 = D_1 \otimes I, \end{aligned} \tag{9}$$

where $\otimes$ is for the Kronecker product of matrices, $P^+(k, m_s)$ can be constructed by calling QPlus( ) with $\{k, m_s, \boldsymbol{\beta}\}$; $Q^-(k, m_s)$ by calling QMinus( ) with $\{k, m_s, \mathbf{S}^0\}$; and $Q(k, m_s)$ by calling Qkm( ) with $\{k, m_s, S\}$. Intuitively, matrix $P^+(k, m_s)$ is for the phase change after the arrival of a customer and $Q^-(k, m_s)$ is for the phase change after a service completion.

For the TPFS approach, the order of transition blocks $\{A_0, A_1, A_2\}$ is $m_a m_s^K$. For the CSFP approach, the order of $\{A_0, A_1, A_2\}$ is $m_a(K + m_s - 1)!/(K!(m_s - 1)!)$. Consequently, the levels of the QBD process corresponding to CSFP is much smaller than that of TPFS.

Verification. The following property can be useful for debugging and checking correctness of subroutines.

Proposition 3.2. *Let $A = A_0 + A_1 + A_2$. Then $\boldsymbol{\theta}_a \otimes \boldsymbol{\phi}$ is the stationary distribution of A, where $\boldsymbol{\phi}$ can be obtained by calling SPhi( ) with $\{K, m_s, \boldsymbol{\theta}_s\}$. In addition, we have $(\boldsymbol{\theta}_a \otimes \boldsymbol{\phi})A_0\mathbf{e} = \boldsymbol{\theta}_a D_1\mathbf{e}$, the arrival rate, and $(\boldsymbol{\theta}_a \otimes \boldsymbol{\phi})A_2\mathbf{e} = K\boldsymbol{\theta}_s\mathbf{S}_0$, the maximum service rate.*

3.3. Superposition of Continuous Time Markovian Arrival Processes

We consider K independent identical continuous time *MAP*s $\{\{(W_k(t), I_k(t)), t > 0\}, k = 1, 2, \ldots, K\}$ with common matrix-representation (D_0, D_1) of order m_a, where $W_k(t)$ is the number of arrivals in $[0, t]$ and $I_k(t)$ is the phase of the underlying CTMC at time t of the k-th *MAP*. Define $W(t) = W_1(t) + \cdots + W_K(t)$. It is easy to see that $\{(W(t), I_1(t), \ldots, I_K(t)), t > 0\}$ is a *MAP* with matrix-representation (C_0, C_1) of order m_a^K, where $C_0 = D_0 \oplus \cdots \oplus D_0$ and $C_1 = D_1 \oplus \cdots \oplus D_1$. Following the CSFP approach, it is easy to see that $\{(W(t), N_1(t), \ldots, N_{m_s}(t)), t > 0\}$ is a *MAP* with matrix-representation (C_0, C_1) of order $(K + m_a - 1)/(K!(m_a - 1)!)$, where C_0 can be obtained by calling Qkm( ) with $\{K, m_a, D_0\}$ and C_1 can be obtained by calling Qkm( ) with $\{K, m_a, D_1\}$. The correctness of (C_0, C_1) can be verified by computing and comparing the average arrival rates of the *MAP*s.

3.4. Order Statistics of Continuous Phase-Type Random Variables

Consider K independent identical phase-type random variables $\{X_1, \dots, X_K\}$ with a common *PH*-representation $(m_s, \boldsymbol{\beta}, S)$ and $\mathbf{S}^0 = -S\mathbf{e}$. Denote by $\{X_{(1)}, \dots, X_{(K)}\}$ the order statistics of $\{X_1, \dots, X_K\}$ with $X_{(K)} = \min\{X_1, \dots, X_K\}$ and $X_{(1)} = \max\{X_1, \dots, X_K\}$. It is well-known that the order statistics are also phase-type distributions (Neuts 1981).

Define $I_k(t)$ the phase of the underlying CTMC associated with X_k, $k = 1, 2, \dots, K$. The state space of $I_k(t)$ is $\{1, \dots, m_s, m_s + 1\}$. Define $q(t) = \sum_{k=1}^K 1_{\{I_k(t) < m_s + 1\}}$. Then $q(t)$ is decreasing, each time by one. It is easy to see that $\{(q(t), I_1(t), \dots, I_K(t)), t \geq 0\}$ is a CTMC (the TPFS approach) and the number of states of the Markov chain is $1 + m_s + m_s^2 + \cdots + m_s^K$. It is also readily seen that $\{(q(t), N_1(t), \dots, N_{m_s}(t)), t \geq 0\}$ is a CTMC (the CSFP approach) with $\sum_{k=1}^K (k + m_s - 1)!/(k!\,(m_s - 1)!)$ states. Both processes are of the pure death-type with Q-matrix of the form:

$$\begin{array}{c} 0\\ 1\\ \vdots\\ K-1\\ K \end{array} \begin{pmatrix} Q_{0,0} & & & & \\ Q_{1,0} & Q_{1,1} & & & \\ & \ddots & \ddots & & \\ & & Q_{K-1,K-2} & Q_{K-1,K-1} & \\ & & & Q_{K,K-1} & Q_{K,K} \end{pmatrix}, \qquad (10)$$

where $Q_{0,0} = 0$.

For the TPFS approach, we have $Q_{k,k} = S \oplus S \oplus \cdots \oplus S$ and $Q_{k,k-1} = \sum_{j=1}^k I_{j-1} \otimes \mathbf{S}^0 \otimes I_{k-j}$, where I_n is the identity matrix of order m_s^n. For the CSFP approach, $Q_{k,k}$ is obtained by calling Qkm( ) with $\{k, m_s, S\}$ and $Q_{k,k-1}$ is obtained by calling QMinus( ) with $\{k, m_s, \mathbf{S}^0\}$.

The relationship between $q(t)$ and $\{X_{(1)}, \dots, X_{(K)}\}$ is

$$X_{(k)} = \min\{t: q(t) = k - 1\}, \quad \text{for } k = 1, 2, \dots, K. \qquad (11)$$

Based on the above relationship and the definition of phase-type distribution, the *PH*-representations of $\{X_{(1)}, \dots, X_{(K)}\}$ can be obtained as

(a) $X_{(K)}$: $(\boldsymbol{\alpha}(K), T(K))$, where $\boldsymbol{\alpha}(K) = \text{SPhi}(K, m_s, \boldsymbol{\beta})$ and $T(K) = Q_{K,K}$;

(b) $X_{(k)}$: $(\boldsymbol{\alpha}(k), T(k))$, where $\boldsymbol{\alpha}(k) = (0, \boldsymbol{\alpha}(k+1))$; and

$$T(k) = \begin{pmatrix} Q_{k,k} & 0 \\ \begin{pmatrix} Q_{k+1,k} \\ 0 \end{pmatrix} & T(k+1) \end{pmatrix}, \quad \text{for } k = K-1, K-2, \dots, 1.$$

The number of phases of the *PH*-representation $(\boldsymbol{\alpha}(k), T(k))$ is $\sum_{j=k}^K (j + m_s - 1)!/(j!\,(m_s - 1)!)$, for $k = 1, \dots, K$.

4. Discrete Time Markov Chains

The discrete time case is significantly more complex than its continuous counterpart. The reason is that transitions of individual Markov chains can occur simultaneously for the discrete case. To handle this issue, we decompose each transition of the constructed Markov chain into transitions of individual original Markov chains. This section is organized similarly to Section 3.

4.1. Independent Identical Discrete Time Markov Chains in Parallel

We consider K independent identical discrete time Markov chains (DTMC), denoted as $\{X_k(t),\ t = 0, 1, 2, \dots\}$, for $k = 1, 2, \dots, K$. Assume that these Markov chains have m_s states and the same P-matrix S. Putting the K Markov chains together, we obtain a new Markov chain $\{(X_1(t), \dots, X_K(t)), t = 0, 1, 2, \dots\}$. It is easy to see that the new Markov chain has m_s^K states and P-matrix $P_{(1,\dots,K)}^{(\text{TPFS})} = S \otimes S \otimes \cdots \otimes S$.

Define $N_j(t)$ the number of DTMCs in phase j at time t. Then it is easy to see that $\{(N_1(t), \dots, N_{m_s}(t)), t = 0, 1, 2, \dots\}$ is also a DTMC. This Markov chain contains the same information as $\{(X_1(t), \dots, X_K(t)), t = 0, 1, 2, \dots\}$, if it is unnecessary to distinguish the K original Markov chains. The state space of the Markov chain is $\Omega(K, m_s)$. Define, for $q = 0, 1, \dots, K$, and $m = 1, \dots, m_s$,

- $P(q, m) = P(\Omega(q, m)\colon \Omega(q, m))$: The one-step transition matrix from the set $\Omega(q, m)$ to $\Omega(q, m)$, given that the transitions within the m phases are governed by $S[1{:}m, 1{:}m]$.

For $P(q, m)$, only phase changes within phases $\{1, 2, \dots, m\}$ are considered. Apparently, we have $P(0, m) = 1$; $P(1, m) = S[1{:}m, 1{:}m]$; and $P(k, 1) = s_{1,1}^k$.

Our target is $P(K, m_s)$. Repeating the idea to decompose the state space $\Omega(K, m_s)$ according to the value of $N_{m_s}(t)$ (see Equation (3)), matrix $P(K, m_s)$ can be decomposed into sub-blocks:

$$P(K, m_s) = \begin{array}{c} \\ \vdots \\ \Omega(j, m_s - 1) \times \{K - j\} \\ \vdots \end{array} \begin{array}{c} \cdots \quad \Omega(q, m_s - 1) \times \{K - q\} \quad \cdots \\ \begin{pmatrix} \vdots & \cdots & \vdots \\ \cdots & P_{u,v}(\Omega(j, m_s - 1) \times \{K - j\}\colon \Omega(q, m_s - 1) \times \{K - q\}) & \cdots \\ \vdots & \cdots & \vdots \end{pmatrix} \end{array}, \qquad (12)$$

where $\mathbf{u} = S[m_s, 1{:}m_s - 1]$, $\mathbf{v} = S[1{:}m_s - 1, m_s]$, $P_{u,v}(\Omega(j, m_s - 1) \times \{K-j\} \colon \Omega(q, m_s - 1) \times \{K-q\})$ is the one-step transition block from $\Omega(j, m_s - 1) \times \{K-j\}$ to $\Omega(q, m_s - 1) \times \{K-q\}$. To find those transition blocks, similar to the continuous time case, we classify transitions into three types based on $N_{m_s}(t)$: (i) out-going transitions from phases $\{1,2,\ldots,m_s - 1\}$ to phase m_s, (ii) transitions within $\{1,2,\ldots,m_s - 1\}$, and (iii) in-coming from m_s to $\{1,2,\ldots,m_s - 1\}$. Different from the continuous time case, all K Markov chains can change their phase at the same time.

In general, $P(k,m)$ is the transition matrix within $\Omega(k,m)$ and has the same decomposition as shown in Equation (). That is, we can decompose the state set $\Omega(k,m)$ and obtain $P_{u,v}(\cdot,\cdot)$ for a process consists of k independent processes with m phases. For each transition, some original Markov chains enter $\{1,2,\ldots, m-1\}$, and some leave the set to $\{m\}$. To find transition block $P_{u,v}(\cdot,\cdot)$, we need to know the exact number of out-going/in-coming transitions. For example, given a transition from $\Omega(j, m-1) \times \{k-j\}$ to $\Omega(q, m-1) \times \{k-q\}$, if $j > q$, the out-going and in-coming transitions can be: $j-q$ (out-going) and 0 (in-coming); $j-q+1$ and 1;…, and $\min\{j, k-q\}$ and $\min\{q, k-j\}$. Thus, if the number of out-going transitions is given, the number of in-coming transitions is known as well. Define

- $P_{u,v}(q, j, m-1 \mid k) = P_{u,v}(\Omega(q, m-1) \colon \Omega(j, m-1) \mid k)$: The one-step transition matrix from the set $\Omega(q, m-1)$ to $\Omega(j, m-1)$, given that there are exactly k out-going transitions, and the transitions within the $m-1$ phases are governed by $S[1{:}m-1, 1{:}m-1]$, the in-coming transitions are determined by probabilities in vector $\mathbf{u}$, and out-going transitions are determined by probabilities in vector $\mathbf{v}$.

Now, the transitions from $\Omega(j, m-1) \times \{k-j\}$ to $\Omega(q, m-1) \times \{k-q\}$ can be categorized according to the number of out-going transitions, which is between $\max\{0, j-q\}$ and $\min\{j, k-q\}$. If the out-going transition number is l, then there are exactly $l+q-j$ original Markov chains going from phase m into phases $\{1,2,\ldots,m-1\}$. Thus, out of the $k-j$ Markov chains in phase m, exactly $k-q-l$ of them remain in phase m. Note that the probability of a Markov chain to remain in phase m is $s_{m,m}$. The corresponding probability is $\binom{k-j}{k-q-l}(s_{m,m})^{k-q-l}$. Conditioning on the number of out-going transitions, we obtain

$$P_{S[m,1:m-1],S[1:m-1,m]}(\Omega(j,m-1)\times\{k-j\} \colon \Omega(q,m-1)\times\{k-q\}) = \sum_{l=\max\{0,j-q\}}^{\min\{j,k-q\}} P_{S[m,1:m-1],S[1:m-1,m]}(j,q,m-1 \mid l) \cdot \binom{k-j}{k-q-l}(s_{m,m})^{k-q-l}. \tag{13}$$

Next, we decompose the transition with exactly k out-going transitions into: (1) k out-going transitions to m; (2) transitions within $\Omega(q, m-1)$; and (3) $j-q+k$ in-coming transitions. Define

- $L_v^-(q+j, q, m-1) = L_v^-(\Omega(q+j, m-1) \colon \Omega(q, m-1))$: The one-step transition matrix from the set $\Omega(q+j, m-1)$ to $\Omega(q, m-1)$ only due to j out-going transitions (i.e., leaving $\{1,2,\ldots,m-1\}$), given that out-going transitions are determined by probabilities in column vector $\mathbf{v}$. (Note that no other type of phase change is considered.)
- $L_u^+(q, q+j, m-1) = L_u^+(\Omega(q, m-1) \colon \Omega(q+j, m-1))$: The one-step transition matrix from the set $\Omega(q, m-1)$ to $\Omega(q+j, m-1)$ only due to j in-coming transitions (i.e., going into $\{1,2,\ldots,m-1\}$), given that in-coming transitions are determined by probabilities in row vector $\mathbf{u}$.

Although the three types of transitions occur simultaneously, we consider them in the following order: out-going, within the current set, and in-coming. Then we handle them separately to obtain

$$\begin{aligned} &P_{u,v}(q,q,m-1 \mid 0) = P(q, m-1), \quad \text{for } q = 1,2,\ldots,K;\\ &P_{u,v}(q,j,m-1 \mid k)\\ &\quad = L_v^-(q, q-k, m-1) P(q-k, m-1) L_u^+(q-k, j, m-1),\\ &\qquad \text{for } k \le q \le k+j. \end{aligned} \tag{14}$$

For the first equation in Equation (14), there is no out-going transition (and no in-coming transition). All transitions are internal between phases $\{1,2,\ldots, m-1\}$. Thus, vectors $\{\mathbf{u}, \mathbf{v}\}$ are not useful and the transition matrix should be $P(q, m-1)$. For the second equation in Equation (14), we first consider the k out-going transitions from $\{1,2,\ldots,m-1\}$ to an outside phase according to $\mathbf{v}$, which are recorded in $L_v^-(q, q-k, m-1)$; then all the transitions of the remaining $q-k$ Markov chains within phases $\{1,2,\ldots,m-1\}$, which are recorded in $P(q-k, m-1)$; and $j-(q-k)$ in-coming transitions from an outside phase into $\{1,2,\ldots,m-1\}$ according to $\mathbf{u}$, which is recorded in $L_u^+(q-k, j, m-1)$.

Further, for the k out-going transitions and the $j-(q-k)$ in-coming transitions, we decompose them into a sequence of single out-going or in-coming transitions:

$$\begin{aligned} L_v^-(q+k, q, m-1) &= \frac{1}{k!} \prod_{j=q+k}^{q+1} L_v^-(j, j-1, m-1), \quad \text{for } k, q \ge 0;\\ L_u^+(q, q+k, m-1) &= \prod_{j=q}^{q+k-1} L_u^+(j, j+1, m-1), \quad \text{for } k, q \ge 0. \end{aligned} \tag{15}$$

Now, we have reached at $L_v^-(j, j-1, m)$ and $L_u^+(j, j+1, m)$ (Note: for convenience, we reset $m-1$ to m), in which only one out-going/in-coming transition is involved. Similar to $Q_v^-(k,m)$ and $Q_u^+(k,m)$ in Section 3.1, simple recursive relationships exist for $L_v^-(j, j-1, m)$ and $L_u^+(j, j+1, m)$, which lead to recursive methods to compute those matrices. Using the above relationships, three subroutines are developed for computing $P(K, m_s)$.

Subroutine PPlus( ): For given $\{k, m, \mathbf{u}\}$, we find $L_u^+\{k, k+1, m\}$. The subroutine is based on the following equation:

$$L_u^+(k,k+1,m) = \begin{array}{cccccc} & \Omega(k+1,m-1)\times\{0\} & & \cdots & & \Omega(0,m-1)\times\{k+1\} \\ \Omega(k,m-1)\times\{0\} & L_u^+(k,k+1,m-1) & u_m I & & & \\ \Omega(k-1,m-1)\times\{1\} & & \ddots & u_m I & & \\ \vdots & & & \ddots & \ddots & \\ \Omega(1,m-1)\times\{k-1\} & & & \ddots & u_m I & \\ \Omega(0,m-1)\times\{k\} & & & & L_u^+(0,1,m-1) & u_m \end{array}, \tag{16}$$

where u_m is the probability that a Markov chain enters phase m from $\{m+1, \ldots, m_s\}$, and $L_u^+(0,1,m) = \mathbf{u}_{[1:m]}$, $L_u^+(k, k+1, 1) = u_1$. MATLAB-codes for PPlus( ) are given in Table A.5 in the online supplement.

Subroutine PMinus( ): For given $\{k, m, \mathbf{v}\}$, we find $L_v^-(k+1,\ k,\ m)$, which is based on the following:

$$L_v^-(k,k-1,m) = \begin{array}{cccccc} & \Omega(k-1,m-1)\times\{0\} & & \cdots & & \Omega(0,m-1)\times\{k-1\} \\ \Omega(k,m-1)\times\{0\} & L_v^-(k,k-1,m-1) & & & & \\ \Omega(k-1,m-1)\times\{1\} & v_m I & \ddots & & & \\ \vdots & & \ddots & \ddots & & \\ \vdots & & & \ddots & \ddots & \\ \Omega(1,m-1)\times\{k-1\} & & & & (k-1)v_m I & L_v^-(1,0,m-1) \\ \Omega(0,m-1)\times\{k\} & & & & & kv_m \end{array}, \tag{17}$$

where v_m is the probability that a Markov chain leaves phase m into $\{m+1, \ldots, m_s\}$, and $L_v^-(1,0,m) = \mathbf{v}_{[1:m]}$, $L_v^-(k+1, k, 1) = (k+1)v_1$. MATLAB-codes for PMinus( ) are given in Table A.6 in the online supplement.

Subroutine Pkm( ): For given $\{k, m\}$, we find $P(k, m)$ based on Equations () to (15) and (sub) P-matrix $S[1{:}m, 1{:}m]$. MATLAB-codes for Pkm( ) are given in Table A.7 in the online supplement. We note that $P(K, m_s)$ can be generated by calling Pkm( ) with $\{K, m_s, S\}$.

Verification. Similar to the continuous time case, the stationary distribution of $P(K, m_s)$ can be constructed from that of S by calling SPhi( ), which can be used to check the correctness of $P(K, m_s)$.

4.2. Discrete Time *MAP/PH/K* Queue

The queueing model has K identical servers and a single queue. The customer arrival process and service time are defined explicitly as follows.

- Arrivals: Discrete time Markovian arrival process (D_0, D_1), where D_0 and D_1 are square matrices of order m_a. The stationary distribution of the underlying Markov chain $\boldsymbol{\theta}_a$ satisfies $\boldsymbol{\theta}_a(D_0 + D_1) = \boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_a \mathbf{e} = 1$.
- Service: Discrete time phase-type distribution with PH-representation $(\boldsymbol{\beta}, S)$ of order m_s. The stationary distribution of the underlying Markov chain $\boldsymbol{\theta}_s$ satisfies $\boldsymbol{\theta}_s(S + \mathbf{S}^0\boldsymbol{\beta}) = \boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_s \mathbf{e} = 1$.

Define $q(t)$ be the queue length at time t, $I_a(t)$ the phase of the underlying Markov chain of the Markovian arrival process, $I_{s,k}(t)$ the phase of the underlying Markov chain associated with the service time of the k-th server (which is working), at time t. It is easy to see that $\{(q(t), I_a(t), I_{s,1}(t), \ldots, I_{s,\min\{q(t),K\}}(t)), t = 0, 1, 2, \ldots\}$ is a DTMC. It is also easy to see that $\{(q(t), I_a(t), N_1(t), \ldots, N_{m_s}(t)), t = 0, 1, 2, \ldots\}$ is a DTMC. For both the TPFS and CSFP approaches, the P-matrix of the associated DTMC is of the $GI/M/1$-type and can be written as follows:

$$P = \begin{pmatrix} A_{0,0} & A_{0,1} & & & & & & & & \\ A_{1,0} & A_{1,1} & A_{1,2} & & & & & & & \\ \vdots & \ddots & \ddots & \ddots & & & & & & \\ A_{K-1,0} & \ldots & A_{K-1,K-2} & A_{K-1,K-1} & A_{K-1,K} & & & & & \\ A_{K,0} & A_{K,1} & \ldots & A_{K,K-1} & A_{K,K} & A_0 & & & & \\ & A_{K+1,1} & A_{K+1,2} & \ddots & A_{K+1,K} & A_1 & A_0 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & A_{2K-1,K-1} & A_{2K-1,K} & \ddots & A_2 & A_1 & A_0 & \\ & & & & A_{2K,K} & A_K & \ldots & A_2 & A_1 & A_0 \\ & & & & & A_{K+1} & A_K & \ldots & A_2 & A_1 & A_0 \\ & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{18}$$

For the CSFP approach, the transition blocks can be obtained as follows:

(1) $A_{k,k+1} = D_1 \otimes P_{\beta,\mathbf{S}^0}(k, k+1, m_s \mid 0)$, for $k \le K-1$;

(2) $A_0 = A_{k,k+1} = D_1 \otimes P_{\beta,\mathbf{S}^0}(K, K, m_s \mid 0)$, for $k \ge K$;

(3) $A_{k,0} = D_0 \otimes P_{\beta,\mathbf{S}^0}(k, 0, m_s \mid k)$, for $k \le K$;

(4) $A_{k,k-K} = D_0 \otimes P_{\beta,\mathbf{S}^0}(K, k-K, m_s \mid K)$,
for $K+1 \le k \le 2K-1$;

(5) $A_{K+1} = A_{k,k-K} = D_0 \otimes P_{\beta,\mathbf{S}^0}(K, K, m_s \mid K)$, for $k \ge 2K$.

(6) $A_{k,j} = D_0 \otimes P_{\beta,\mathbf{S}^0}(k, j, m_s \mid k-j)$
$+ D_1 \otimes P_{\beta,\mathbf{S}^0}(k, j, m_s \mid k-j+1)$,
for $k \le K$, $1 \le j \le k$;

(7) $A_{k,j} = D_0 \otimes P_{\beta,\mathbf{S}^0}(K, \min\{j, K\}, m_s \mid k-j)$
$+ D_1 \otimes P_{\beta,\mathbf{S}^0}(K, \min\{j, K\}, m_s \mid k-j+1)$,
for $K+1 \le k \le 2K-1$, $k-K+1 \le j \le k$;

(8) $A_{k-j+1} = A_{k,j} = D_0 \otimes P_{\beta,\mathbf{S}^0}(K, K, m_s \mid k-j)$
$+ D_1 \otimes P_{\beta,\mathbf{S}^0}(K, K, m_s \mid k-j+1)$,
for $2K \le k$, $k-K+1 \le j \le k$, (19)

and $P_{\beta,\mathbf{S}^0}(k, j, m_s \mid l)$ is obtained as in Equation (14) by calling subroutine Pkm( ), PPlus( ), and PMinus( ). See Table A.8 in the online supplement for MATLAB-codes.

Proposition 4.1. *Let $A = A_0 + A_1 + \cdots + A_{K+1}$. Then $\boldsymbol{\theta}_a \otimes \boldsymbol{\phi}$ is the stationary distribution of A, where $\boldsymbol{\phi}$ can be obtained by calling SPhi( ) with $\{K, m_s, \boldsymbol{\theta}_s\}$.*

4.3. Superposition of Discrete Time Markovian Arrival Processes

We consider K independent identical discrete time *MAP*s $\{\{(W_k(t), I_k(t)), t > 0\}, k = 1, 2, \ldots, K\}$ with common matrix-representation (D_0, D_1) of order m_a. Define $W(t) = W_1(t) + \cdots + W_K(t)$. It is well-known that $\{(W(t), I_1(t), \ldots, I_K(t)), t > 0\}$ is a batch Markovian arrival process with matrix-representation $(C_0, C_1, \ldots, C_K)$ of order m_a^K, where $C_0 = C_{K,0}, C_1 = C_{K,1}, \ldots, C_K = C_{K,K}$, and

$$\begin{aligned}
&C_{0,0} = 1, \quad C_{k,0} = C_{k-1,0} \otimes D_0, \quad \text{for } k = 1, 2, \ldots, K;\\
&C_{k,j} = C_{k-1,j} \otimes D_0 + C_{k-1,j-1} \otimes D_1,\\
&\qquad\qquad\qquad \text{for } k = 1, 2, \ldots, K,\ j = 1, 2, \ldots, k-1;\\
&C_{k,k} = C_{k-1,k-1} \otimes D_1, \quad \text{for } k = 1, 2, \ldots, K. \qquad (20)
\end{aligned}$$

Following the CSFP approach, it is easy to see that $\{(W(t), N_1(t), \ldots, N_{m_s}(t)), t > 0\}$ is a *MAP* with matrix-representation $(C_0, C_1, \ldots, C_K)$ of order $(K + m_a - 1)!/(K!(m_a - 1)!)$. However, it is not straightforward to construct $\{C_0, C_1, \ldots, C_K\}$ from $\{D_0, D_1\}$ by using the subroutines developed in Section 4.1. The reason is that transitions of different (original) underlying Markov chains can come from D_0 or D_1, which is hard to distinguish. Let $P_0(k, m_a)$ be generated by calling Pkm( ) with $\{k, m_s = m_a, S = D_0\}$ and $P_1(k, m_a)$ be generated by calling Pkm( ) with $\{k, m_s = m_a, S = D_1\}$. Then $P_0(k, m_a)$ contains the transition probabilities from $\Omega(k, m_a)$ to $\Omega(k,\ m_a)$ without arrivals for k *MAP*s, and $P_1(k,\ m_a)$ contains the transition probabilities from $\Omega(k, m_a)$ to $\Omega(k, m_a)$ with k arrivals for k *MAP*s. It is then clear that $C_0 = P_0(K, m_a)$ and $C_K = P_1(K, m_a)$. It is also clear that $P_0(K-k, m_a) \otimes P_1(k, m_a)$ contains all the probabilities that there are exactly k arrivals from K *MAP*s, which is defined on state space $\Omega(K-k, m_a) \times \Omega(k, m_a)$. We call state $(\mathbf{x}_1, \mathbf{x}_2)$ in $\Omega(K-k, m_a) \times \Omega(k, m_a)$ *equivalent to* state $\mathbf{x}$ in $\Omega(K, m_a)$ if $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{x}$. It is readily seen that any state in $\Omega(K-k, m_a) \times \Omega(k, m_a)$ is equivalent (uniquely) to a state in $\Omega(K, m_a)$. Let Γ be the matrix for the mapping from $\Omega(K-k, m_a) \times \Omega(k, m_a)$ to $\Omega(K, m_a)$ between equivalent states, i.e., $\Gamma((\mathbf{x}_1, \mathbf{x}_2), \mathbf{x}) = 1$, if $(\mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{x}$ are equivalent; 0, otherwise. Then C_k can be obtained by normalizing each row of $\Gamma'(P_0(K-k, m_a) \otimes P_1(k, m_a))\Gamma$ to one, for $k = 0, 1, 2, \ldots, K$. Note that Γ' is the transpose of Γ. We would like to point out that the implementation of the scheme is significantly compromised by the order of matrix Γ, which can be too big for practical use.

For the *PH*-renewal process with phase-type renewal times with *PH*-representation $\{m_a, \boldsymbol{\alpha}, T\}$, for which $D_0 = T$, $D_1 = \mathbf{T}^0\boldsymbol{\alpha}$, and $\mathbf{T}^0 = (I - T)\mathbf{e}$, $\{C_0, C_1, \ldots, C_K\}$ can be constructed in a straightforward manner. In fact, we have $C_0 = P(K, m_a)$ generated using Equation () with $S = T$, and $C_k = P_{\alpha,\mathbf{T}^0}(K, K, m_a \mid k)$, for $k = 1, 2, \ldots, K$, which is defined in Equation (14) with $m = m_a + 1$ and $S[1{:}m-1, 1{:}m-1] = T$, can be obtained by calling subroutines Pkm( ), PPlus( ), and PMinus( ). The idea behind the method is that an external phase, i.e., phase $m_a + 1$, can be introduced for out-going transitions that correspond to all the arrivals.

4.4. Order Statistics of Discrete Phase-Type Random Variables

Consider K independent identical discrete phase-type random variables $\{X_1, \ldots, X_K\}$ with a common *PH*-representation $(m_s, \boldsymbol{\beta}, S)$ (see Neuts 1981) and $\mathbf{S}^0 = \mathbf{e} - S\mathbf{e}$. Denote by $\{X_{(1)}, \ldots, X_{(K)}\}$ the order statistics of $\{X_1, \ldots, X_K\}$ with $X_{(K)} = \min\{X_1, \ldots, X_K\}$ and $X_{(1)} = \max\{X_1, \ldots, X_K\}$. It is well-known that the order statistics are also discrete phase-type distributions.

Define $I_k(t)$ the phase of the underlying DTMC associated with X_k, $k = 1, 2, \ldots, K$. The state space of $I_k(t)$ is $\{1, \ldots, m_s, m_s + 1\}$. Similar to the continuous case, we define $q(t) = \sum_{k=1}^K 1_{\{I_k(t) < m_s + 1\}}$. Then $q(t)$ is decreasing. It is easy to see that $\{(q(t), I_1(t), \ldots, I_K(t)), t = 0, 1, \ldots\}$ is a DTMC (the TPFS approach) and the number of states of this Markov chain is $1 + m_s + m_s^2 + \cdots + m_s^K$. It is also readily seen that $\{(q(t), N_1(t), \ldots, N_{m_s}(t)), t \geq 0\}$ is a DTMC (the CSFP approach) with $\sum_{k=1}^K (k + m_s - 1)!/(k!(m_s - 1)!)$ states. Both processes are of the pure death-type with *P*-matrix:

$$P(K, m_s) = \begin{array}{c} 0 \\ 1 \\ \vdots \\ K-1 \\ K \end{array} \begin{pmatrix} P_{0,0} & & & & \\ P_{1,0} & P_{1,1} & & & \\ \vdots & \ddots & \ddots & & \\ P_{K-1,0} & \cdots & P_{K-1,K-2} & P_{K-1,K-1} & \\ P_{K,0} & \cdots & \cdots & P_{K,K-1} & P_{K,K} \end{pmatrix}. \qquad (21)$$

For the TPFS approach, we have

$$\begin{aligned}
&P_{0,0} = 1, \quad P_{k,0} = P_{k-1,0} \otimes \mathbf{S}^0, \quad \text{for } k = 1, 2, \ldots;\\
&P_{k,j} = P_{k-1,j} \otimes \mathbf{S}^0 + P_{k-1,j-1} \otimes S,\\
&\qquad\qquad\qquad \text{for } k = 1, 2, \ldots,\ j = 1, 2, \ldots, k-1;\\
&P_{k,k} = P_{k-1,k-1} \otimes S, \quad \text{for } k = 1, 2, \ldots. \qquad (22)
\end{aligned}$$

For the CSFP approach, $P_{k,k}$ can be obtained by calling Pkm( ) with $\{k, m_s, S\}$, and $P_{k,j}$ can be obtained as follows:

$$P_{k,j} = L^-_{\mathbf{S}^0}(k, j, m_s)P(j, m_s)/(k-j)!, \quad \text{for } k > j. \qquad (23)$$

which can be constructed by calling PMinus( ) and Pkm( ) with $\{k, m_s, S, \mathbf{S}^0\}$. See Table A.9 for MATLAB-codes.

The relationship between $q(t)$ and $\{X_{(1)}, \ldots, X_{(K)}\}$ is

$$X_{(k)} = \min\{t\colon q(t) \le k-1\}, \quad \text{for } k = 1, 2, \ldots, K. \qquad (24)$$

Based on the above relationship and the definition of phase-type distribution, the PH-representations of $\{X_{(1)}, \ldots, X_{(K)}\}$ can be obtained as

(c) $X_{(K)}$: $(\boldsymbol{\alpha}(K), T(K))$, where $\boldsymbol{\alpha}(K) = \mathrm{SPhi}(K, m_s, \boldsymbol{\beta})$ and $T(K) = P_{K,K}$;

(d) $X_{(k)}$: $(\boldsymbol{\alpha}(k), T(k))$, where $\boldsymbol{\alpha}(k) = (0, \boldsymbol{\alpha}(k+1))$ and

$$T(k) = \begin{pmatrix} P_{k,k} & 0 \\ \begin{pmatrix} P_{k+1,k} \\ \vdots \\ P_{K,k} \end{pmatrix} & T(k+1) \end{pmatrix}, \quad \text{for } k = K-1, K-2, \ldots, 1.$$

5. Conclusion

This paper demonstrates that by using the CSFP, many Markov chains that are otherwise deemed too huge to handle for practical purposes can now be considered feasible for real-life problems. The space complexity of Markov chains can be significantly reduced, but the construction of the Markov chains is quite involved. The subroutines provided in this paper now make it possible to construct and compute them efficiently, making the required tools very accessible.

References

Asmussen S, Möller JR (2001) Calculating of the steady state waiting time distribution in *GI/PH/c* and *MAP/PH/c* queues. *Queueing Systems* 37(1):9–29.

Asmussen S, O'Cinneide CA (1998) Representations for matrix-geometric and matrix-exponential steady-state distributions with applications to many-server queues. *Stochastic Models* 14(1–2):369–387.

Bladt M, Nielsen BF (2017) *Applied Probability with Matrix-Exponential Methods and Statistical Considerations* (Springer, New York).

Breuer L, Dudin A, Klimenok V (2002) A retrial *BMAP/PH/N* system. *Queueing Systems* 40(4):433–457.

He QM, Alfa AS (2015) Construction of Markov chains for discrete time *MAP/PH/K* queues. *Performance Evaluation* 93:17–26.

He QM, Zhang H, Ye QQ (2017) An *M/PH/K* queue with constant impatient time. *Math. Methods Oper. Res.* Forthcoming.

Kim CS, Mushko VV, Dudin AN (2012) Computing of the steady state distribution for multi-server retrial queues with phase type service process. *Ann. Oper. Res.* 201(1):307–323.

Neuts MF (1979) A versatile Markovian point process. *J. Appl. Probab.* 16(4):764–779.

Neuts MF (1981) *Matrix-Geometric Solution in Stochastic Model: An Algorithmic Application* (Johns Hopkins University Press, Baltimore).

Ramaswami V (1985) Independent Markov process in parallel. *Stochastic Models* 1(3):419–432.

Ramaswami V, Lucantoni DM (1985) Algorithms for the multi-server queue with phase type service. *Stochastic Models* 1(3):393–417.

Thompson C, Chandra K, Mulpur S, Davis J (2001) Modeling packet delay in multiplexed video traffic. *Telecomm. Systems* 16(3, 4):335–345.

Wagner D (1997) Analysis of mean values of a multi-server model with non-preemptive priorities and non-renewal input. *Stochastic Models* 13(1):67–84.