EFFICIENT MATRIX-ANALYTIC SOLUTION OF MULTI-TYPE QUEUEING SYSTEMS WITH CORRELATED TRAFFIC

GÁBOR HORVÁTH

Dissertation submitted to the Hungarian Academy of Sciences
for the degree of Doctor of Sciences

2017, Budapest

dc_1412_17

ABSTRACT

Erlang defined and solved the first queueing model 100 years ago to characterize the number of active calls in a telephone exchange. Since then, queueing theory has been an essential tool in the research of telecommunication systems.

The application of classical queueing models for the analysis of modern telecommunication networks is increasingly challenging: both the stochastic behavior of the traffic and the schedulers forwarding the packets in the network devices are becoming more and more complex. The so-called matrix-analytic methods allow to solve many of the corresponding complex queueing systems efficiently, with Markovian tools. This dissertation provides an overview on these modern techniques of queueing theory and presents several new results.

In the first part the main tools of the Markovian workload characterization, the phase-type distributions and the Markovian arrival processes are considered. For both traffic models, the role of the representations is discussed, special representations are introduced and new moment matching methods are developed. These results make it possible to create Markovian models for the network traffic, that can be used both in simulation based and in analytical performance analysis.

The second part of the dissertation presents the solution of single-class and multi-class queues with correlated arrival processes and Markovian service times. In the multi-class case both the first-come-first-served and the priority service policy are considered. Many performance measures are derived based on the analysis of the queue length process, the workload process and the age process. The characteristics of the traffic departing from these queues are also investigated.

In the third part a novel queueing network solution approach is described, that integrates the results of the first two parts. In this approach the traffic of the queueing network is characterized by Markovian arrival processes discussed in the first part, and the nodes of the network are the queues discussed in the second part of the dissertation. The Markovian arrival processes representing the internal traffic are obtained by moment matching. An extensive numerical study investigates the behavior of the presented approach and compares it with other existing solutions from the literature.

dc_1412_17

ACKNOWLEDGMENTS

There are many people to whom I am grateful for their contribution to the results presented in this dissertation, it is impossible to list everyone. Let me mention only a couple of them. At the first place I would like to thank my family for their patience. I have to thank the Department of Networked Systems and Services for providing the flexibility and the intellectual freedom in my work. Last but not least, to Miklós Telek for the pleasant common work and for the many valuable advice and encouragement in both personal and scientific questions.

dc_1412_17

CONTENTS

ACRONYMS

CTMC	continuous time Markov chain
DTMC	discrete time Markov chain
PH	phase-type
ME	matrix-exponential
FEB	feedback Erlang block
MAP	Markovian arrival process
MMAP	marked Markovian arrival process
SMMAP	structured marked Markovian arrival process
SMAP	structured Markovian arrival process
RAP	rational arrival process
MRAP	marked rational arrival process
QBD	quasi birth-death process
MFM	Markovian fluid model
LST	Laplace-Stieltjes transform
GF	generating function
SCV	squared coefficient of variation
NARE	non-symmetric algebraic Riccati equation
FCFS	first-come-first-served
pdf	probability density function
cdf	cumulative distribution function
iid.	independent and identically distributed

NOTATIONS

$\boldsymbol{I}$	Identity matrix of appropriate size
$\mathbb{1}$	Column vector of ones of appropriate size
$\mathcal{L}(t)$	The level process of QBDs and Markovian fluid models at time t
$\mathcal{J}(t)$	The phase process at time t
$\mathcal{C}(t)$	The type of the customer in the server at time t
N	The number of phases
$\underline{\pi}$	The (row) vector of stationary probabilities
$\underline{\pi}(x)$	The (row) vector of stationary densities
$\mathcal{X}$	The stationary number of customers in the system at departure instants
$\mathcal{Y}$	The stationary number of customers in the system at random instants
$\mathcal{A}(t)$	The age process
$\mathcal{V}(t)$	The workload process
$\mathcal{B}$	The busy period
$\mathcal{T}$	The stationary sojourn time of customers
$\mathcal{W}$	The stationary waiting time of customers
$\boldsymbol{D_0}, \boldsymbol{D_1}$	The matrices characterizing the MAP of the arrivals
$\mathcal{S}, (\mathcal{S}_k)$	The random variable representing the (type k) service times
$\underline{\sigma}, \boldsymbol{S}$	The parameters of the PH distributed service times
$\boldsymbol{S_0}, \boldsymbol{S_1}$	The matrices characterizing the MAP of the service times
$\mathcal{H}$	The random variable representing the inter-departure times
$\boldsymbol{H_0}, \boldsymbol{H_1}$	The matrices characterizing the MAP of the departures
$\boldsymbol{Q}$	The generator matrix of a continuous time Markov chain
$\underline{\theta}$	The stationary phase distribution of the background process

dc_1412_17

1

INTRODUCTION

1.1 MOTIVATION

1.1.1 *Queueing theory for analyzing computer and communication systems*

The application of queueing theory in the field of telecommunication has a long history.

At the beginning of the last century, the wide spread deployment of plain old telephone systems established the need for efficient dimensioning methods. The goal was to determine the number of trunks required to meat a certain level of service quality. Erlang was the first to model the number of active calls with a queueing system. He proved that the telephone traffic can be modeled by a Poisson process and created the classic formulas for call loss and waiting time. This was the first engineering problem that was solved by queueing theory.

Over time, newer and newer telecommunication technologies have been developed. With the appearance of data traffic the packet switched communication method started to gain popularity, packet based technologies like the ATM (Asynchronous Transfer Mode) and the IP (Internet Protocol) have slowly superseded the circuit switched technology. The packet switched networking gave an other boost to the application of queueing models in telecommunication. Since the quality of service (QoS) measures like packet loss and delay are associated with the buffers in the network nodes (where packets are residing temporarily before getting forwarded to the next node), open queueing networks appeared to be the ideal modeling tools for performance analysis [32, 15].

Compared to the detailed simulation of the system, the analytical solution of queueing models usually need less input data (hence easier to use) and is usually very fast. Consequently, it enables the fast evaluation of different possibilities for providing service, performing parameter sensitivity analysis, or just to gain insight into the behavior of the system.

Modern telecommunication systems, however, have some unique properties that make the application of queueing models increasingly challenging. Some of the reasons are listed below.

- The characteristic of the traffic has changed considerably in the recent decades. While the Poisson process is still a reasonable model for voice call arrivals, the data traffic behaves differently. Statistical properties like long-term correlations, self-similar behavior and heavy tailed distributions make the direct application of the classical queueing models difficult [38].

- The network traffic is grouped into various classes according to the quality demands. While there are some basic queueing models available that support such traffic differentiation, more general queueing models are substantially more difficult to solve in the multi-class case.

- Communication technologies are becoming very complex: the underlying protocols have multiple interacting layers, and use several (possibly interfering) flow control algorithms. It is becoming increasingly difficult to identify the components of the system that have the dominant impact on the overall performance. Analytical models for such complex systems have to introduce many assumptions and simplifications.

Looking for answers for these challenges has been the main driver of queueing theory research in the recent years.

1.1.2 *Further application fields of queueing theory*

Although the modeling of telecommunication systems is still the most popular application area of queueing theory, there are several other fields where queueing models are used frequently.

One of these emerging application fields is *healthcare* ([31]), where queueing systems are applied to optimize the cost by minimizing the inefficiencies and delays. In a healthcare process the demands waiting in the queue are the patients, and the servers are the doctors or any other specialized equipment (like an operating room). Compared to telecommunication systems, queues in healthcare have some specialties, including that

- patients are impatient,
- the arrival rate is varying and depends on the state of the system (patients do not join the waiting line when it is too long),
- the service discipline is more complex.

Queueing models can be used for health care system design (to determine the optimal number of beds, ambulance cars, etc.), for health care operation (for the optimal scheduling of resources, patients, appointments, bed and staff management, etc.) and also in health care system analysis (to calculate the utilization, cost, delays, etc.) [55].

Queueing network models are used in the design and analysis of *manufacturing systems* since the 1950s as well ([34]). In a manufacturing system there are resources (machines), which perform various processing steps on products. From queueing point of view the customers are the parts, which, after arriving into the queue, have to wait till the suitable machines (that are the servers in queueing terminology) become available. Both the arrival process and the processing times are stochastic. A manufacturing system typically consists of many processing stations with their own buffers for the parts, and the parts have to go through many processing stations to become a final product. Hence, a queueing network model is a natural choice both for performance evaluation and optimization purposes, thus to analyze or optimize the material flow in the system, the utilization of the machines, the capacity of the buffers, etc.

Apart of these two application fields, queueing models have been successfully applied in many other areas as well, including vehicular traffic analysis [4], inventory systems [35], housing [52], banking [29], or even to optimize crowd sourcing systems [24].

1.2 MARKOVIAN PERFORMANCE ANALYSIS

The solution of many tractable queueing models is based on the analysis of a closely related Markov chain.

The queue length of basic queues having memoryless inter-arrival and service times (including the M/M/1, M/M/m, M/M/m/m queues [53], etc.) can be represented by a Markov chain directly, that, due to the regular tri-diagonal structure of the generator has a simple and explicit stationary solution, even for infinite systems.

The analysis of more complex queueing systems, were either the arrival or the service times are generally distributed, is based on the solution of an appropriately defined Markov chain as well. In case of G/M/1 (M/G/1) systems the Markov chain characterizing the queue length at specific embedded time instants has an upper Hessenberg (lower Hessenberg) structure, respectively. The regular structure enables the efficient solution of these systems.

Starting from the eighties, Markovian performance modeling has undergone an enormous development. The introduction of phase-type distributions and Markovian arrival processes made it possible to characterize a reasonably general class of arrival and service processes in a Markovian way. The generators of the Markov chains describing the related queues also have a regular structure, at block level. An elegant, numerically efficient solution methodology has been developed to solve such block-structured Markov chains, called *matrix-analytic method* ([65, 57]), which became one of the cornerstones of modern queueing theory.

The results derived in this dissertation rely on matrix-analytic methods heavily.

1.2.1 *Workload models*

To obtain relevant, meaningful results from a queueing model the workload must be characterized as accurately as possible. The workload characterization has two ingredients: the characterization of the arrival process of the demands and the characterization of the work brought into the system by a single demand.

Workload modes have to provide an appropriate balance between accuracy and tractability. A very accurate workload model can easily be useless if it can not be incorporated into an analytical or into a simulation model. The simplest workload models consisting of exponential distributions are easy to apply both in analytical and simulation models, but they might not capture the real behavior accurate enough making the result of the performance evaluation irrelevant [69].

Phase-type distributions and Markovian arrival processes provide a reasonable compromise between accuracy and tractability.

The convolution and the probabilistic (Bernoulli-) mixture of several exponential distributions, namely the Erlang and the hyper-exponential distributions, have been used for a long time to represent non-exponential behavior. The phase-type (PH) distributions ([65]), associated with the absorption time of transient Markov chains, are the generalizations of this concept. PH distributions have some very appealing properties, as listed below.

- The expressions providing the properties of PH distributions like the density function, moments, etc. are similar to those of the exponential distributions. PH distributions are the matrix-based counterparts of exponential distributions.
- They are proven to be dense, which means that any distribution can be approximated with a sufficiently large PH distribution.
- Several specific, widely used distributions including the Erlang-, hyper-exponential- and exponential distributions are the sub-classes of PH distributions.
- The sum, minimum, maximum and the mixture of PH distributed random variables are PH distributed as well.

- It is easy to replace exponentially distributed state transitions in a Markov model by PH distributed ones.
- PH distributed random variates can be generated efficiently in discrete event simulations.

Markovian arrival processes (MAPs) can characterize correlated point processes like inter-arrival times or a sequence of correlated service times. Similar to PH distributions, they are composed by exponential phases; they can be interpreted as Markov chains in which (arrival) events are generated when the Markov chain traverses some marked transitions.

The application of MAPs has many benefits:

- The differential equations characterizing the number of events generated by a MAP up to a given point of time are similar to those of a Poisson process, but are defined with matrices instead of scalars.
- MAPs are proven to be dense, hence, with the necessary number of states MAPs can approximate any point processes arbitrary close.
- MAPs include the Poisson process as a special case.
- The aggregation (superposition) and probabilistic splitting of MAPs remain MAPs as well.
- Queueing models involving Poisson processes can usually be extended to the more general MAPs easily.
- MAPs are easy to incorporate into simulation models.

The applicability of PH distributions and MAPs relies on the availability of effective fitting methods which obtain these models based on the real, empirical behavior.

Many PH fitting methods have been published in the literature. Some of them perform an optimization on the underlying Markov chain, while others aim to capture some statistical parameters exactly and compute the parameters of the PH directly by solving a system of (typically not linear) equations. Methods falling into the first category are the expectation-maximization based methods ([6]) and other optimization based methods like [46]. The second category is often referred to as *matching*. Examples for methods belonging to this group are the moment matching methods ([13]), and the Feldmann-Whitt algorithm aiming to match the density function at certain points [30].

There are much fewer results available on fitting MAPs since it is a more complex task. The first ones were the expectation-maximization based methods ([16]), but due to the computational complexity they were applicable only on small measurement traces. A number of MAP fitting methods were published when the"two-step" approach appeared ([50]), that suggested to split the fitting task to two steps: fitting of the inter-arrival times in the first step by a PH fitting method and fitting the correlations in the second step. However, it is still an open question what are the statistics that capture the correlation structure of the traffic the best. Recent results on the characterization of MAPs [81] revealed the importance of joint moments of two consecutive inter-arrival times, and that these joint moments can be better suited to describe the correlations of MAPs than the auto-correlation function used traditionally for this purpose.

According to our numerical experience, the moment and joint-moment based fitting methods for PH distributions and MAPs perform well in the practice. The moment based representation is compact, a few moments represent the distribution or the process relatively well. Additionally, there are performance measures of some queueing systems that are insensitive to moments higher than a given order (like the mean waiting time of M/G/1 queue, which depends only on the first two moments of the service time distribution). Hence, in this dissertation we are going to focus on moment based fitting methods: new fitting methods will be developed of such kind, and these methods will be used every time a PH distribution or a MAP needs to be created for a queueing system.

1.2.2 *Queues*

The purpose of queueing analysis is to obtain various performance measures like

- properties of the number of customers in the system or the queue length,
- properties of the sojourn time or waiting time of customers,
- the utilization of the system,
- the properties of the departure process,
- etc.,

given the arrival process, the service process and the service discipline.

There are three frequently used approaches to analyze queues:

- based on the queue length process,
- based on the workload process,
- and based on the age process.

The *queue length process* based approach is perhaps the most well-know method, upon which most classical textbooks are building. According to the queue length based approach a Markov chain is constructed to keep track of the queue length either at arbitrary time or at some embedded time instants. The queue length related performance measures are easy to derive from such a model. The sojourn times are usually calculated based on the law of total probability, by characterizing the time to leave the system conditioning on the queue length at customer arrivals.

In the *workload process* based approach the first step of the solution is the stationary analysis of the workload of the system. The workload (or backlog) of the system increases at arrival instants by the amount of work brought into the system, and decreases at a slope of one between arrivals expressing that the server is processing the backlog (see Figure 27). The sojourn time and waiting time related performance measures are given by the workload at arrival instants. The queue length properties, however, are a bit more challenging to derive by this method.

The *age process* based approach derives all performance measures from the age process, which represents the age of the oldest customer (the total time spent) in the system (Figure 25). It increases by a slope of one, and decreases at customer departures, when the next (younger) customer becomes the oldest one. The sojourn time of a customer is its age right

before the departure, and the queue length is the number of arrivals during the age of the oldest customer.

The queue length process based analysis of queues like the MAP/PH/1, MAP/G/1, G/MAP/1, etc. queues leads to Markov chains with a regular block structure, that can be solved efficiently by matrix-analytic methods since the late 1980s. The sojourn time properties of these queues are much easier to characterize based on the workload or the age process, that, as opposed to the queue length process, are continuous state processes, for which the solution methodology appeared only later. Some important results were published in [75] and [78], but the numerically efficient (matrix-analytic) solution became possible only by the combination of [71] and [28], since year 2005. Furthermore, it has been recognized that the workload and age process based approaches are the only reasonable ways to analyze multi-type queues like the MMAP[K]/PH[K]/1 and the MAP/G/1-Priority queues [41, 78].

Consequently, the matrix-analytic solution of the workload and age processes, and its application to the analysis of multi-class queues is a recent, elegant, and very effective analysis technology in modern queueing theory.

In this dissertation all three solution approaches are applied for different purposes.

1.2.3 *Queueing networks*

Open queueing networks are popular modeling tools for the performance analysis of computer and telecommunication systems. Exact solution methods are available only for networks with Poisson traffic input, specific service time distributions and service disciplines. These restrictive assumptions make the exact solutions unlikely to use in the practice. The main reason is that in real systems the Poisson process is usually not a good model for the traffic behavior. Instead, the real traffic can be bursty and correlated, and the service times in the service stations can be correlated as well. Since these features have an impact on the performance measures, they have to be taken into consideration.

The attempts to analyze queueing networks with non-Poisson traffic and non-exponential service time distributions dates back to the second half of the last century. The first attempts were to consider the second moments of the inter-arrival and the service time distributions in the computations. A widely applied approximation of this kind was integrated into the QNA tool [87, 88]. The intrinsic assumption in these approximations is that the consecutive inter-arrival times and the consecutive service times are independent. The evolution of packet switched communication networks during the eighties and nineties resulted in traffic with significant correlation which lead to the development of new modeling paradigms.

Several modeling approaches were developed to describe the properties of packet traffic better [73]. One of the lines of research is based on Markovian models with the aim of extending the Poisson arrival process in order to capture more statistical properties of the traffic behavior. A long series of efforts resulted in the application of MAPs. The main advantage of using MAPs for traffic description of queues is that they are closed for the basic traffic operations like superposition and splitting, and that the queueing models driven by MAPs can be solved in a numerically efficient way by matrix-analytic methods. Using MAPs for the traffic description gave a new impulse to the research on queueing network analysis [74, 42].

In this dissertation we present a new method along this line of research which is based on a recent result about the joint moment based representation of MAPs.

1.3 THE STRUCTURE OF THE DISSERTATION

The topic of Part I of the dissertation is workload modeling.

Chapter 2 introduces the PH distribution and its main properties. PH distributions will be used to represent the service times in the queueing models defined throughout the dissertation. It is demonstrated that a PH distribution can have many representations. Various tools and theorems are provided to perform representation transformations. After these preliminaries, the canonical representations are introduced, which, apart of their theoretical importance, are shown to be beneficial for PH fitting as well. The chapter ends with a moment matching method based on a special, so-called generalized hyper-Erlang representation, that will be used many times in the subsequent chapters.

The MAPs, and their multi-class extensions, the marked Markovian arrival processes (MMAPs) are discussed in details in Chapter 3. After introducing the representation transformations and the significance of Markovian representations, the main results are presented, namely the lag-1 joint moment based representation of MAPs, and the corresponding moment matching method. The results in this chapter can be applied to create MAPs based on empirical measurement data. Furthermore, the joint moments play a principal role in the novel queueing network analysis approach introduced in the last chapter.

The PH distributions and MMAPs obtained by the presented procedures can be applied to represent packet service times and packet inter-arrival times for both analytical and simulation based performance evaluation of telecommunication systems.

The performance analysis of various queueing models are covered in Part II of the dissertation.

Chapter 4 lays down the theoretical background by providing an overview on discrete and continuous state-space skip-free processes.

Chapter 5 is still an introductory chapter, that demonstrates how to use the queue length based, the workload process based and the age process based analysis techniques to obtain the performance measures of the well known MAP/MAP/1 queue. At the end of the chapter, the lag-1 joint moments of the departure process are also derived that will be essential for the queueing network analysis.

The multi-type first-come-first-served (FCFS) MMAP[K]/PH[K]/1 queue is considered in Chapter 6. What makes this system interesting is that it can not be solved by the classical queue length based approach. All performance measures, including the ones related to the departure process, are obtained from the age process.

In Chapter 7 an other multi-type system, with preemptive and non-preemptive priority queue is investigated. This system has been analyzed several times in the past, and the solution is proven to be challenging. With some transformations of the workload process of the system, however, it becomes possible to derive all performance measures efficiently.

The results of all the above mentioned chapters are integrated in Part III, where a novel queueing network solution approach is described. In the proposed method the traffic of the internal links of the network are characterized by MAPs, that are created from the lag-1 joint moments of the departure processes of the associated queues. The presented numerical examples demonstrate that this approach has several advantages: the MAPs of the internal traffic are compact, there are no scalability problems, and the performance measures approximate the exact results with a reasonable accuracy.

dc_1412_17

Part I

WORKLOAD MODELING

dc_1412_17

2

PHASE-TYPE DISTRIBUTIONS

Phase-type distributions are non-negative distributions with a Markovian structure [65, 57]. Due to their computational advantages and easy integration in complex stochastic models, they are widely used for modeling the workload in telecommunication systems.

2.1 INTRODUCTION TO PHASE-TYPE DISTRIBUTIONS

2.1.1 *Matrix-exponential distributions*

Before providing the definition of PH distributions we first define the more general matrix-exponential (ME) distributions. Historically, PH distributions were introduced before ME distributions, but to show their (non-trivial) relation it is better to discuss them in the reverse order.

Definition 1. *A row vector* $\underline{\sigma} = \{\sigma_i, i = 1, \ldots, N\}$, $\underline{\sigma}\mathbb{1} = 1$ *and square matrix* $\boldsymbol{S} = \{q_{ij}, i, j = 1, \ldots, N\}$ *define a ME distribution if the probability density function (pdf) given by*

$$f(x) = \underline{\sigma} e^{\boldsymbol{S}x}(-\boldsymbol{S})\mathbb{1} \tag{1}$$

is non-negative for $x \geq 0$.

The vector-matrix pair $(\underline{\sigma}, \boldsymbol{S})$ is called the *representation* of the ME distribution, and N is the size of the representation. From (1) it follows that the cumulative distribution function (cdf) denoted by $F(x)$, the Laplace-Stieltjes transform (LST) of the cdf $f^*(s)$ and the kth moment of a ME($\underline{\sigma}$,$\boldsymbol{S}$) distributed random variable $\mathcal{F}$ are

$$F(x) = P(\mathcal{F} < x) = 1 - \underline{\sigma} e^{\boldsymbol{S}x}\mathbb{1}, \tag{2}$$

$$f^*(s) = E(e^{-s\mathcal{F}}) = \underline{\sigma}(s\boldsymbol{I} - \boldsymbol{S})^{-1}(-\boldsymbol{S})\mathbb{1}, \tag{3}$$

$$m_k = E(\mathcal{F}^k) = \int_0^\infty x^k f(x)dx = k!\underline{\sigma}(-\boldsymbol{S})^{-k}\mathbb{1}, \tag{4}$$

where $\mathbb{1}$ is a column vector of ones and $\boldsymbol{I}$ is the identity matrix of appropriate size.

Definition 2. *A* $(\underline{\sigma}, \boldsymbol{S})$ *representation is called a* Markovian representation *if*

- *the entries of vector* $\underline{\sigma}$ *are valid probabilities* ($0 \leq \sigma_i \leq 1, i = 1, \ldots, N$),
- *matrix* $\boldsymbol{S}$ *is a valid generator of a transient continuous time Markov chain (CTMC), thus,* $q_{ii} < 0$ *and* $q_{ij} \geq 0, \forall i \neq j$,
- *and for the row sum we have that* $\sum_{j=1}^{N} q_{ij} \leq 0$, $i = 1, \ldots, N$, *with at least one state where the row sum is strictly negative.*

Otherwise the representation is called non-Markovian.

In the example below, $(\underline{\sigma}, \boldsymbol{S})$ is a Markovian, while $(\underline{\gamma}, \boldsymbol{G})$ is a non-Markovian representation of a ME distribution:

$$\underline{\sigma} = \begin{bmatrix} 0.5 & 0.4 & 0.1 \end{bmatrix}, \qquad \underline{\gamma} = \begin{bmatrix} 0.3 & -0.5 & 1.2 \end{bmatrix},$$
$$\boldsymbol{S} = \begin{bmatrix} -8 & 1 & 0 \\ 2 & -6 & 4 \\ 0 & 2 & -4 \end{bmatrix}, \qquad \boldsymbol{G} = \begin{bmatrix} -7 & 2 & 1 \\ 5 & -10 & 10 \\ 3 & -2 & -1 \end{bmatrix}. \tag{5}$$

The pdf $f(x)$ can be expressed in a spectral form, too. Suppose the number of distinct eigenvalues of $\boldsymbol{S}$ is n_d. Let us denote the eigenvalues by $-\lambda_i$, and their multiplicity by r_i ($\sum_{i=1}^{n_d} r_i = K \leq N$). From (1) we have

$$f(x) = \sum_{i=1}^{n_d} \sum_{j=1}^{r_i} b_{ij} \frac{(\lambda_i x)^{j-1}}{(j-1)!} \lambda_i e^{-\lambda_i x}. \tag{6}$$

Note that if $\lambda_i \in \mathbb{C} \backslash \mathbb{R}$ then $\exists j \neq i : \lambda_j = \bar{\lambda}_i$ ($\bar{\lambda}_i$ denotes the complex conjugate of λ_i). To define a valid distribution the integral of the density function must exist, implying that the real part of all eigenvalues must be strictly negative, hence $\mathrm{re}\langle \lambda_i \rangle > 0, i = 1, \ldots, n_d$ must hold. Furthermore, as a consequence of the Perron-Frobenius theorem the dominant eigenvalue (i.e., the eigenvalue with the largest real part) must be real.

When $K = N$, the $(\underline{\sigma}, \boldsymbol{S})$ representation is called *minimal*. If $K < N$ then matrix $\boldsymbol{S}$ has at least one eigenvalue that does not play a role in the pdf, because the corresponding coefficient is zero.

2.1.2 *Representation transformations*

It is easy to see that the representation of a ME distribution given by $(\underline{\sigma}, \boldsymbol{S})$ is not unique [26, 66]. For instance, applying a permutation on the elements of $\underline{\sigma}$ and $\boldsymbol{S}$ leads to a different representation while the distribution remains obviously the same. However, there are many more possibilities to create different representations of the same ME distribution. With *any* non-singular square matrix $\boldsymbol{B}$ satisfying $\boldsymbol{B}\mathbb{1} = \mathbb{1}$ the cdf given by (1) can be transformed as

$$\begin{aligned} F(x) &= 1 - \underline{\sigma} e^{\boldsymbol{S}x} \mathbb{1} = 1 - \underline{\sigma} \boldsymbol{B} e^{\boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}x} \boldsymbol{B}^{-1} \mathbb{1} \\ &= 1 - \underline{\gamma} e^{\boldsymbol{G}x} \mathbb{1}, \end{aligned} \tag{7}$$

with $\underline{\gamma} = \underline{\sigma}\boldsymbol{B}$ and $\boldsymbol{G} = \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$. Hence, representations $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\gamma}, \boldsymbol{G})$ are different, but define the exactly same cdf, thus the exactly same ME distribution. For example, the two representations $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\gamma}, \boldsymbol{G})$, as defined by (5), correspond to the same distribution, and the transformation matrix relating them is

$$\boldsymbol{B} = \begin{bmatrix} 0.5 & -1 & 1.5 \\ 0 & 0 & 1 \\ 0.5 & 0 & 0.5 \end{bmatrix}. \tag{8}$$

The definitions, results and properties regarding the representation transformation are elaborated in the following theorems.

Theorem 1. *[81] Let $(\underline{\sigma}, \boldsymbol{S})$ of size N and $(\underline{\gamma}, \boldsymbol{G})$ of size N represent two ME distributions with cdf $F_{\mathcal{F}}(x)$ and $F_{\mathcal{G}}(x)$, respectively. The two distributions are identical if there exists a non-singular matrix $\boldsymbol{B}$ of size $N \times N$, such that $\underline{\gamma} = \underline{\sigma}\boldsymbol{B}$, $\boldsymbol{G} = \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$ and $\boldsymbol{B}\mathbb{1} = \mathbb{1}$.*

Proof. See (7) for the derivation. □

Theorem 1 uses the square matrix $\boldsymbol{B}$ to transform between representations of the same size. This operation is called a *similarity transformation* in the sequel.

Definition 3. *The* similarity transform *of $(\underline{\sigma}, \boldsymbol{S})$ with matrix $\boldsymbol{B}$ is $(\underline{\sigma}\boldsymbol{B}, \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B})$ if $\boldsymbol{B}$ is non-singular and $\boldsymbol{B}\mathbb{1} = \mathbb{1}$.*

The main properties of a similarity transformation are as follows (cf. [11])

- $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\sigma}\boldsymbol{B}, \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B})$ have the same size,
- the eigenvalues of $\boldsymbol{S}$ and $\boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$ are identical,
- if $(\underline{\sigma}, \boldsymbol{S})$ is Markovian then $(\underline{\sigma}\boldsymbol{B}, \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B})$ can be both Markovian and non-Markovian.

Representations with different sizes can be transformed into each other in a similar manner, using a non-square transformation matrix. This is stated by the following two theorems, which are symmetric to each other.

Theorem 2. *[66, 21] Let $(\underline{\sigma}, \boldsymbol{S})$ of size N and $(\underline{\gamma}, \boldsymbol{G})$ of size M $(M > N)$ be two ME distributions with cdf $F_{\mathcal{F}}(x)$ and $F_{\mathcal{G}}(x)$, respectively. If there exists a matrix $\boldsymbol{V}$ of size $M \times N$, such that $\underline{\sigma} = \underline{\gamma}\boldsymbol{V}$, $\boldsymbol{V}\boldsymbol{S} = \boldsymbol{G}\boldsymbol{V}$, $\boldsymbol{V}\mathbb{1} = \mathbb{1}$ then the distributions given by $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\gamma}, \boldsymbol{G})$ are identical.*

Proof. If $\underline{\sigma} = \underline{\gamma}\boldsymbol{V}$, $\boldsymbol{V}\boldsymbol{S} = \boldsymbol{G}\boldsymbol{V}$, $\boldsymbol{V}\mathbb{1} = \mathbb{1}$ then

$$\begin{aligned} F_{\mathcal{F}}(x) &= 1 - \underline{\sigma}e^{\boldsymbol{S}x}\mathbb{1} = 1 - \underline{\sigma}\sum_{i=0}^{\infty}\boldsymbol{S}^i\frac{x^i}{i!}\mathbb{1} = 1 - \underline{\gamma}\boldsymbol{V}\sum_{i=0}^{\infty}\boldsymbol{S}^i\frac{x^i}{i!}\mathbb{1} \\ &= 1 - \underline{\gamma}\sum_{i=0}^{\infty}\boldsymbol{G}^i\frac{x^i}{i!}\boldsymbol{V}\mathbb{1} = 1 - \underline{\gamma}\sum_{i=0}^{\infty}\boldsymbol{G}^i\frac{x^i}{i!}\mathbb{1} = 1 - \underline{\gamma}e^{\boldsymbol{G}x}\mathbb{1} = F_{\mathcal{G}}(x). \end{aligned} \tag{9}$$

□

Theorem 3. *[21] Let $(\underline{\sigma}, \boldsymbol{S})$ of size N and $(\underline{\gamma}, \boldsymbol{G})$ of size M $(M > N)$ be two ME distributions with cdf $F_{\mathcal{F}}(x)$ and $F_{\mathcal{G}}(x)$, respectively. If there exists a matrix $\boldsymbol{W}$ of size $N \times M$, such that $\underline{\sigma}\boldsymbol{W} = \underline{\gamma}$, $\boldsymbol{S}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{G}$, $\boldsymbol{W}\mathbb{1} = \mathbb{1}$ then the distributions given by $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\gamma}, \boldsymbol{G})$ are identical.*

Proof. If $\underline{\sigma}\boldsymbol{W} = \underline{\gamma}$, $\boldsymbol{S}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{G}$, $\boldsymbol{W}\mathbb{1} = \mathbb{1}$ then

$$\begin{aligned} F_{\mathcal{F}}(x) &= 1 - \underline{\sigma}e^{\boldsymbol{S}x}\mathbb{1} = 1 - \underline{\sigma}\sum_{i=0}^{\infty}\boldsymbol{S}^i\frac{x^i}{i!}\mathbb{1} = 1 - \underline{\sigma}\sum_{i=0}^{\infty}\boldsymbol{S}^i\frac{x^i}{i!}\boldsymbol{W}\mathbb{1} \\ &= 1 - \underline{\sigma}\boldsymbol{W}\sum_{i=0}^{\infty}\boldsymbol{G}^i\frac{x^i}{i!}\mathbb{1} = 1 - \underline{\gamma}\sum_{i=0}^{\infty}\boldsymbol{G}^i\frac{x^i}{i!}\mathbb{1} = 1 - \underline{\gamma}e^{\boldsymbol{G}x}\mathbb{1} = F_{\mathcal{G}}(x). \end{aligned} \tag{10}$$

□

For equivalent representations with different sizes we have the following properties.

- The eigenvalues of $\boldsymbol{S}$ are all eigenvalues of $\boldsymbol{G}$ with at least the same multiplicity.
- If $(\underline{\sigma}, \boldsymbol{S})$ is Markovian then $(\underline{\gamma}, \boldsymbol{G})$ can be either Markovian or non-Markovian.

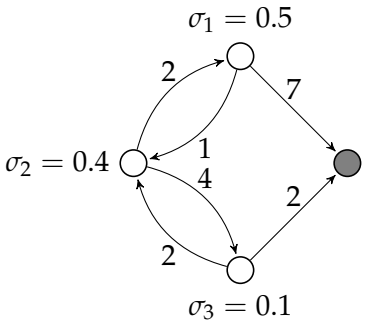


Figure 1.: The transient Markov chain belonging to (5)

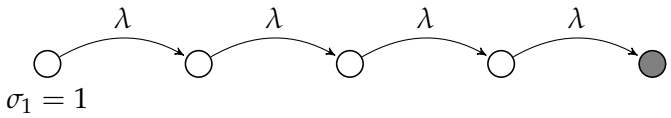


Figure 2.: An Erlang distribution

2.1.3 *Phase-type distributions*

PH distributions of size N are the subclass of ME distributions of the same size having a Markovian representation.

If $(\underline{\sigma}, \boldsymbol{S})$ is a Markovian representation of a ME distributed random variable $\mathcal{F}$, then the corresponding PH distribution has a probabilistic interpretation: $\mathcal{F}$ is the time to absorption of a transient Markov chain with sub-generator $\boldsymbol{S}$ and initial state probability vector $\underline{\sigma}$ for the non-absorbing states. The transient Markov chain and the initial state probabilities belonging to $(\underline{\sigma}, \boldsymbol{S})$ defined in (5) are depicted in Figure 1.

There exist order N non-Markovian $(\underline{\sigma}, \boldsymbol{S})$ representations that define a valid (non-negative) density function but can not be transformed to an order N Markovian representation. These distributions do not belong to the PH class, while they are still ME distributions. Consequently, the set of ME distributions is a superset of the one of PH distributions, thus we have $PH(N) \subset ME(N)$.

The significance of PH distributions in Markovian performance modeling is a consequence of two properties. First, it is proven in [67] that PH distributions form a dense subset of the set of all positive-valued distributions, which means that any distribution having a positive density in $(0, \infty)$ can be approximated arbitrarily well by a PH distribution. The second benefit of using PH distributions is their closeness on many operations: the sum, the minimum and the maximum of independent PH distributed random variables are PH distributed as well.

2.1.4 *Important representations*

There are some particular, frequently used representations of PH distributions.

The most well-known PH sub-classes are the Erlang (Figure 2), the hyper-exponential and the hyper-Erlang (Figure 3) distributions. These distributions have especially simple pdf and moment formula, enabling efficient simulation, moment matching, fitting, and analytical studies.

There are two further representations playing an essential role in the theory of PH distributions: the acyclic and the monocyclic representations.

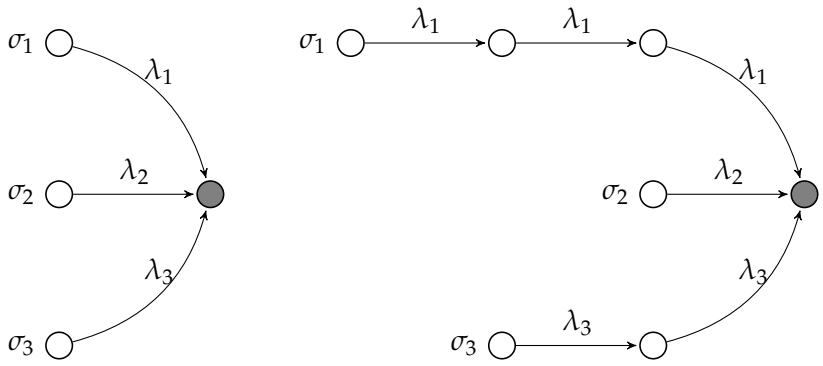


Figure 3.: Hyper-exponential and hyper-Erlang distributions

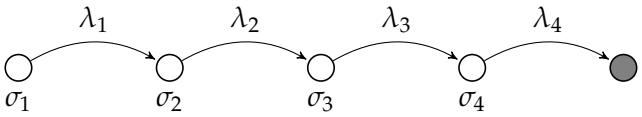


Figure 4.: One of the canonical forms of acyclic PH distributions

ACYCLIC PH DISTRIBUTIONS have either upper or lower triangular generators, thus their state transition graph does not have any loops. Their distinguishing feature is that they have *minimal*, *unique* (so-called *canonical*) representations. In [26] three such canonical forms have been defined, the first one (for four states) is depicted in Figure 4. This structure consists of a row of exponential stages with possibly different, but increasing transition rates, and an arbitrary initial distribution. In the following example $(\underline{\sigma}, \boldsymbol{S})$ is an acyclic representation and $(\underline{\gamma}, \boldsymbol{G})$ is its canonical form:

$$\underline{\sigma} = \begin{bmatrix} 0.1 & 0.2 & 0.7 \end{bmatrix}, \qquad \underline{\gamma} = \begin{bmatrix} 0.34 & 0.3 & 0.36 \end{bmatrix},$$
$$\boldsymbol{S} = \begin{bmatrix} -5 & 2 & 1 \\ 0 & -4 & 3 \\ 0 & 0 & -2 \end{bmatrix}, \qquad \boldsymbol{G} = \begin{bmatrix} -2 & 2 & 0 \\ 0 & -4 & 4 \\ 0 & 0 & -5 \end{bmatrix}. \tag{11}$$

MONOCYCLIC REPRESENTATIONS have been defined in [64]. They consists of feedback Erlang blocks (FEBs) arranged in a row. The i*th* feedback Erlang block FEB_i is characterized by a rate parameter ν_i, a size parameter k_i and a feedback probability z_i (see Figure 5, where $k_1 = 1, k_2 = 4, k_3 = 1$).

Monocyclic representations have a unique feature phrased by the following theorem.

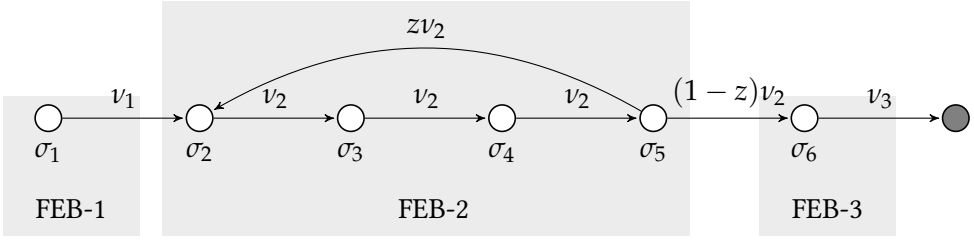


Figure 5.: Monocyclic representation of a PH distribution

Theorem 4. *[64] Every ME distribution has a finite-dimensional Markovian monocyclic representation, if the density function $f(x)$ satisfies $f(x) > 0,\ x \in (0, \infty)$.*

[64] provides a constructive algorithm to obtain the appropriate Markovian monocyclic representation. The transformation of a non-Markovian representation $(\underline{\sigma}, \boldsymbol{S})$ to a possibly larger Markovian (monocyclic) representation $(\underline{\gamma}, \boldsymbol{G})$ consists of the following steps.

1. In the first step matrix $\boldsymbol{G}$ is constructed. Each FEB implements one real eigenvalue or a conjugate complex eigenvalue pair of $\boldsymbol{S}$. Let us denote the jth eigenvalue of $\boldsymbol{S}$ by $-\lambda_j$, or, if it is a complex conjugate eigenvalue pair, by $-\lambda_j = a_j + b_j\mathbb{i}$ and $-\bar{\lambda}_j = a_j - b_j\mathbb{i}$.

   - If λ_j is real, the parameters of the jth FEB are $\nu_j = \lambda_j, k_j = 1, z_j = 0$.
   - If λ_j is complex, the parameters of the corresponding FEB are determined as

$$k_j = \text{the smallest integer for which } a_j/b_j > \tan(\pi/k_j), \tag{12}$$

$$\nu_j = \frac{1}{2}\left(2a_j - b_j \tan\frac{\pi}{k_j} + b_j \cot\frac{\pi}{k_j}\right), \tag{13}$$

$$z_j = \left(1 - \left(a_j - b_j \tan\frac{\pi}{k_j}\right)\right)^{k_j}. \tag{14}$$

   With these parameters matrix $\boldsymbol{G}$ is Markovian by construction and contains all eigenvalues of $\boldsymbol{S}$ with the proper multiplicities. However, the size of matrix $\boldsymbol{G}$ can be larger than the size of matrix $\boldsymbol{S}$, meaning that new eigenvalues are introduced. These extra eigenvalues can not play a role in the pdf, thus vector $\underline{\gamma}$ must ensure that their coefficients in the spectral form of the pdf are zeros.

2. The second step of the procedure is calculating the initial vector $\underline{\gamma}$. To this end, we need to obtain the transformation matrix $\boldsymbol{W}$ that transforms matrix $\boldsymbol{S}$ to matrix $\boldsymbol{G}$.

   According to Theorem 3 matrix $\boldsymbol{W}$ is the solution of $\boldsymbol{SW} = \boldsymbol{WG}$, $\boldsymbol{W}\mathbb{1} = \mathbb{1}$, which is a linear system of equations with regards to the entries of $\boldsymbol{W}$. With the presented construction of $\boldsymbol{G}$, this linear system always has a unique solution. If the size of $\boldsymbol{S}$ is N, and the size of $\boldsymbol{G}$ is M, equation $\boldsymbol{SW} = \boldsymbol{WG}$ has $N \times M$ unknowns and defines $N \times M$ equations. However, only $N \times M - N$ equations will be independent, since N eigenvalues of $\boldsymbol{S}$ and $\boldsymbol{G}$ are the same. By adding $\boldsymbol{W}\mathbb{1} = \mathbb{1}$ we get $N \times M$ independent equations and obtain a unique solution.

   The initial vector is then given by $\underline{\gamma} = \underline{\sigma}\boldsymbol{W}$, which may contain negative entries, hence may not be a valid Markovian probability vector.

3. The third, last step is necessary only if vector $\underline{\gamma}$ is not a proper probability vector. In this case, an Erlang tail (a number of extra phases with the same transition rates) needs to be appended to the row of FEBs. This Erlang tail is added to matrix $\boldsymbol{G}$, the corresponding transformation matrix $\boldsymbol{W}$ is re-calculated, and we get a new initial vector. It is proven in [64] that an appropriate Erlang tail always makes the representation Markovian, if $(\underline{\sigma}, \boldsymbol{S})$ defines a ME distribution with positive density. Unfortunately there is no explicit way to obtain the size and the rate parameters of the Erlang tail to be added. One can apply a simple heuristic algorithm that increases the size of the Erlang tail successively and applies the secant method to find the rate parameter that makes $\underline{\gamma}$ a valid probability vector.

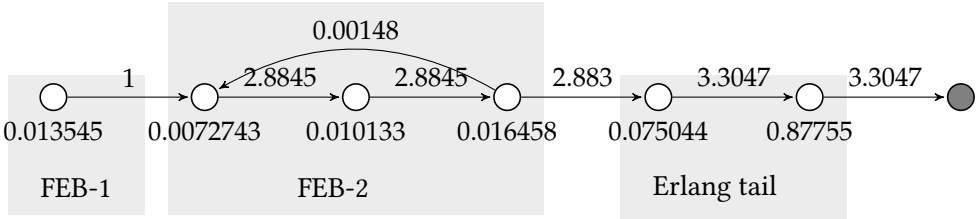


Figure 6.: Monocyclic representation of example (15)

As demonstration, the non-Markovian representation

$$\underline{\sigma} = \begin{bmatrix} 0.1 & 0.2 & 0.7 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -3 & 0.2 \\ 0 & -0.2 & -3 \end{bmatrix}, \tag{15}$$

is transformed to a monocyclic representation with the procedure outlined above. The resulting initial vector and transient generator are

$$\underline{\alpha} = \begin{bmatrix} 0.013545 & 0.0072743 & 0.010133 & 0.016458 & 0.075044 & 0.87755 \end{bmatrix},$$
$$\boldsymbol{G} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -2.8845 & 2.8845 & 0 & 0 & 0 \\ 0 & 0 & -2.8845 & 2.8845 & 0 & 0 \\ 0 & 0.0014803 & 0 & -2.8845 & 2.883 & 0 \\ 0 & 0 & 0 & 0 & -3.3047 & 3.3047 \\ 0 & 0 & 0 & 0 & 0 & -3.3047 \end{bmatrix}, \tag{16}$$

and the corresponding state transition graph is depicted in Figure 6. The original $(\underline{\sigma}, \boldsymbol{S})$ representation has a real eigenvalue ($\lambda_1 = -1$), and a complex conjugate eigenvalue pair ($\lambda_{2,3} = -3 \pm 0.2\mathrm{i}$). The first, degenerate FEB (lacking a real feedback) realizes λ_1. The second FEB realizes λ_2 and λ_3, but also introduces an extra eigenvalue. The initial vector ensures that the coefficient of this eigenvalue is zero in the spectral form of the pdf (see (6)). After the second FEB an Erlang tail follows, with the eigenvalues introduced there having zero coefficients as well. Without the Erlang tail (or with a shorter one) the initial vector remains non-Markovian, containing negative elements.

Theorem 4 has a striking consequence: every ME can be converted to a PH distribution with more states, blurring the difference between the two distribution classes.

2.2 CANONICAL FORMS

As seen in the previous sections, the $(\underline{\sigma}, \boldsymbol{S})$ representation of PH distributions is known to be non-unique and can be non-minimal, thus there can be many $(\underline{\alpha}, \boldsymbol{G})$ representations defining the same distribution. Furthermore, the number of parameters (non-determined elements) of this representation is $N^2 + N - 1$ when the size of vector $\underline{\sigma}$ and square matrix $\boldsymbol{S}$ is N (since $\boldsymbol{S}$ has N^2 elements and $\underline{\sigma}$ has $N - 1$), while the spectral form of the pdf of order N PH distributions (6) has only at most $2N - 1$ parameters.

To overcome these drawbacks unique, minimal, hence *canonical* representations are required. Canonical forms (among other benefits) make the PH fitting procedures more efficient,

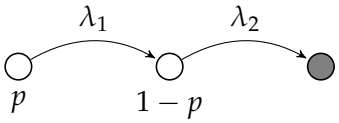


Figure 7.: Canonical representation of PH(2) distributions

since the underlying optimization procedure does not have to go back and forth between very different representations of almost the same distribution.

As mentioned earlier, canonical representations are available for a long time for acyclic PH distributions ([26]). For general (non-acyclic) PHs, however, finding canonical forms is more challenging.

2.2.1 *Canonical representation of PH(2) distributions*

According to [25], any ME(2) distribution can be transformed to an acyclic form, for which a canonical form exists due to [26]. This way the same canonical form is applicable for all (not only acyclic and even for non-Markovian) PH(2) representations.

Theorem 5. *The distribution sets ME(2), PH(2) and APH(2) are equivalent, i.e.,* $ME(2) \equiv PH(2) \equiv APH(2)$.

Proof. Based on the definition of these classes, we have $ME(2) \supset PH(2) \supset APH(2)$. Here, we only prove that any ME(2) distribution has an APH(2) representation.

From [25] we have that the LST $f^*(s)$ for a second-order representation $(\underline{\alpha}, \boldsymbol{S})$ has the form

$$f^*(s) = \frac{1 + s/\alpha}{(1 + s/\lambda_1) \cdot (1 + s/\lambda_2)}, \tag{17}$$

where λ_1 and λ_2 denote the eigenvalues of $-\boldsymbol{S}$. This LST corresponds to a valid density function if and only if λ_1, λ_2 and α are all real and

$$0 < \min(\lambda_1, \lambda_2) \leq \alpha \leq \infty \tag{18}$$

is satisfied.

Let us rewrite this LST as

$$f^*(s) = \frac{1 - \lambda_1/\alpha}{(1 + s/\lambda_1) \cdot (1 + s/\lambda_2)} + \frac{\lambda_1/\alpha}{(1 + s/\lambda_2)}.$$

This structure reveals an analogy to a Laplace transform of a Bernoulli mixture of a hypo-exponential density and an exponential density, which leads us to the following matrix-exponential representation

$$\underline{\alpha} = \begin{bmatrix} p & 1-p \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} -\lambda_1 & \lambda_1 \\ 0 & -\lambda_2 \end{bmatrix}, \tag{19}$$

with $p = 1 - \frac{\lambda_1}{\alpha}$. Figure 7 visualizes this acyclic PH(2) representation. It is easily verified that (3) with these settings for $\underline{\alpha}$ and $\boldsymbol{G}$ yields (17). Due to condition (18), i.e., $\lambda_1 \leq \alpha$, it follows $0 \leq \frac{\lambda_1}{\alpha} \leq 1$ so that (19) is indeed a valid APH(2) representation. □

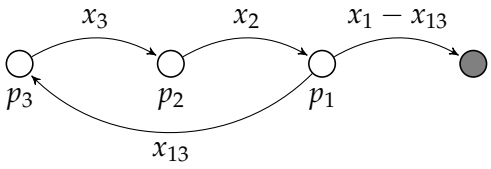


Figure 8.: The structure of the considered PH(3) distribution

In the example below the non-Markovian $(\underline{\sigma}, \boldsymbol{S})$ representation is transformed to the canonical form $(\underline{\gamma}, \boldsymbol{G})$:

$$\underline{\sigma} = \begin{bmatrix} -1.4 & 2.4 \end{bmatrix}, \qquad \underline{\gamma} = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix},$$
$$\boldsymbol{S} = \begin{bmatrix} -22 & 24 \\ -15 & 16 \end{bmatrix}, \qquad \boldsymbol{G} = \begin{bmatrix} -4 & 4 \\ 0 & -2 \end{bmatrix}. \tag{20}$$

The steps of the transformation were:

- creating matrix $\boldsymbol{G}$, where $-\lambda_1$ and $-\lambda_2$ are the eigenvalues of $\boldsymbol{S}$,
- obtaining the transformation matrix $\boldsymbol{B}$ by solving the linear system $\boldsymbol{BG} = \boldsymbol{SB}, \boldsymbol{B}\mathbb{1} = \mathbb{1}$,
- obtaining the initial vector of the canonical form from $\underline{\gamma} = \underline{\sigma}\boldsymbol{B}$.

2.2.2 *Canonical representation of PH(3) distributions*

Since the dominant eigenvalue must be real, the pdf of PH(2) distributions can not have complex eigenvalues. Order-3 PH distributions, however, can have complex eigenvalues. This essential difference leads to a more complicated canonical form, which is more involved to derive.

In [40] it is proven that every PH(3) distribution given by a *Markovian representation* can be transformed to a *unicyclic* form (see Figure 8), which is almost the same as a canonical acyclic structure, but has an extra feedback arc.

The theorem below summarizes the results of [40].

Theorem 6. *[40] If $(\underline{\sigma}, \boldsymbol{S})$ is a Markovian representation of a PH(3) distribution then it can be similarity transformed to the Markovian unicyclic representation*

$$\underline{\gamma} = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} -x_1 & 0 & x_{13} \\ x_2 & -x_2 & 0 \\ 0 & x_3 & -x_3 \end{bmatrix}, \tag{21}$$

where $x_1 \geq x_2 \geq x_3 > 0,\ 0 \leq x_{13} < x_1, 0 \leq \gamma_1, \gamma_2, \gamma_3, \gamma_1 + \gamma_2 + \gamma_3 = 1.$

The structure of the resulting unicyclic PH distribution is depicted in Figure 8.

Theorem 6 and the corresponding algorithm assume that the initial representation $(\underline{\sigma}, \boldsymbol{S})$ is Markovian; if this assumption is violated, hence the input is a non-Markovian representation, the algorithm returns invalid results with possibly complex entries. This assumption can be restrictive in many situations including moment matching (discussed later in Section 2.3), since the moment matching procedures typically return non-Markovian representations.

Hence, in the rest of this section we introduce an alternative algorithm that is able to perform the canonical transformation of *any* non-Markovian $(\underline{\sigma}, \boldsymbol{S})$ representations as well.

Let $\lambda_1, \lambda_2, \lambda_3$ denote the eigenvalues of $-\boldsymbol{S}$ which are ordered such that $\mathrm{re}\langle\lambda_1\rangle \geq \mathrm{re}\langle\lambda_2\rangle \geq \mathrm{re}\langle\lambda_3\rangle$ and a_0, a_1, a_2 the coefficients of the characteristic polynomial of $-\boldsymbol{S}$, i.e.,

$$a_0 = \lambda_1\lambda_2\lambda_3,\ a_1 = \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3,\ a_2 = \lambda_1 + \lambda_2 + \lambda_3. \tag{22}$$

The next lemma provides the similarity transformation matrix $\boldsymbol{B}$ connecting $(\underline{\sigma}, \boldsymbol{S})$ and $(\underline{\gamma}, \boldsymbol{G})$, and relates its elements with the coefficients of the characteristic polynomial.

Lemma 1. *The similarity transformation matrix $\boldsymbol{B}$, composed by the column vectors $[\underline{b}_1, \underline{b}_2, \underline{b}_3]$, to transform an arbitrary size 3 matrix $\boldsymbol{S}$ to a unicyclic representation is given by*

$$\begin{aligned}
\underline{b}_1 &= \frac{1}{x_{13} - x_1}\, \boldsymbol{S}\mathbb{1}, \\
\underline{b}_2 &= \frac{1}{(x_{13} - x_1)x_2}\, (x_1\boldsymbol{I} + \boldsymbol{S})\boldsymbol{S}\mathbb{1}, \\
\underline{b}_3 &= \frac{1}{(x_{13} - x_1)x_2x_3}\, (x_2\boldsymbol{I} + \boldsymbol{S})(x_1\boldsymbol{I} + \boldsymbol{S})\boldsymbol{S}\mathbb{1},
\end{aligned} \tag{23}$$

and

$$\begin{aligned}
x_{13} &= x_1 - \frac{a_0}{x_1^2 - a_2x_1 + a_1}, \\
x_2 &= \frac{a_2 - x_1 + \sqrt{(a_2 - x_1)^2 - 4(x_1^2 - a_2x_1 + a_1)}}{2}, \\
x_3 &= \frac{a_2 - x_1 - \sqrt{(a_2 - x_1)^2 - 4(x_1^2 - a_2x_1 + a_1)}}{2}.
\end{aligned} \tag{24}$$

The proof of the Lemma can be found in Appendix B.1.

The transformation matrix $\boldsymbol{B}$ and the transformed unicyclic representation $\boldsymbol{G}$ depend on the choice of x_1. [40] showed the following properties of PH(3) distributions and this similarity transform.

P1) When $\boldsymbol{S}$ is a Markovian generator then

$$\vartheta_u = \frac{a_2 + 2\sqrt{a_2^2 - 3a_1}}{3}, \tag{25}$$

$$\vartheta_0 = \frac{a_2 + \sqrt{a_2^2 - 3a_1}}{3}, \tag{26}$$

$$\vartheta_\ell = \begin{cases} \lambda_1, & \text{if } \lambda_1 \in \mathbb{R}, \\ \vartheta_0, & \text{if } \lambda_1 \in \mathbb{C} \end{cases} \tag{27}$$

are real and positive such that $\vartheta_\ell \leq \vartheta_u$.

P2) When $\vartheta_\ell \leq x_1 \leq \vartheta_u$ then the transformed generator matrix, $\boldsymbol{G} = \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$ is Markovian such that $x_1 \geq x_2 \geq x_3 > 0$.

Indeed, property P2 holds also for any non-Markovian matrix $\boldsymbol{S}$ if its eigenvalues satisfy the requirements of PH(3) distributions:

- λ_3, the dominant eigenvalue, is real and positive,
- $a_2^2 - 3a_1 \geq 0$.

Due to the fact that the similarity transform leaves the eigenvalues unchanged, this generalization of property P2 is a consequence of property P1 and Theorem 6.

We can summarize the results of [40] as follows. It defines a similarity transformation of PH(3) distributions to a unicyclic representation. This transformation depends on a parameter, x_1. [40] also defines the range of parameter x_1, $(\vartheta_\ell, \vartheta_u)$, where the transformed generator matrix is Markovian. The critical step is how to set parameter x_1 such that the initial vector is Markovian, i.e., is a proper probability vector. Unfortunately the way the algorithm in [40] sets x_1 is not sufficient when we have a non-Markovian $(\underline{\sigma}, \boldsymbol{S})$ representation.

Next we investigate the range of x_1 where the initial vector is Markovian.

Using the similarity matrix defined in (23) the elements of the initial vector $\underline{\gamma} = \underline{\sigma}\boldsymbol{B}$ are:

$$\gamma_1 = \frac{-\underline{\sigma}\boldsymbol{S}\mathbb{1}}{x_1 - x_{13}} = \frac{d_1}{x_1 - x_{13}}, \tag{28}$$

$$\gamma_2 = \frac{-\underline{\sigma}(x_1\boldsymbol{I} + \boldsymbol{S})\boldsymbol{S}\mathbb{1}}{(x_1 - x_{13})x_2} = \frac{x_1 d_1 + d_2}{(x_1 - x_{13})x_2}, \tag{29}$$

$$\gamma_3 = \frac{-\underline{\sigma}(x_2\boldsymbol{I} + \boldsymbol{S})(x_1\boldsymbol{I} + \boldsymbol{S})\boldsymbol{S}\mathbb{1}}{(x_1 - x_{13})x_2x_3} = \frac{x_1x_2d_1 + (x_1 + x_2)d_2 + d_3}{(x_1 - x_{13})x_2x_3}, \tag{30}$$

where $d_i = -\underline{\sigma}\boldsymbol{S}^i\mathbb{1}, i = \{1, 2, 3\}$. The derivatives of the density function at zero are closely related to these parameters since $f^{(i)}(0) = d_{i+1} = -\underline{\sigma}\boldsymbol{S}^{i+1}\mathbb{1}$. Consequently, for a Markovian $(\underline{\sigma}, \boldsymbol{S})$ pair

P3) either $d_1 > 0$, or $d_1 = 0$ and $d_2 \geq 0$,

must hold for having a non-negative density around zero.

The canonical form we propose in this section is based on the following theorem.

Theorem 7. *If $(\underline{\sigma}, \boldsymbol{S})$ has a Markovian representation, then the similarity transform with matrix $\boldsymbol{B}$, defined in (23), with parameter*

$$x_1 = \begin{cases} \max\{\vartheta_2, \vartheta_\ell\}, & \textbf{\textit{if}}\ \underline{\sigma}\boldsymbol{S}\mathbb{1} < 0, \\ \vartheta_\ell, & \textbf{\textit{if}}\ \underline{\sigma}\boldsymbol{S}\mathbb{1} = 0, \end{cases} \tag{31}$$

$$\vartheta_2 = -\frac{\underline{\sigma}\boldsymbol{S}^2\mathbb{1}}{\underline{\sigma}\boldsymbol{S}\mathbb{1}}, \tag{32}$$

provides a Markovian representation.

The proof of the Theorem is provided in Appendix B.2.

The corresponding transformation procedure is presented in Figure 1. If the procedure fails to produce a valid Markovian output then the input does not represent a PH(3) distribution. If the procedure completes, it gives back the canonical representation of the given PH(3) distribution, which is Markovian, minimal and unique.

If $\underline{\sigma}$ is an arbitrary vector and $\boldsymbol{S}$ is an arbitrary matrix of order three such that $(\underline{\sigma}, \boldsymbol{S})$ represents an order three PH distribution, then $(\underline{\gamma}, \boldsymbol{G})$ is a Markovian representation of this PH(3) distribution.

$(\underline{\gamma}, \boldsymbol{G})$ is unique, in the sense that for any $(\underline{\sigma}, \boldsymbol{S})$ representation of a PH(3) distribution the procedure provides the same $(\underline{\gamma}, \boldsymbol{G})$ pair.

Algorithm 1 Canonical transformation of PH(3) distributions

procedure CANONICAL–PH(3)–TRANSFORMATION($\underline{\sigma}, \boldsymbol{S}$) ▷ $(\underline{\sigma}, \boldsymbol{S})$ can be non-Markovian

$\lambda_1, \lambda_2, \lambda_3 \leftarrow$ decreasingly ordered eigenvalues of $-\boldsymbol{S}$

$a_0 \leftarrow \lambda_1 \lambda_2 \lambda_3, \quad a_1 \leftarrow \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3, \quad a_2 \leftarrow \lambda_1 + \lambda_2 + \lambda_3$

$\vartheta_u \leftarrow \frac{1}{3}\left(a_2 + 2\sqrt{a_2^2 - 3\,a_1}\right), \quad \vartheta_0 \leftarrow \frac{1}{3}\left(a_2 + \sqrt{a_2^2 - 3\,a_1}\right)$

$$\vartheta_\ell \leftarrow \begin{cases} \lambda_1 & \textbf{if } \lambda_1 \in \mathbb{R} \\ \vartheta_0 & \textbf{if } \lambda_1 \in \mathbb{C} \end{cases}$$

$$\vartheta_2 \leftarrow \begin{cases} -\underline{\sigma}\boldsymbol{H}^2\mathbb{1} / \underline{\sigma}\boldsymbol{H}\mathbb{1} & \textbf{if } \underline{\sigma}\boldsymbol{H}\mathbb{1} < 0 \\ 0 & \textbf{if } \underline{\sigma}\boldsymbol{H}\mathbb{1} = 0 \end{cases}$$

$x_1 \leftarrow \max\{\vartheta_2, \vartheta_\ell\}$

$x_{13} \leftarrow x_1 - a_0 \;/\; (x_1^2 - a_2\,x_1 + a_1)$

$x_2 \leftarrow \frac{1}{2}\left(a_2 - x_1 + \sqrt{(a_2 - x_1)^2 - 4\,(x_1^2 - a_2\,x_1 + a_1)}\right)$

$x_3 \leftarrow \frac{1}{2}\left(a_2 - x_1 - \sqrt{(a_2 - x_1)^2 - 4\,(x_1^2 - a_2\,x_1 + a_1)}\right)$

$\gamma_1 \leftarrow \underline{\sigma}\,\boldsymbol{S}\,\mathbb{1} \;/\; (x_{13} - x_1)$

$\gamma_2 \leftarrow \underline{\sigma}\,(x_1\,\boldsymbol{I} + \boldsymbol{S})\,\boldsymbol{S}\,\mathbb{1} \;/\; ((x_{13} - x_1)\,x_2)$

$\gamma_3 \leftarrow \underline{\sigma}\,(x_2\,\boldsymbol{I} + \boldsymbol{S})\,(x_1\,\boldsymbol{I} + \boldsymbol{S})\,\boldsymbol{S}\,\mathbb{1} \;/\; ((x_{13} - x_1)\,x_2\,x_3)$

return $\alpha = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix}$, $\boldsymbol{G} = \begin{bmatrix} -x_1 & 0 & x_{13} \\ x_2 & -x_2 & 0 \\ 0 & x_3 & -x_3 \end{bmatrix}$

end procedure

The PH(3) distributions are known to be determined by five parameters. E.g., the first five moments or the five coefficients of the Laplace rational transform uniquely determine a PH(3) distribution. Although not obvious at the first sight, the presented canonical form is also determined by exactly five independent parameters. In the unicyclic form [40] there are six parameters $(x_1, x_2, x_3, x_{13}, \gamma_1, \gamma_2)$ and in the transformation procedure just presented one of these parameters is additionally set to a special value. The following constraint decreases the number of parameters to five:

f1) $\lambda_1 \in \mathbb{R}, \gamma_2 < \vartheta_\ell \quad \rightarrow \quad x_{13} = 0,$

f2) $\lambda_1 \in \mathbb{C}, \gamma_2 < \vartheta_\ell \quad \rightarrow \quad x_1 = x_2,$

f3) $\vartheta_\ell < \gamma_2 \quad \rightarrow \quad \gamma_2 = 0.$

Indeed, these cases represent three different forms of the canonical representation. Applying the procedure for the example defined in (15) the output is

$$\alpha = \begin{bmatrix} 0.9683 & 0 & 0.0317 \end{bmatrix},$$
$$\boldsymbol{G} = \begin{bmatrix} -3.0221 & 0 & 0.027123 \\ 2.9572 & -2.9572 & 0 \\ 0 & 1.0207 & -1.0207 \end{bmatrix}, \tag{33}$$

hence form f3) is returned for this particular case.

It is an additional nice feature of the proposed canonical form that it is compatible with the widely used canonical representation of acyclic PH distributions [26], since when $(\underline{\sigma}, \boldsymbol{S})$

represents an order 3 acyclic PH distribution, then form f1 gives Cumani's canonical representation of that distribution. For instance, for $(\underline{\sigma}, \boldsymbol{S})$ defined in example (11) it returns the same $(\underline{\gamma}, \boldsymbol{G})$ as given there.

2.2.3 *Canonical representation of PH(4) distributions*

Based on the structure of the canonical representation of PH(3) distributions this section studies the following unicyclic PH(4) structure.

Let $(\underline{\sigma}, \boldsymbol{S})$ be a general matrix representation of a PH(4) distribution and $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ its eigenvalues. The characteristic polynomial of $\boldsymbol{S}$ is $x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ where

$$\begin{aligned} a_0 &= \lambda_1\lambda_2\lambda_3\lambda_4, && (34)\\ a_1 &= \lambda_1\lambda_2\lambda_3 + \lambda_1\lambda_2\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_2\lambda_3\lambda_4, && (35)\\ a_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 + \lambda_1\lambda_4 + \lambda_2\lambda_4 + \lambda_3\lambda_4, && (36)\\ a_3 &= \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 && (37) \end{aligned}$$

Theorem 8. *The $(\underline{\sigma}, \boldsymbol{S})$ representation can be transformed to the $(\underline{\gamma}, \boldsymbol{G})$ unicyclic form where $\underline{\gamma} = \underline{\sigma}\boldsymbol{B}$, $\boldsymbol{G} = \boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$, $\boldsymbol{B}\mathbb{1} = \mathbb{1}$, and matrix $\boldsymbol{G}$ has the form*

$$\boldsymbol{G} = \begin{bmatrix} -x_1 & 0 & x_{13} & x_{14} \\ x_2 & -x_2 & 0 & 0 \\ 0 & x_3 & -x_3 & 0 \\ 0 & 0 & x_4 & -x_4 \end{bmatrix}.$$

The similarity matrix of this transformation, $\boldsymbol{B} = [\underline{b}_1, \underline{b}_2, \underline{b}_3, \underline{b}_4]$, is composed by the following column vectors

$$\underline{b}_1 = \frac{-\boldsymbol{S}\mathbb{1}}{x_1 - x_{13} - x_{14}},\ \underline{b}_2 = \frac{(x_1\boldsymbol{I} + \boldsymbol{S})\underline{b}_1}{x_2},$$

$$\underline{b}_3 = \frac{(x_2\boldsymbol{I} + \boldsymbol{S})\underline{b}_2}{x_3},\ \underline{b}_4 = \frac{(x_3\boldsymbol{I} + \boldsymbol{S})\underline{b}_3}{x_4} - \frac{x_{13}\underline{b}_1}{x_4},$$

where x_1 and x_{13} are arbitrary parameters and x_{14}, x_2, x_3, x_4 are the solution of the following set of equations

$$\begin{aligned} a_0 &= (x_1 - x_{13} - x_{14})x_2x_3x_4, && (38)\\ a_1 &= (x_1 - x_{13})x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4, && (39)\\ a_2 &= x_1x_2 + x_1x_3 + x_2x_3 + x_1x_4 + x_2x_4 + x_3x_4, && (40)\\ a_3 &= x_1 + x_2 + x_3 + x_4\,. && (41) \end{aligned}$$

Proof. The coefficients of the characteristic polynomial of $\boldsymbol{G}$ are given at the right hand side of (38)-(41). $\boldsymbol{S}$ and $\boldsymbol{G}$ are similar since their characteristic polynomials are identical due to (38)-(41). The columns of the similarity matrix $\boldsymbol{B}$ can be obtained from the columns of the matrix equation $\boldsymbol{S}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{G}$, which are

$$\begin{aligned} \boldsymbol{S}\underline{b}_1 &= -x_1\underline{b}_1 + x_2\underline{b}_2\,, && (42)\\ \boldsymbol{S}\underline{b}_2 &= -x_2\underline{b}_2 + x_3\underline{b}_3\,, && (43)\\ \boldsymbol{S}\underline{b}_3 &= -x_3\underline{b}_3 + x_4\underline{b}_4 + x_{13}\underline{b}_1\,, && (44)\\ \boldsymbol{S}\underline{b}_4 &= -x_4\underline{b}_4 + x_{14}\underline{b}_1\,. && (45) \end{aligned}$$

Summing up (42)-(45) and using $\boldsymbol{B}\mathbb{1} = \mathbb{1}$ we have

$$\boldsymbol{S}\mathbb{1} \quad = \quad -x_1\underline{b}_1 + x_{13}\underline{b}_1 + x_{14}\underline{b}_1\,, \tag{46}$$

from which $\underline{b}_1 = \dfrac{-\boldsymbol{S}\mathbb{1}}{x_1 - x_{13} - x_{14}}$. Consecutively substituting the result into (42)-(44) we obtain $\underline{b}_2, \underline{b}_3, \underline{b}_4$, respectively. □

Corollary 1. *Starting from (38) - (41) and having x_1 and x_{13} fixed, x_{14}, x_2, x_3, x_4 are obtained as the solution of an order-6 equation.*

Consequently, there is no symbolic transformation method to the $(\gamma, \boldsymbol{G})$ unicyclic form.

Corollary 1 remains valid also when $x_{13} = 0$.

We have implemented the transformation method defined in Theorem 8 and additionally we have implemented transformation methods to the following simple order-4 generators

$\boldsymbol{G_{14}} = \boldsymbol{G}$ with $x_{13} = 0$,

$$\boldsymbol{G_{13}} = \begin{bmatrix} -x_1 & 0 & x_{13} & 0 \\ x_2 & -x_2 & 0 & 0 \\ 0 & x_3 & -x_3 & 0 \\ 0 & 0 & x_4 & -x_4 \end{bmatrix}, \quad \boldsymbol{G_{24}} = \begin{bmatrix} -x_1 & 0 & 0 & 0 \\ x_2 & -x_2 & 0 & x_{24} \\ 0 & x_3 & -x_3 & 0 \\ 0 & 0 & x_4 & -x_4 \end{bmatrix}.$$

Having these transformation methods we checked if general PH(4) distributions can be transformed to the given specific forms. We found that none of the $\boldsymbol{G_{13}}$, $\boldsymbol{G_{14}}$ and $\boldsymbol{G_{24}}$ forms are sufficiently general to transform all PH(4) distributions into that form. However, we found that it is usually impossible to transform between these forms. I.e., having a PH(4) distribution whose generator has the form of $\boldsymbol{G_{24}}$, it is commonly not possible to transform it to the form of $\boldsymbol{G_{13}}$ and $\boldsymbol{G_{14}}$, and so on.

In contrast, we found that the $(\gamma, \boldsymbol{G})$ representation, with properly chosen x_1 and x_{13} parameters, is general enough to cover all PH(4) examples we tried with.

The $(\gamma, \boldsymbol{G})$ representation is defined by nine parameters, x_1, x_2, x_3, x_4, x_{13}, x_{14}, γ_1, γ_2, γ_3.

Assuming that the $(\gamma, \boldsymbol{G})$ representation is a candidate for the canonical representation of PH(4) distributions and that the canonical representation of PH(4) distributions contains the minimal number of parameters (which is seven), two additional constraints should apply. Some of the possible constraints are $x_{13} = 0$, $x_{14} = 0$, $\pi_2 = 0$, $\pi_3 = 0$, $x_1 = x_2$, $x_2 = x_3$, $x_3 = x_4$. Considering only these constraints we have a wide variety of different constraint-pairs. Some of them might be too restrictive, but e.g., $x_{13} = x_{14} = 0$ results in the acyclic subclass of PH(4) distributions.

2.3 PH FITTING WITH CANONICAL FORMS

The applicability of PH distributions for modeling real systems relies on efficient *fitting procedures*. A fitting procedure constructs a PH distribution based on empirical samples or based on an other known distribution. This section demonstrates the benefits of canonical forms in PH distribution fitting.

2.3.1 *Moment matching*

In case of moment matching a PH distribution is created that has the same moments as the target distribution. Recall that PH(2) distributions have 3, and PH(3) distributions have 5 free parameters, thus they can match 3 and 5 moments, respectively.

Moment matching is not straight forward, since it involves the solution of a polynomial system of equations (see (4)). Apart from the low order cases, such equations have no explicit solutions.

Nevertheless, there exists a procedure, published in [82], that solves the moment problem. For a given set of $\{m_1, \dots m_{2N-1}\}$ moments this algorithm creates a size N vector and matrix pair, $(\underline{\sigma}, \boldsymbol{S})$, for which $i!\underline{\sigma}(-\boldsymbol{S})^{-i}\mathbb{1} = m_i,\ i = 1, \dots, 2N-1$ holds[1].

The output of the procedure $(\underline{\sigma}, \boldsymbol{S})$, however, while providing the appropriate moments, can have arbitrary elements. It is either a non-Markovian representation of a PH distribution, or not even a distribution at all (as the density is negative at some points). Transforming this $(\underline{\sigma}, \boldsymbol{S})$ to the canonical representation gives the answer to this question. If the transformation to the canonical form fails (the result is not a Markovian representation), then $(\underline{\sigma}, \boldsymbol{S})$ is either not a valid distribution or does not have an order N Markovian representation.

In the next example the moments to fit are extracted from a real measurement trace file, which captures the packet inter-arrival times over two hours of wide-area TCP traffic[2]. In this trace the first five moments of the inter-arrival times are $\{1, 2.942, 16.84, 150.73, 1876.8\}$.

First a PH(2) is created based on the first three moments. The procedure of [82] returns

$$\underline{\sigma}^{(2)} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \quad \boldsymbol{S}^{(2)} = \begin{bmatrix} -1.5779 & -0.11239 \\ -0.5 & -0.5 \end{bmatrix},$$

which is clearly non-Markovian. Transforming it to the canonical form described in Section 2.2.1 gives

$$\underline{\alpha}^{(2)} = \begin{bmatrix} 0.1736 & 0.8264 \end{bmatrix}, \quad \boldsymbol{G}^{(2)} = \begin{bmatrix} -0.45017 & 0.45017 \\ 0 & -1.6277 \end{bmatrix},$$

which is a PH(2) distribution with a Markovian representation, matching the first three target moments. Repeating these steps with 5 moments and 3 states yields

$$\underline{\sigma}^{(3)} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}, \quad \boldsymbol{S}^{(3)} = \begin{bmatrix} -1.7356 & 0.34074 & -0.95214 \\ -0.18575 & -0.63031 & -0.042169 \\ -0.48092 & -0.036353 & -0.6245 \end{bmatrix}$$

in the first step and leads to

$$\underline{\alpha}^{(3)} = \begin{bmatrix} 0.71787 & 0.26156 & 0.02057 \end{bmatrix}, \ \boldsymbol{G}^{(3)} = \begin{bmatrix} -2.0185 & 0 & 0 \\ 0.63653 & -0.63653 & 0 \\ 0 & 0.33542 & -0.33542 \end{bmatrix},$$

1 The original procedure in [82] obtains the result in a slightly different form with the closing vector being different from $\mathbb{1}$, that can be transformed to the ME representation used in this section by applying a simple similarity transformation

2 Downloaded from http://ita.ee.lbl.gov/html/contrib/LBL-TCP-3.html

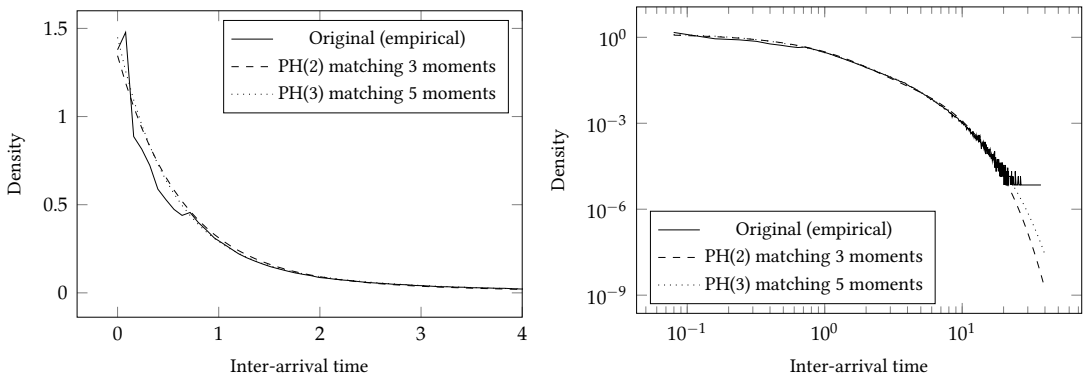


Figure 9.: Results of moment matching with canonical forms

after the canonical transformation by Algorithm 1. Figure 9 depicts the original (empirical) pdf and the pdf of PH distributions $(\gamma^{(2)}, \boldsymbol{G}^{(2)})$ and $(\gamma^{(3)}, \boldsymbol{G}^{(3)})$. Both PH distributions approximate the target distribution relatively well, the 3-state one being a bit closer, especially in the log-log plot.

2.3.2 *Fitting the density function*

There is a large number of PH distribution fitting methods available in the literature (for a survey see [56]). Some of them operate on the full PH class while others look for the solution in a subclass of the PH distributions. The most commonly used sub-classes for fitting purposes are the APH class, the hyper-exponential distributions and the hyper-Erlang structure. At the first sight these structural restrictions seem to decrease the efficiency of the fitting methods, since they look for the best fit in a smaller class of distributions. However, based on practical experiments, the opposite seems to be true: fitting a distribution with a restricted PH sub-class often provides better results, both in terms of distance and speed. The reason is that methods optimizing the full PH generator matrix and initial probability vector are often circling around different representations of the same distribution. Methods operating on the restricted PH sub-classes have an easier job, since they optimize fewer parameters.

The canonical forms of PH(2) and PH(3) distributions can be utilized to develop more efficient PH fitting methods. These canonical forms are minimal representations having minimal number of parameters while covering the whole PH(2) and PH(3) classes, consequently the optimization methods find the solution more easily.

To show the benefits of canonical forms in distribution fitting some numerical examples are presented. We developed a simple fitting method in MATLAB based on the line search algorithm with the subject function set to the relative entropy. Relative entropy ([14], also known as the Kullback–Leibler divergence) is a popular quantity to measure the goodness of fit (for discrete samples it is equal to the log-likelihood). It is defined by

$$D(f,\hat{f}) = \int_0^\infty f(x) |\log \frac{f(x)}{\hat{f}(x)}| dx, \tag{47}$$

where $\hat{f}(t)$ is the density function of the fitting PH distribution and $f(t)$ denotes the pdf of the distribution to fit.

The initial point was the best selected from hundred random PHs distributions.

	W1	U1	ME	BC
Full PH3:	$1.9532 \cdot 10^{-3}$	0.16659	0.89605	0.14093
Form f1):	$1.9531 \cdot 10^{-3}$	0.16659	0.90059	0.14087
Form f2):	$1.9532 \cdot 10^{-3}$	0.16659	0.89605	0.27127
Form f3):	$4.5383 \cdot 10^{-3}$	0.16659	0.89605	0.28222

Table 1.: Minimal distance obtained by optimizing with different representations

	W1	U1	ME	BC
Full PH3:	$1.2 \cdot 10^{-4}$	$1.4 \cdot 10^{-5}$	$1.9 \cdot 10^{-4}$	$4.3 \cdot 10^{-3}$
Form f1):	$2.6 \cdot 10^{-7}$	$1.4 \cdot 10^{-5}$	$6.9 \cdot 10^{-5}$	$3.5 \cdot 10^{-3}$
Form f2):	$8.5 \cdot 10^{-7}$	$3.7 \cdot 10^{-4}$	$4.1 \cdot 10^{-5}$	$6.2 \cdot 10^{-3}$
Form f3):	$2.3 \cdot 10^{-4}$	$3.2 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$6.3 \cdot 10^{-4}$

Table 2.: Distance variances obtained by optimizing with different representations

During the numerical experiments the target distributions were W1, U1 and ME distributions defined in [14]:

$$f_{W1}(x) = \frac{\beta}{\eta}\left(\frac{x}{\eta}\right)^{\beta-1} e^{-\left(\frac{x}{\eta}\right)^{\beta}} \quad \text{with } \eta = 1, \beta = 1.5,$$
$$f_{U1}(x) = 1, \qquad 0 \le x \le 1,$$
$$f_{ME}(x) = \left(1 + \frac{1}{(2\pi)^2}\right)(1 - \cos(2\pi x))e^{-x}.$$

A non-synthetic distribution taken from real time measurements is included in the experiment as well. These real time measurements record one million packet arrivals on an Ethernet network[3], and will be referred to as BC in the sequel.

Since in case of PH(3) there are three different canonical forms, the optimization has to be performed with all three structures and the best fit should be selected as a final result.

The optimization has been performed 100 times with different random initial points. The best (minimal) distance obtained out of the 100 runs is shown in Table 1. In case of the W1 distribution the f1 canonical form turned out to be the best. All representations gave the same result in the U1 case. For the ME distribution, the APH structure (f1) was not able to capture the characteristics of the target distribution, but all other representations returned the same result. For real traffic fitting (BC case) the APH was found to be the most suitable, the full representation is not far behind. These examples also demonstrate the capabilities of the built-in optimization function of MATLAB, it found the solution even with redundant (non-minimal) representations.

According to Tables 2 and 3, however, the benefits of canonical forms in optimization are clear. The optimization finds the solution in fewer iterations (Table 3), and the solution depends less on the initial guess (Table 2).

3 Downloaded from http://ita.ee.lbl.gov/html/contrib/BC.html

	W1	U1	ME	BC
Full PH3:	150.98	110.4	117.49	146.6
Form f1):	52.04	43.34	40.56	119.62
Form f2):	64.27	55.22	52.53	111.14
Form f3):	108.49	71.33	71.84	122.93

Table 3.: Number of iterations when optimizing with different representations

2.4 PH FITTING WITH A FLEXIBLE STRUCTURE

Flexible structure PH fitting methods use representations that have *more* free parameters than the number of parameters to match. A set of parameters are set to ensure the matching of moments or some other subject function, while the remaining parameters add extra degrees of freedom to obtain a valid PH distribution to match *any* set of moments. Such a method is described in [51], which uses a representation called *mixture of Erlang distributions of common order* (MECO). Mixing K Erlang distributions of common order has $2K$ parameters: $K-1$ initial probabilities (which Erlang component to choose, sums up to 1), the intensity parameters of the Erlang distributions (there are K of them), and the common order of the Erlang distributions (1 parameter). With these $2K$ parameters this procedure can match $2K-1$ moments. The free parameter not involved in moment matching is the order parameter, which is common for all Erlang components. The moment matching is performed with order=1, order=2, etc., the order is increased till a Markovian solution is found. If order R Erlang branches are needed to get a Markovian representation, the procedure results in an order $N = K \cdot R$ PH distribution. It is guaranteed that, with appropriately large order, this procedure is able to match any set of $2K-1$ moments belonging to a positive distribution. An other method operating on a flexible structure has been published in [13]. It matches 3 moments with an exponential and an Erlang distribution connected after each other in a row. The degree of freedom is the order of the Erlang component again. It is proven that, by choosing the order of the Erlang component appropriately large, it is possible to match any 3 moments with this structure.

In this section we propose a new sub-class of PH distributions for matching any number of moments.

Definition 4. *Random variable $\mathcal{F}$ has order-K generalized hyper-Erlang distribution iff its density function is*

$$f(x) = \sum_{i=1}^{K} \sigma_i \frac{(\lambda_i x)^{r_i-1}}{(r_i-1)!} \lambda_i e^{-\lambda_i x}, \tag{48}$$

with $f(x) \geq 0$ and $\int_0^\infty f(x)dx = 1$. For the parameters we have that $\lambda_i \in \mathbb{C}, re\langle\lambda_j\rangle \geq 0, \sigma_j \in \mathbb{C}, \sum_{i=1}^{K} \sigma_i = 1$ and $r_i \in \mathbb{N}$ for $i = 1, \ldots, K$.

Thus, generalized hyper-Erlang distributions (GHErD) are similar to hyper-Erlang distributions, the difference is that coefficients σ_i do not need to be valid probabilities, and that λ_i can be complex as well. The kth moment of generalized hyper-Erlang distributions is calculated as

$$m_k = \int_0^\infty x^k f(x)dx = \sum_{i=1}^{K} \sigma_i \frac{(k+r_i-1)!}{(r_i-1)!} \frac{1}{\lambda_i^k}, \quad k \geq 0. \tag{49}$$

(Note that $m_0 = 1$).

2.4.1 *Solution of the moment matching problem with fixed r_i parameters*

For matching moments $m_1, \ldots, m_{2K-1}$ with order-K GHErD having the r_i parameters fixed we have to solve a system of polynomial equations defined by (49) for $k = 0, \ldots, 2K-1$, such that the unknown variables are $\underline{\lambda} = \{\lambda_i, i = 1, \ldots, K\}$ and $\underline{\beta} = \{\sigma_i, i = 1, \ldots, K\}$, which give $2K$ unknowns in total.

Due to the structure of the system of polynomial equations it is not possible to derive an explicit solution for arbitrary K. However, there are excellent tools available that are able to solve polynomial systems numerically[4]. It is important to emphasize that this does not mean that we are applying a non-linear programming or other optimization methods to find the solution of the moment matching problem (as it was done in [18]). What we are doing is the numerical solution of the polynomial system, that is able to provide *all* the solutions of the system of polynomial equations.

This polynomial system has typically several solutions, and it is also possible that it has no solutions at all (it can be inconsistent). If it does have solutions, each solution either defines a valid PH distribution, or it does not. To decide if a solution is valid, we try to obtain a Markovian monocyclic representation by using the method described in Section 2.1.4. From a solution given by vectors $\underline{\lambda}$ and $\underline{\beta}$ the initial (non-Markovian) representation is obtained in a direct way as

$$\underline{\sigma} = \left[\sigma_1 \underbrace{0 \ldots 0}_{r_1-1} \sigma_2 \underbrace{0 \ldots 0}_{r_2-1} \sigma_K \underbrace{0 \ldots 0}_{r_K-1}\right],$$

$$\boldsymbol{S} = \left[\begin{array}{ccc|ccc|c|ccc}
-\lambda_1 & \lambda_1 & & & & & & & & \\
 & \ddots & \ddots & & & & & & & \\
 & & -\lambda_1 & & & & & & & \\
\hline
 & & & -\lambda_2 & \lambda_2 & & & & & \\
 & & & & \ddots & \ddots & & & & \\
 & & & & & -\lambda_2 & & & & \\
\hline
 & & & & & & \ddots & & & \\
\hline
 & & & & & & & -\lambda_K & \lambda_K & \\
 & & & & & & & & \ddots & \ddots \\
 & & & & & & & & & -\lambda_K
\end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} r_1 \\ \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} r_2 \\ \\ \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} r_K \end{array} \tag{50}$$

If the output $(\underline{\gamma}, \boldsymbol{G})$ of the algorithm of Section 2.1.4 is Markovian, then we found a valid solution. Given vector $\underline{r} = \{r_i, i = 1, \ldots, K\}$ it may happen that several different PH distributions are found, but it is also possible that no solutions exist. In the latter case the entries of vector $\underline{r}$ need to be increased to obtain a valid solution.

2.4.2 *Optimizing the r_i parameters*

Finding the appropriate vector $\underline{r}$ can be made automatic as well. In this case the user just has to enter a single parameter, R, and the algorithm repeats the moment matching with all vectors $\underline{r}$ satisfying $\sum_{i=1}^{K} r_i \leq R$.

4 For this purpose, we are using PHCpack (see [85]), which is a multi-platform open-source tool and is under continuous development and refinement

As the MECO is a sub-class of GHErD, and according to [51] it is always possible to find a MECO for any set of moments, this procedure always finds a Markovian solution with an appropriately large R parameter.

Algorithm 2 GHErD fitting algorithm based on moment matching

procedure FITGHERD($m_1, \ldots, m_{2K-1}, R, D(\cdot)$)
  $res \leftarrow \emptyset$
  for all $\underline{r} = [r_1 \quad \ldots \quad r_K]$ with $\sum_{i=1}^{K} r_i \leq R$ **do**
    $\{(\underline{\lambda}, \underline{\beta})_i\} \leftarrow$ Solve polynomial equations of (49) for $k = 0, \ldots, 2K-1$
    for each solution $(\underline{\lambda}, \underline{\beta})$ **do**
      $(\underline{\sigma}, \boldsymbol{S}) \leftarrow$ Create non-Markovian representation based on (50)
      $(\gamma, \boldsymbol{G}) \leftarrow$ Transform $(\underline{\sigma}, \boldsymbol{S})$ to a monocyclic representation (Section 2.1.4)
      if $(\gamma, \boldsymbol{G})$ is Markovian **then**
        Add $(\gamma, \boldsymbol{G})$ to set res
      end if
    end for
  end for
  if $res = \emptyset$ **then**
    error "No solutions found up to size R. Parameter R has to be increased."
  end if
  $(\gamma, \boldsymbol{G}) \leftarrow \arg\min_{(\gamma, \boldsymbol{G}) \in res} D(\gamma, \boldsymbol{G})$
  return $(\gamma, \boldsymbol{G})$
end procedure

The proposed algorithm is depicted in Algorithm 2. First, the algorithm solves the moment matching problem with different $\underline{r}$ vectors up to $\sum_{i=1}^{K} r_i \leq R$ and collects all solutions that have a Markovian representation in set res. Notice that all solutions in res match the first $2K-1$ moments of the target distribution. In the second step (last line of the algorithm) the best solution is selected according to a secondary distance function $D(\cdot)$. Any distance function can be used that quantifies the distance between two distributions. Two possible distance functions are:

- Moment distance (MD): the sum of the squared relative difference of the moments up to moment M. Denoting the kth moment of the target distribution by $\hat{m}_k$ this means

$$D(\gamma, \boldsymbol{G}) = \sum_{k=1}^{M} \left(\frac{m_k^{(\gamma, \boldsymbol{G})} - \hat{m}_k}{\hat{m}_k} \right)^2. \tag{51}$$

  (Note that for $k \leq 2K-1$ we have $m_k^{(\gamma, \boldsymbol{G})} = \hat{m}_k$).

- Relative entropy (RE): as defined by (47).

Both distance functions have their advantage. The moment distance relies only on the moments, thus the exact shape of the target pdf is not required. The relative entropy, however, quantifies the similarity of the shapes of the density functions better. Both distance functions will be evaluated in the subsequent numerical examples.

# of moments to match	# of different $\underline{r}$ vectors	# of valid solutions	Execution speed
3	100	88	28 sec
5	237	688	337 sec
7	408	3920	810 min

Table 4.: The number of different $\underline{r}$ vectors, the number of valid solutions, and the total execution time in the LBL example

2.4.3 *Numerical examples*

In this section we apply the presented fitting method on two well known traffic measurement traces, the BC-pAug89 trace (also used in Section 2.3.2) and the LBL-TCP-3 trace (used in Section 2.3.1).

Algorithm 2 has been implemented in MATLAB. To solve the system of polynomial equations we used PHCpack v2.3.76[5]. All the results have been obtained on an average PC with a CPU clocked at 3.4 GHz and 4 GB of memory.

In the subsequent case studies the following 4 flexible moment matching-based PH fitting methods are compared:

1. The presented method, where the PH distribution with the smallest moment distance (up to 10 moments) is selected from the set of all PH distributions matching the moments of the trace;

2. The presented method, where the PH distribution with the lowest relative entropy is selected from the set of all PH distributions matching the moments of the trace;

3. Moment matching with a mixture of Erlang distributions of common order (MECO, [51]);

4. The method of [13], which is able to match the first three moments only.

EXPERIMENTS WITH THE LBL TRACE

In this example the R parameter (the sum of the multiplicities of the eigenvalues) is set to 20. According to Algorithm 2 this means that the moment matching is performed with all vectors $\underline{r}$ satisfying $\sum_{i=1}^{K} r_i \leq R$. With a given vector $\underline{r}$, the moment matching problem can result in several different valid PH distributions. At the end we have a large number of PH distributions from which we can select the best according to some distance function. Table 4 shows how many $\underline{r}$ vectors and valid solutions there are, and how long the execution time of the algorithm is.

The numerical results are shown in Table 5. When the first three moments were matched, all methods found the same solution. Even if a large number of $\underline{r}$ vectors have been checked by our method, the best solution has been found to be the same according to both distance functions.

5 It can be obtained from http://homepages.math.uic.edu/~jan/download.html

Num. of moms.	Method	MD	RE	Num. of states
3	Our method (MD)	1.786	0.3024	2 ($\underline{r} = [1\ \ 1]$)
	Our method (RE)	1.786	0.3024	2 ($\underline{r} = [1\ \ 1]$)
	MECO [51]	1.786	0.3024	2 ($\underline{r} = [1\ \ 1]$)
	ErlExp [13]	1.786	0.3024	2
5	Our method (MD)	0.0072	0.0984	3 ($\underline{r} = [1\ \ 1\ \ 1]$)
	Our method (RE)	0.0386	0.0953	5 ($\underline{r} = [1\ \ 1\ \ 2]$)
	MECO [51]	0.0072	0.0984	3 ($\underline{r} = [1\ \ 1\ \ 1]$)
7	Our method (MD)	8.26×10^{-6}	3.9727	20 ($\underline{r} = [2\ \ 2\ \ 3\ \ 13]$)
	Our method (RE)	0.00499	0.0727	8 ($\underline{r} = [1\ \ 1\ \ 3\ \ 3]$)
	MECO [51]	0.00475	0.1339	8 ($\underline{r} = [2\ \ 2\ \ 2\ \ 2]$)

Table 5.: Results of the fitting of the LBL trace

When 5 moments were matched, the MECO matching method returned a hyper-exponential distribution that was found to be optimal by our method as well according to the distance of moments. However, our method was able to find a PH distribution that has lower relative entropy. This PH distribution has 5 states and the corresponding $\underline{r}$ vector is $\underline{r} = [1\ \ 1\ \ 2]$. The sum of the elements of vector $\underline{r}$ is only 4, which means that a new eigenvalue has been introduced and the size of the PH has been increased by 1 to obtain a Markovian representation (this new eigenvalue cancels out, it has no effect on the pdf).

The advantage and the flexibility of the proposed method can be seen the best when 7 moments are matched. This method was able to find a PH distribution with significantly lower moment distance, and an other one with significantly lower relative entropy as well.

Figures 10, 11 and 12 plot the density functions belonging to the methods discussed, both on linear and on logarithmic scale. While the tail of the pdf is fitted well by all methods, the plots differ significantly in the body of the pdf. Based on a visual comparison, the solution found by our method by matching 7 moments and selecting the best according to the RE distance seems to capture the shape of the pdf best.

EXPERIMENTS WITH THE BC TRACE

The numerical results corresponding to the BC trace are summarized in Table 6. When fitting the first 3 moments, the same hyper-exponential distribution turned out to be optimal by all the methods involved into the comparison. When fitting 5 phases, the proposed method has found a PH distribution with very slightly lower relative entropy. In the case when 7 moments are matched, we have a PH distribution with better moment distance, and an other one with significantly better relative entropy than the MECO based method.

The density functions corresponding to the investigated cases are depicted in Figure 13, 14 and 15. Figure 14 demonstrates how different the shapes of the density functions can be even if the first 7 moments are the same. The MECO-based method looks to be the least successful in this example, while the proposed method with the relative entropy based selection managed to capture the characteristics of the density function reasonably well.

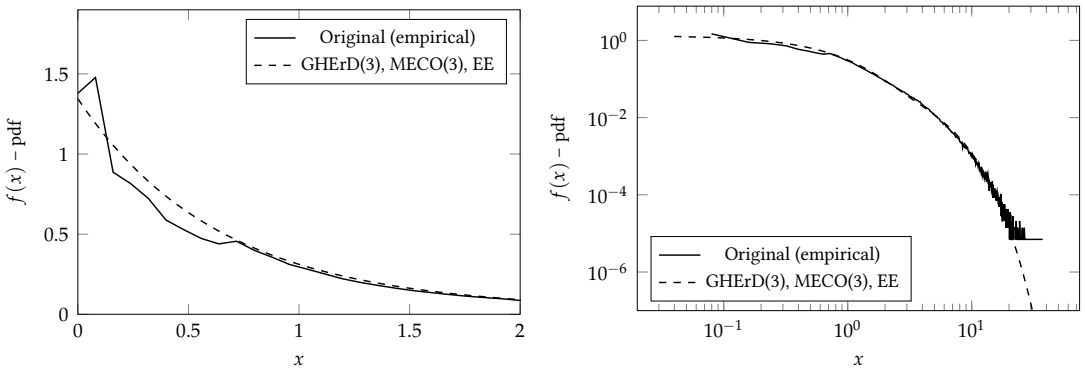


Figure 10.: Comparison of the density functions matching 3 moments of the LBL trace

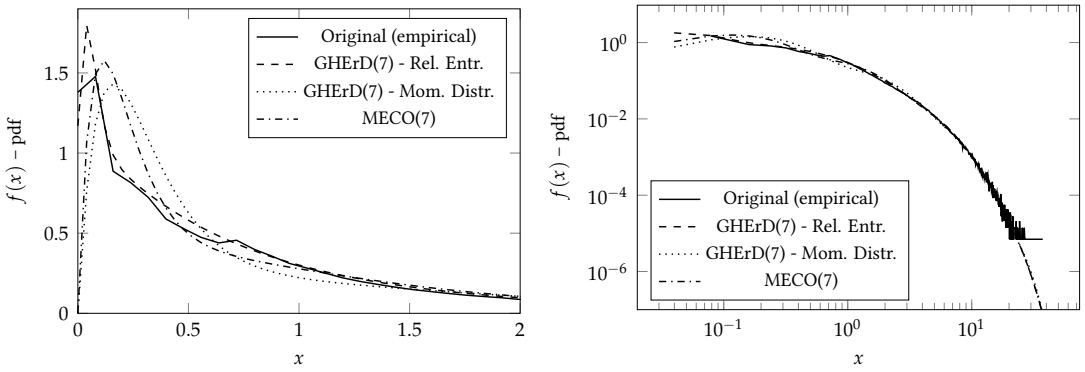


Figure 11.: Comparison of the density functions matching 7 moments of the LBL trace

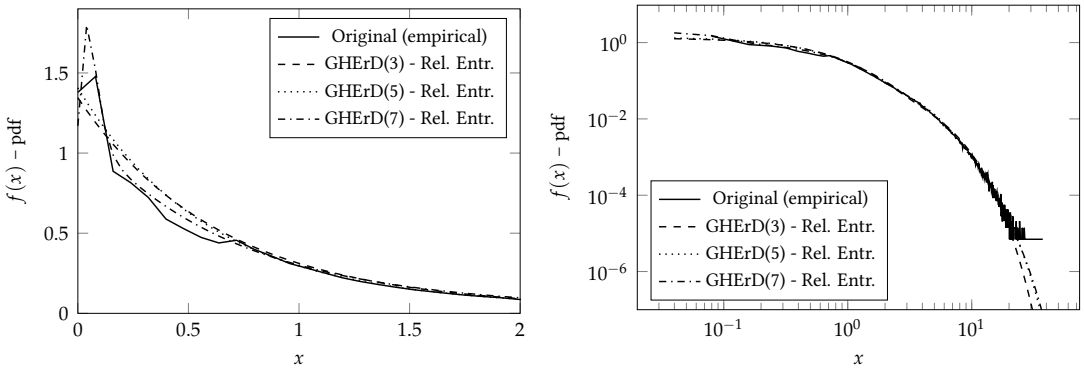


Figure 12.: Comparison of the results of our method, matching 3, 5 and 7 moments

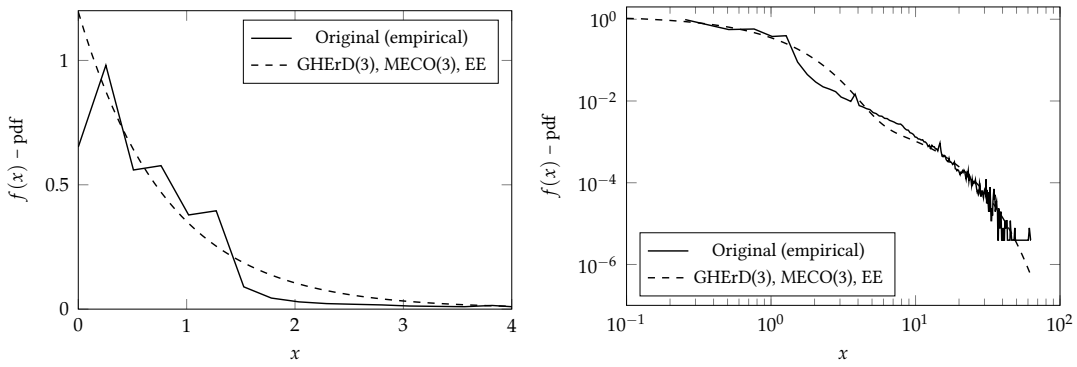


Figure 13.: Comparison of the density functions matching 3 moments of the BC trace

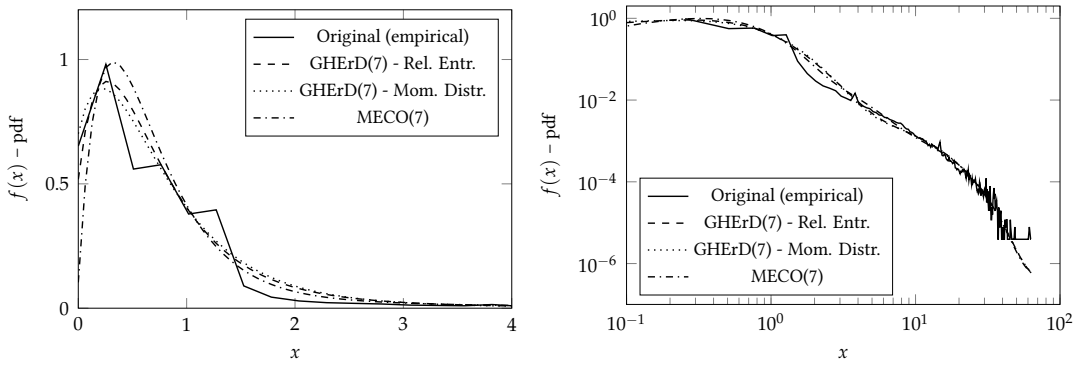


Figure 14.: Comparison of the density functions matching 7 moments of the BC trace

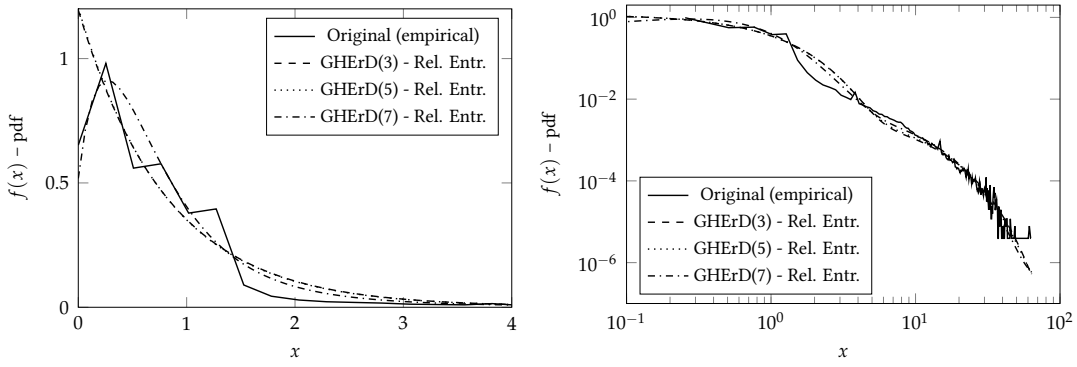


Figure 15.: Comparison of the results of our method, matching 3, 5 and 7 moments

Num. of moms.	Method	MD	RE	Num. of states
3	Our method (MD)	3.0509	0.30244	2 ($\underline{r} = [1\ \ 1]$)
	Our method (RE)	3.0509	0.30244	2 ($\underline{r} = [1\ \ 1]$)
	MECO [51]	3.0509	0.30244	2 ($\underline{r} = [1\ \ 1]$)
	ErlExp [13]	3.0509	0.30244	2
5	Our method (MD)	0.00198	0.30521	14 ($\underline{r} = [1\ \ 4\ \ 8]$)
	Our method (RE)	55.4699	0.30212	5 ($\underline{r} = [1\ \ 1\ \ 2]$)
	MECO [51]	55.4699	0.30212	3 ($\underline{r} = [1\ \ 1\ \ 1]$)
7	Our method (MD)	0.0056	0.48178	20 ($\underline{r} = [1\ \ 2\ \ 2\ \ 15]$)
	Our method (RE)	0.0072	0.185	19 ($\underline{r} = [1\ \ 1\ \ 2\ \ 15]$)
	MECO [51]	0.0391	0.3536	16 ($\underline{r} = [4\ \ 4\ \ 4\ \ 4]$)

Table 6.: Results of the fitting of the BC trace

dc_1412_17

3

MARKOVIAN ARRIVAL PROCESSES

In many practical applications (including computer networks and telecommunication systems) the inter-arrival times of the demands are correlated. Markovian arrival processes are able to characterize such correlated point processes with Markovian tools, they serve as the arrival process in many queueing systems.

3.1 INTRODUCTION TO MARKOVIAN ARRIVAL PROCESSES

3.1.1 *Definition and basic properties*

First we introduce the rational arrival processes (RAPs), from which the MAPs are derived. Let $\mathcal{F}(t)$ be a point process with joint probability density function of inter-event times denoted by $f(x_0, x_1, \ldots, x_k)$ for $k \geq 1$.

Definition 5. *The square matrix pair of size N, $(\boldsymbol{D_0}, \boldsymbol{D_1})$, satisfying $(\boldsymbol{D_0} + \boldsymbol{D_1})\,\mathbb{1} = \underline{0}$ defines a stationary RAP iff the joint density function of the inter-arrival times*

$$f(x_1, \ldots, x_k) = \underline{\alpha} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_1} e^{\boldsymbol{D_0} x_2} \boldsymbol{D_1} \ldots e^{\boldsymbol{D_0} x_k} \boldsymbol{D_1} \mathbb{1} \tag{52}$$

is non-negative for all $k \geq 1$ and $x_1, x_2, \ldots, x_k \geq 0$ and $\underline{\alpha}$ is the unique solution of $\underline{\alpha}(-\boldsymbol{D_0})^{-1}\boldsymbol{D_1} = \underline{\alpha}$, $\underline{\alpha}\mathbb{1} = 1$.

A RAP is a point process in which the inter-arrival times are ME distributed [5, 62]. RAPs inherit several properties from ME distributions. The real parts of the eigenvalues of matrix $\boldsymbol{D_0}$ are negative; consequently the matrix is non-singular. Furthermore, the dominant eigenvalue of $\boldsymbol{D_0}$, having the maximal real part, must be real.

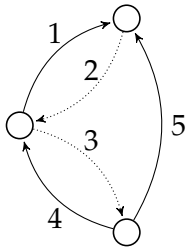
The non-negativity of the joint density function of RAPs is hard to check. Next, we introduce a Markovian subset of RAPs that have a stochastic interpretation and can be easily used in Markovian performance models.

Definition 6. *If $\mathcal{F}(t)$ is a RAP$(\boldsymbol{D_0}, \boldsymbol{D_1})$, where $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$ have the following properties:*

- $\boldsymbol{D_1}_{ij} \geq 0$,
- $\boldsymbol{D_0}_{ii} < 0$, $\boldsymbol{D_0}_{ij} \geq 0$ *for* $i \neq j$, $\boldsymbol{D_0}\mathbb{1} \leq 0$,

then we say that $\mathcal{F}(t)$ is a MAP with representation $(\boldsymbol{D_0}, \boldsymbol{D_1})$.

In case of MAPs one can interpret the off-diagonal elements of matrix $\boldsymbol{D_0}$ and the elements of $\boldsymbol{D_1}$ as transition rates corresponding to "hidden" and "visible" events, respectively. The sum of the matrices $\boldsymbol{D} = \boldsymbol{D_0} + \boldsymbol{D_1}$ is the generator of a CTMC, whose states are referred to as *phases* in this context. Whenever the CTMC traverses a transition in $\boldsymbol{D_1}$, an arrival is



$$\boldsymbol{D_0} = \begin{bmatrix} -4 & 1 & 0 \\ 0 & -2 & 0 \\ 4 & 5 & -9 \end{bmatrix}, \quad \boldsymbol{D_1} = \begin{bmatrix} 0 & 0 & 3 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 16.: An example for a MAP

generated (this is the "visible" event), while transitions in matrix $\boldsymbol{D_0}$ are not accompanied by arrivals (they are "hidden" events). As a consequence of the probabilistic interpretation the joint density function (52) of MAPs is always positive. Figure 16 depicts the Markov chain and the corresponding matrices of a MAP.

By this interpretation matrix $\boldsymbol{P} = (-\boldsymbol{D_0})^{-1}\boldsymbol{D_1}$ is the transition probability matrix of the discrete time Markov chain (DTMC) describing the phase transitions right after arrival events, and vector $\underline{\alpha}$ is its stationary distribution, hence the phase distribution at arrivals.

Consequently, the distribution of the inter-arrival times $\mathcal{F}$ is PH distributed with initial vector $\underline{\alpha}$ and transient generator $\boldsymbol{D_0}$, thus (see (1))

$$P(\mathcal{F} < t) = 1 - \underline{\alpha} e^{\boldsymbol{D_0} t} \mathbb{1}. \tag{53}$$

The marginal moments of the inter-arrival times are then (see (4))

$$m_k = E(\mathcal{F}^k) = k! \underline{\alpha} (-\boldsymbol{D_0})^{-k} \mathbb{1}, \tag{54}$$

thus the mean arrival rate is $\lambda = 1/m_1$. The mean arrival rate can be derived in an alternative way as well. If the stationary distribution of the background process is $\underline{\theta}$ (which is the unique solution to $\underline{\theta}\boldsymbol{D} = \underline{0}, \underline{\theta}\mathbb{1} = 1$), then we also have that $\lambda = \underline{\theta}\boldsymbol{D_1}\mathbb{1}$.

The joint moments of the inter-arrival times play an important role in the minimal representation of MAPs and will be used frequently in the forthcoming sections. By denoting the ℓth inter-arrival time by $\mathcal{F}_\ell$, the joint moments of the $a_0 = 0 < a_1 < a_2 < \cdots < a_k$-th inter arrival times can be derived as

$$E(\mathcal{F}_0^{i_0} \mathcal{F}_{a_1}^{i_1} \dots \mathcal{F}_{a_k}^{i_k}) = \underline{\alpha} i_0! (-\boldsymbol{D_0})^{-i_0} \boldsymbol{P}^{a_1 - a_0} i_1! (-\boldsymbol{D_0})^{-i_1} \dots \boldsymbol{P}^{a_k - a_{k-1}} i_k! (-\boldsymbol{D_0})^{-i_k} \mathbb{1}. \tag{55}$$

Throughout this dissertation the lag-1 joint moments appear frequently, therefore we introduce a shorter notation here as

$$\eta_{ij} = E(\mathcal{F}_0^i \mathcal{F}_1^j) = \underline{\alpha} i! (-\boldsymbol{D_0})^{-i} \boldsymbol{P} j! (-\boldsymbol{D_0})^{-j} \mathbb{1}. \tag{56}$$

Several statistical quantities can be used in the practice to characterize the dependency structure of the inter-arrival times generated by MAPs. One of the most popular ones is the lag-k auto-correlation, ρ_k, which is the correlation between $\mathcal{F}_0$ and $\mathcal{F}_k$. It can be expressed from the lag-k joint moment as

$$\rho_k = \frac{E(\mathcal{F}_0 \mathcal{F}_k) - m_1^2}{m_2 - m_1^2}. \tag{57}$$

MAPs have the following appealing features that make them suitable to model the internal traffic of queueing networks:

- The departure process of the waist majority of queues is correlated (apart from the simplest cases), and MAPs are able to capture correlations in a Markovian way.
- The superposition of two MAPs is a MAP as well. If the two MAPs to superpose are represented by matrices $(\boldsymbol{D_0}^{(1)}, \boldsymbol{D_1}^{(1)})$ and $(\boldsymbol{D_0}^{(2)}, \boldsymbol{D_1}^{(2)})$, then the matrices representing the superposed process are

$$\begin{aligned} \boldsymbol{D_0}^{(superposed)} &= \boldsymbol{D_0}^{(1)} \oplus \boldsymbol{D_0}^{(2)}, \\ \boldsymbol{D_1}^{(superposed)} &= \boldsymbol{D_1}^{(1)} \oplus \boldsymbol{D_1}^{(2)}. \end{aligned} \tag{58}$$

  (For the definition and properties of the Kronecker summation operator $\oplus$ see Appendix A.1).
- The probabilistic splitting of a MAP is also a MAP. If the departing jobs are directed to a given consecutive node with probability p, then this traffic is represented by matrices

$$\begin{aligned} \boldsymbol{D_0}^{(split)} &= \boldsymbol{D_0} + (1-p)\boldsymbol{D_1}, \\ \boldsymbol{D_1}^{(split)} &= p\boldsymbol{D_1}. \end{aligned} \tag{59}$$

MAPs have some distinguished special sub-classes, which are used frequently in the practice.

- *PH renewal processes* are point processes where the inter-arrival times are independent and identically distributed (iid.). If the vector-matrix representation of the PH distribution is denoted by $(\underline{\sigma}, \boldsymbol{S})$, then the matrices of the corresponding MAP are

$$\begin{aligned} \boldsymbol{D_0} &= \boldsymbol{S}, \\ \boldsymbol{D_1} &= (-\boldsymbol{S})\mathbb{1} \cdot \underline{\sigma}, \end{aligned} \tag{60}$$

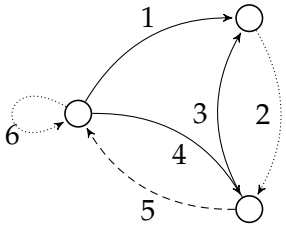
  which means that transitions to the absorbing state are accompanied by an arrival and at the same time the PH is re-initialized according to probability distribution $\underline{\sigma}$.
- *Markov-modulated Poisson processes* are arrival processes that have a CTMC background process with generator matrix $\boldsymbol{Q}$, and arrivals are generated according to a Poisson process with state-dependent rates. If the vector of the arrival rates is $\underline{r} = \{r_i, i = 1, \ldots, N\}$, then the matrix parameters of the MAP are

$$\begin{aligned} \boldsymbol{D_0} &= \boldsymbol{Q} - \mathrm{diag}\langle \underline{r} \rangle, \\ \boldsymbol{D_1} &= \mathrm{diag}\langle \underline{r} \rangle. \end{aligned} \tag{61}$$

3.1.2 *Marked Markovian arrival processes*

A MAP defines a point process where there is no difference between the arrival events. In many practical applications, however, several *types* (or classes) of arrivals can be distinguished. E.g., some arrival types need longer, some others need shorter service, or some arrival types need more urgent service than others.

MMAPs are the multi-type extensions of MAPs ([39]). Let us start the discussion with a more general model class, the multi-type extension of RAPs, the marked rational arrival processes (MRAPs) ([9]).



$$\boldsymbol{D_0} = \begin{bmatrix} -11 & 1 & 4 \\ 0 & -2 & 0 \\ 0 & 3 & -8 \end{bmatrix}, \boldsymbol{D_1} = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{D_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 0 & 0 \end{bmatrix}$$

Figure 17.: An example for a MMAP

Definition 7. *A set of square matrices of size N, $(\boldsymbol{D_0}, \boldsymbol{D_1}, \ldots, \boldsymbol{D_K})$, satisfying $\sum_{k=0}^{K} \boldsymbol{D_k}\,\mathbb{1} = \underline{0}$, defines a stationary MRAP with K event types iff the joint density function of the arrival sequence (consecutive interarrival times and event types)*

$$f(x_1, k_1, \ldots, x_n, k_n) = \underline{\alpha} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{k_1}} e^{\boldsymbol{D_0} x_2} \boldsymbol{D_{k_2}} \ldots e^{\boldsymbol{D_0} x_n} \boldsymbol{D_{k_n}} \mathbb{1} \tag{62}$$

is non-negative for all $j \geq 1$ and $x_1, x_2, \ldots, x_n \geq 0, 1 \leq k_1, k_2, \ldots, k_n \leq K$ and $\underline{\alpha}$ is the unique solution of $\underline{\alpha}(-\boldsymbol{D_0})^{-1} \sum_{k=1}^{K} \boldsymbol{D_k} = \underline{\alpha}$, $\underline{\alpha}\mathbb{1} = 1$.

The following definition introduces the Markovian sub-class of MRAPs, similar to the single-type case.

Definition 8. *If for the matrices of a MRAP $\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}$ we have that*

- $\boldsymbol{D_k}_{ij} \geq 0$, *for* $k = 1, \ldots, K$,
- $\boldsymbol{D_0}_{ii} < 0$, $\boldsymbol{D_0}_{ij} \geq 0$ *for* $i \neq j$, $\boldsymbol{D_0}\mathbb{1} \leq 0$,

then we say that this MRAP is a MMAP with representation $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$.

Obviously, the class of MRAPs contains MMAPs.

In the single-type case MAPs were interpreted as Markov chains with two kinds of transitions: transitions generating arrivals and transitions not accompanied by arrivals. The stochastic interpretation of the multi-type variant is similar: there is a CTMC modulating the arrivals with generator $\boldsymbol{D} = \sum_{k=1}^{K} \boldsymbol{D_k}$ (which is assumed to be irreducible), where the transitions are marked. Transitions in $\boldsymbol{D_0}$ are just internal transitions, while transitions in matrix $\boldsymbol{D_k}$ are accompanied by type-k arrivals. In the example in Figure 17 the transitions leading to type-1 and type-2 arrivals are denoted by dotted and dashed lines, respectively.

The transition probability matrix of the DTMC embedded to arrival instants (of any type) is $\boldsymbol{P} = (-\boldsymbol{D_0})^{-1} \sum_{k=1}^{K} \boldsymbol{D_k}$, and its stationary probability vector is $\underline{\alpha}$. Hence, like in the single-class case, the steady state distribution of the inter-arrival times $\mathcal{F}$ is PH distributed with initial vector $\underline{\alpha}$ and transient generator $\boldsymbol{D_0}$, see (53), and the marginal moments are computed according to (54).

If the stationary phase distribution of the background process $\boldsymbol{D}$ is vector $\underline{\theta}$, the arrival intensity of type-k customers is given by $\lambda_k = \underline{\theta}\boldsymbol{D_k}\mathbb{1}$. The total arrival rate, $\lambda = \sum_{k=1}^{K} \lambda_k$, can also be obtained as the inverse of the mean of the inter-arrival times $\lambda = 1/m_1$.

Let $\mathcal{F}_i^{(k)}$ be $\mathcal{F}_i$ (the inter-arrival time between the ith and the $i+1$th arrival) if the $i+1$th arrival is of class k and 0 otherwise. Then, the joint moments of the joint distribution (62), playing a central role in the queuing network analysis approach proposed in this dissertation, are

$$\begin{aligned} &E\left(\left(\mathcal{F}_0^{(k_0)}\right)^{i_0}\left(\mathcal{F}_1^{(k_1)}\right)^{i_1}\ldots\left(\mathcal{F}_n^{(k_n)}\right)^{i_n}\right) = \\ &\underline{\alpha}\; i_0!(-\boldsymbol{D_0})^{-i_0-1}\boldsymbol{D_{k_0}}\; i_1!(-\boldsymbol{D_0})^{-i_1-1}\boldsymbol{D_{k_1}}\ldots i_n!(-\boldsymbol{D_0})^{-i_n-1}\boldsymbol{D_{k_n}}\mathbb{1}. \end{aligned} \tag{63}$$

Particularly, the lag-1 joint moments of two consecutive arrivals, denoted by $\eta_{i,j}^{(k)}$ will be used frequently in the sequel:

$$\eta_{i,j}^{(k)} = E\left((\mathcal{F}_0^{(k)})^i(\mathcal{F}_1)^j\right) = i!j!\,\underline{\alpha}(-\boldsymbol{D_0})^{-i-1}\boldsymbol{D_k}(-\boldsymbol{D_0})^{-j}\mathbb{1}, \quad i > 0, j \geq 0, \tag{64}$$

and for $i = 0$ we define $\eta_{0,j}^{(k)}$ as

$$\eta_{0,j}^{(k)} = Pr(\mathcal{F}_0^{(k)} > 0)E\left((\mathcal{F}_1)^j\right) = j!\,\underline{\alpha}(-\boldsymbol{D_0})^{-1}\boldsymbol{D_k}(-\boldsymbol{D_0})^{-j}\mathbb{1}, \quad j \geq 0. \tag{65}$$

Like the single-type MAPs, MMAPs are also closed for the superposition and random splitting operation. Superposing MMAPs $(\boldsymbol{D_0}^{(1)}, \ldots, \boldsymbol{D_K}^{(1)})$ and $(\boldsymbol{D_0}^{(2)}, \ldots, \boldsymbol{D_K}^{(2)})$ the result is also a MMAP with representation

$$\boldsymbol{D_k}^{(superposed)} = \boldsymbol{D_k}^{(1)} \oplus \boldsymbol{D_k}^{(2)}, \quad \text{for } k = 0, \ldots, K. \tag{66}$$

Similarly, if the type-k arrivals of a MMAP characterized by $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ are directed to a given direction with probability p_k, the MMAP describing the traffic is

$$\begin{aligned} \boldsymbol{D_0}^{(split)} &= \boldsymbol{D_0} + \sum_{k=1}^{K}(1-p_k)\boldsymbol{D_k}, \\ \boldsymbol{D_k}^{(split)} &= p_k\boldsymbol{D_k}, \quad \text{for } k = 1, \ldots, K. \end{aligned} \tag{67}$$

3.1.3 *Representation transformation*

Section 2.1.2 showed that the vector-matrix representation of ME (and also PH) distributions is not unique. The same holds for the $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ representation of MMAPs as well. The joint distribution defined by (62) can be transformed with any non-singular square matrix $\boldsymbol{B}$ satisfying $\boldsymbol{B}\mathbb{1} = \mathbb{1}$ as

$$\begin{aligned} &f(x_1, k_1, \ldots, x_n, k_n) = \underline{\alpha}e^{\boldsymbol{D_0}x_1}\boldsymbol{D_{k_1}}e^{\boldsymbol{D_0}x_2}\boldsymbol{D_{k_2}}\ldots e^{\boldsymbol{D_0}x_n}\boldsymbol{D_{k_n}}\mathbb{1} \\ &= \underline{\alpha}\boldsymbol{B}e^{\boldsymbol{B}^{-1}\boldsymbol{D_0}\boldsymbol{B}x_1}\boldsymbol{B}^{-1}\boldsymbol{D_{k_1}}\boldsymbol{B}e^{\boldsymbol{B}^{-1}\boldsymbol{D_0}\boldsymbol{B}x_2}\boldsymbol{B}^{-1}\boldsymbol{D_{k_2}}\boldsymbol{B}\ldots\boldsymbol{B}e^{\boldsymbol{B}^{-1}\boldsymbol{D_0}\boldsymbol{B}x_n}\boldsymbol{B}^{-1}\boldsymbol{D_{k_n}}\boldsymbol{B}\boldsymbol{B}^{-1}\mathbb{1} \\ &= \underline{\gamma}e^{\boldsymbol{G_0}x_1}\boldsymbol{G_{k_1}}e^{\boldsymbol{G_0}x_2}\boldsymbol{G_{k_2}}\ldots e^{\boldsymbol{G_0}x_n}\boldsymbol{G_{k_n}}\mathbb{1}, \end{aligned} \tag{68}$$

which means that the MMAPs given by representations $(\boldsymbol{D_k}, k = 0, \ldots, K)$ and $(\boldsymbol{G_k} = \boldsymbol{B}^{-1}\boldsymbol{D_k}\boldsymbol{B}, k = 0, \ldots, K)$ are the same, even though the representations are different.

Similarity transformations can be extended to matrix representations of different sizes [21] as well. We recall the possible similarity transformations without proof (the proofs are similar to the ones of Theorems 2 and 3).

Theorem 9 ([21],Theorem 1). *If there is a matrix $\boldsymbol{V} \in \mathbb{R}^{N,M}$, $M \geq N$ such that $\mathbb{1} = \boldsymbol{V}\mathbb{1}$ and $\boldsymbol{D_k}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{G_k}$ for $k = 0, \ldots, K$ then $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ and $(\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})$ define the same MRAP.*

Theorem 10 ([21],Theorem 2). *If there is a matrix $\boldsymbol{W} \in \mathbb{R}^{M,N}$, $M \geq N$ such that $\mathbb{1} = \boldsymbol{W}\mathbb{1}$ and $\boldsymbol{W}\boldsymbol{D_k} = \boldsymbol{G_k}\boldsymbol{W}$ for $k = 0, \ldots, K$ then $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ and $(\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})$ define the same MRAP.*

Like in case of PH distributions, the existence of multiple representations defining the same process makes many analytical investigations and also the development of matching/fitting procedures relatively hard.

3.2 MINIMAL CHARACTERIZATION OF MMAPS AND A MOMENT MATCHING METHOD

As shown in Section 3.1.3, the traditional representation of MMAPs, given by matrices $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$, hence $(K+1)N^2$ parameters, is redundant, there are infinitely many matrix sets defining the exactly same stochastic process. This section addresses two related questions:

- If $(K+1)N^2$ parameters are two much, what is the minimal number of parameters that determine MMAPs uniquely?
- What exactly are the parameters that determine MMAPs uniquely?

3.2.1 *Minimal characterization of single-type RAPs*

For technical simplicity, let us define the double transform of the number of arrivals in the $(0,t)$ interval starting from an arrival at time 0. If $\mathcal{N}(t)$ is the number of arrivals generated up to time t, from [57] we have that the double transform of $\mathcal{N}(t)$ is

$$f(s,z) = \int_{t=0}^{\infty} e^{-st} E(z^{\mathcal{N}(t)}) dt = \underline{\alpha}(s\boldsymbol{I} - \boldsymbol{D_0} - z\boldsymbol{D_1})^{-1}\mathbb{1}. \tag{69}$$

The next definition introduces the *non-redundant* property of RAPs, that will be assumed to hold in all forthcoming results of this section.

Definition 9. *RAP($\boldsymbol{D_0}, \boldsymbol{D_1}$) is* non-redundant *if its rank equals to its order, where the* rank *is the size the square matrices $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$, and the* order *is the degree of the denominator of $f(s,z)$ as a polynomial of s.*

In the following theorem we prove that the joint moments, defined by (55), determine a RAP completely.

Theorem 11. *If the joint moments of the $a_0 = 0 < a_1 < a_2 < \ldots < a_k$-th inter-arrival times of RAP($\boldsymbol{D_0}, \boldsymbol{D_1}$) and RAP($\boldsymbol{D_0'}, \boldsymbol{D_1'}$) are identical for all $k \geq 0$; $i_0, \ldots, i_k$ and $a_1, \ldots, a_k$ then $f(s,z) \equiv f'(s,z)$.*

Proof. In the convergence region of $f(s,z)$ we have

$$\begin{aligned} f(s,z) &= \underline{\alpha}(s\boldsymbol{I} - \boldsymbol{D_0} - z\boldsymbol{D_1})^{-1}\mathbb{1} = \underline{\alpha}\left(s(-\boldsymbol{D_0})^{-1} + \boldsymbol{I} - z\boldsymbol{P}\right)^{-1}(-\boldsymbol{D_0})^{-1}\mathbb{1} \\ &= \sum_{i=0}^{\infty} \underline{\alpha}\left(-s(-\boldsymbol{D_0})^{-1} + z\boldsymbol{P}\right)^{i}(-\boldsymbol{D_0})^{-1}\mathbb{1}. \end{aligned} \tag{70}$$

The ith term of the above sum, $\underline{\alpha}(-s(-\boldsymbol{D_0})^{-1} + z\boldsymbol{P})^i(-\boldsymbol{D_0})^{-1}\mathbb{1}$, is composed by the permutations of the $(-\boldsymbol{D_0})^{-1}$ and the $\boldsymbol{P}$ matrices. The permutations that start with $\underline{\alpha}\boldsymbol{P}^j$ can be simplified to the

$$\underline{\alpha}(-\boldsymbol{D_0})^{-i_0}\boldsymbol{P}^{j_0}(-\boldsymbol{D_0})^{-i_1}\ldots\boldsymbol{P}^{j_{k-1}}(-\boldsymbol{D_0})^{-i_k}\mathbb{1} \tag{71}$$

form, since $\underline{\alpha} = \underline{\alpha}\boldsymbol{P}$. Indeed, (71) is $E(\mathcal{F}_0^{i_0}\mathcal{F}_{a_1}^{i_1}\ldots\mathcal{F}_{a_k}^{i_k})/(i_0!i_1!\ldots i_k!)$, where $a_k = \sum_{\ell=0}^{k-1} j_\ell$. Due to the equality of the joint moments of RAP($\boldsymbol{D_0}, \boldsymbol{D_1}$) and RAP($\boldsymbol{D_0'}, \boldsymbol{D_1'}$) all terms of the (70) composition of $f(s,z)$ and $f'(s,z)$ are identical, which implies the theorem. □

The main theorem below provides the minimal number of parameters characterizing a RAP.

Theorem 12. *The distribution of an order-N non-redundant irreducible RAP is determined by at most N^2 independent parameters.*

Proof. To prove the theorem we provide a description of all joint moments based on N^2 parameters and Theorem 11 ensures that this description also defines the distribution.

Let $-\boldsymbol{D_0}^{-1} = \boldsymbol{\Gamma}^{-1}\boldsymbol{E}\boldsymbol{\Gamma}$ be the Jordan decomposition of $-\boldsymbol{D_0}^{-1}$ normalized such that $\boldsymbol{\Gamma}\mathbb{1} = \mathbb{1}$ and $\boldsymbol{R} = \boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{\Gamma}^{-1}$. The $\boldsymbol{E}$ matrix has the Jordan-block structure $\boldsymbol{E} = \mathrm{diag}\langle \boldsymbol{E}_j \rangle$ and $\boldsymbol{R}$ satisfies $\boldsymbol{R}\mathbb{1} = \mathbb{1}$ since $\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{\Gamma}^{-1}\mathbb{1} = \boldsymbol{\Gamma}\boldsymbol{P}\mathbb{1} = \boldsymbol{\Gamma}\mathbb{1} = \mathbb{1}$.

Using these notations the joint moments can be written as

$$\begin{aligned} & E(\mathcal{F}_{a_0}^{i_0}\mathcal{F}_{a_1}^{i_1}\dots\mathcal{F}_{a_k}^{i_k})/(i_0!i_1!\dots i_k!) \\ &= \underline{\alpha}(-\boldsymbol{D_0})^{-i_0}\boldsymbol{P}^{a_1-a_0}(-\boldsymbol{D_0})^{-i_1}\dots\boldsymbol{P}^{a_k-a_{k-1}}(-\boldsymbol{D_0})^{-i_k}\mathbb{1} \\ &= \underline{\alpha}\,\boldsymbol{\Gamma}^{-1}\boldsymbol{E}^{i_0}\boldsymbol{\Gamma}\,\boldsymbol{P}^{a_1-a_0}\,\boldsymbol{\Gamma}^{-1}\boldsymbol{E}^{i_1}\boldsymbol{\Gamma}\dots\boldsymbol{P}^{a_k-a_{k-1}}\,\boldsymbol{\Gamma}^{-1}\boldsymbol{E}^{i_k}\boldsymbol{\Gamma}\,\mathbb{1} \\ &= \underline{v}\boldsymbol{E}^{i_0}\,\boldsymbol{R}^{a_1-a_0}\,\boldsymbol{E}^{i_1}\dots\boldsymbol{R}^{a_k-a_{k-1}}\,\boldsymbol{E}^{i_k}\mathbb{1}. \end{aligned} \tag{72}$$

where $\underline{v} = \underline{\alpha}\boldsymbol{\Gamma}^{-1}$. Vector $\underline{v}$ is determined by $\boldsymbol{R}$ because $\underline{v}\boldsymbol{R} = \underline{v}$ and $\underline{v}\mathbb{1} = 1$, since

$$\underline{v} = \underline{\alpha}\boldsymbol{\Gamma}^{-1} = \underline{\alpha}\boldsymbol{P}\boldsymbol{\Gamma}^{-1} = \underline{\alpha}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}\boldsymbol{P}\boldsymbol{\Gamma}^{-1} = \underline{v}\boldsymbol{R}, \tag{73}$$

and

$$\underline{v}\mathbb{1} = \underline{\alpha}\boldsymbol{\Gamma}^{-1}\mathbb{1} = \underline{\alpha}\mathbb{1} = \mathbb{1}. \tag{74}$$

Based on (72) any joint moment is determined by $\boldsymbol{E}$ and $\boldsymbol{R}$. Matrix $\boldsymbol{E}$ is determined by the N (potentially partially coinciding, potentially complex) eigenvalues of $(-\boldsymbol{D_0})^{-1}$. Matrix $\boldsymbol{R}$ is determined by its $N(N-1)$ (potentially complex) elements, since $\boldsymbol{R}\mathbb{1} = \mathbb{1}$. All together these give N^2 parameters. □

3.2.2 *A moment matching method*

Theorem 12 has answered the first question arisen at the top of the section, thus N^2 parameters uniquely determine a RAP. Next, a moment matching method is provided that actually creates a RAP based on N^2 moment and joint moment related parameters.

This procedure consists of two steps: matrix $\boldsymbol{D_0}$ is created in the first, matrix $\boldsymbol{D_1}$ in the second step.

Observe that the distribution of the inter-arrival times (see (53)) depends solely on matrix $\boldsymbol{D_0}$. Consequently, this matrix is constructed from the marginal moments, while the joint moments characterizing the correlations are ignored in this step. Hence, first we create a PH distribution based on $2N-1$ marginal moments of the inter-arrival time, $m_1, \dots, m_{2N-1}$. The algorithm presented in [82] (also used in Section 2.3.1) returns a vector-matrix pair $(\underline{\sigma}, \boldsymbol{S})$ such that $m_n = n!\underline{\sigma}(-\boldsymbol{S})^{-n}\mathbb{1}$ holds for $n = 1, \dots, 2N-1$. Matrix $\boldsymbol{E}$ of Theorem 12, containing the eigenvalues of matrix $(-\boldsymbol{D_0})^{-1}$, is given by $\boldsymbol{E} = (-\boldsymbol{S})^{-1}$.

The second step of the procedure is to obtain matrix $\boldsymbol{R}$ from the joint moments $\eta_{ij}, i, j = 1, \dots, N-1$. (Note that $\eta_{i0} = \eta_{0i} = m_i$.) Based on the $(\underline{\sigma}, \boldsymbol{S})$ representation of the inter-arrival times, the $m_i = E(\mathcal{F}_0^i), i = 1, \dots, N-1$ moments and the $\eta_{ij} = E(\mathcal{F}_0^i\mathcal{F}_1^j), i, j = 1, \dots, N-1$ joint moments we compose 3 matrices of size $N \times N$. Matrix $\boldsymbol{N}$ contains the

moments such that $N_{ij} = \eta_{i-1,j-1}$, matrix $\boldsymbol{\Lambda}_\sigma$ and $\boldsymbol{\Lambda}_\mathbb{1}$ are such that the ith row of $\boldsymbol{\Lambda}_\sigma$ is $\underline{\sigma}(i-1)!\boldsymbol{S}^{-(i-1)}$ and the jth column of $\boldsymbol{\Lambda}_\mathbb{1}$ is $(j-1)!\boldsymbol{S}^{-(j-1)}\mathbb{1}$. That is

$$\boldsymbol{N} = \begin{bmatrix} 1 & m_1 & m_2 & \dots \\ m_1 & \eta_{1,1} & \eta_{1,2} & \dots \\ m_2 & \eta_{2,1} & \eta_{2,2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \boldsymbol{\Lambda}_\sigma = \left[\begin{array}{c} \underline{\sigma} \\ \hline \underline{\sigma}\boldsymbol{E} \\ \hline 2!\underline{\sigma}\boldsymbol{E}^2 \\ \hline \vdots \end{array}\right], \boldsymbol{\Lambda}_\mathbb{1} = \left[\begin{array}{c|c|c|c} \mathbb{1} & \boldsymbol{E}\mathbb{1} & 2!\boldsymbol{E}^2\mathbb{1} & \dots \end{array}\right].$$

Observe that with these matrices we have that $\boldsymbol{N} = \boldsymbol{\Lambda}_\sigma \boldsymbol{R} \boldsymbol{\Lambda}_\mathbb{1}$, since

$$\boldsymbol{N}_{ij} = \eta_{i-1,j-1} = E(\mathcal{F}_0^{i-1}\mathcal{F}_1^{j-1}) = \underline{\sigma}(i-1)!\boldsymbol{E}^{i-1}\ \boldsymbol{R}\ (j-1)!\boldsymbol{E}^{j-1}\mathbb{1}. \tag{75}$$

Thus, matrix $\boldsymbol{R}$ is obtained as $\boldsymbol{R} = \boldsymbol{\Lambda}_\sigma^{-1}\boldsymbol{N}\boldsymbol{\Lambda}_\mathbb{1}^{-1}$.

Now we have matrix $\boldsymbol{E}$ created from $2N-1$ marginal moments, and matrix $\boldsymbol{R}$ created from $(N-1)^2$ additional joint moments, hence these two matrices were constructed from $(N-1)^2 + 2N - 1 = N^2$, that is, minimal number of parameters. It remains to obtain the traditional $(\boldsymbol{D_0}, \boldsymbol{D_1})$ representation of the RAP from $\boldsymbol{E}$ and $\boldsymbol{R}$.

Matrices $\boldsymbol{E}$ and $\boldsymbol{R}$ have been defined in the proof of Theorem 12 such that they are similar (in the sense of a similarity transformation) to matrices $(-\boldsymbol{D_0})^{-1}$ and $\boldsymbol{P} = (-\boldsymbol{D_0})^{-1}\boldsymbol{D_1}$, respectively. This means that matrices $\boldsymbol{D_0} = (-\boldsymbol{E})^{-1} = \boldsymbol{S}$ and $\boldsymbol{D_1} = -\boldsymbol{S}\boldsymbol{R}$ form a proper (possibly non-Markovian) representation of the RAP having the target marginal and joint moments.

3.2.3 *Extension to the multi-type case*

In this section we will show that an appropriately chosen set of marginal and joint moments provide a unique representation of a MRAP and present a method to obtain a MRAP from a set of joint moments.

Theorem 13. *Consider a non-redundant MRAP of order N whose moments and joint moments are m_i and $\eta_{i,j}^{(k)}$ $(\forall i, j \geq 0, k = 1, \dots, K)$. If a vector $\underline{\sigma}$ and a matrix $\boldsymbol{S}$ of order N are such that $m_i = i!\,\underline{\sigma}(-\boldsymbol{S})^i\mathbb{1}, \quad \forall i \geq 0$ then*

$$\boldsymbol{D_0} = \boldsymbol{S}, \quad \boldsymbol{D_k} = -\boldsymbol{D_0}\boldsymbol{\Lambda}_\sigma^{-1}\boldsymbol{N_k}\boldsymbol{\Lambda}_\mathbb{1}^{-1}, \ \ 1 \leq k \leq K, \tag{76}$$

is a representation of the MRAP where

$$\boldsymbol{N_k} = \begin{bmatrix} \eta_{0,0}^{(k)} & \eta_{0,1}^{(k)} & \cdots & \eta_{0,N-1}^{(k)} \\ \eta_{1,0}^{(k)} & \eta_{1,1}^{(k)} & \cdots & \eta_{1,N-1}^{(k)} \\ \vdots & \vdots & & \vdots \\ \eta_{N-1,0}^{(k)} & \eta_{N-1,1}^{(k)} & \cdots & \eta_{N-1,N-1}^{(k)} \end{bmatrix}, \tag{77}$$

$$\boldsymbol{\Lambda}_\sigma = \left[\begin{array}{c} \underline{\sigma} \\ \hline \underline{\sigma}\boldsymbol{E} \\ \hline \vdots \\ \hline (N-1)!\underline{\sigma}\boldsymbol{E}^{N-1} \end{array}\right], \tag{78}$$

$$\mathbf{\Lambda}_{\mathbb{1}} = \left[\begin{array}{c|c|c|c} \mathbb{1} & \boldsymbol{E}\mathbb{1} & \dots & (N\text{–}1)!\boldsymbol{E}^{N-1}\mathbb{1} \end{array}\right], \tag{79}$$

with $\boldsymbol{E} = (-\boldsymbol{S})^{-1}$.

Proof. The following is a direct consequence of results of Theorem 12 for RAPs: for a non-redundant MRAP

- $\mathbf{\Lambda}_\sigma$ and $\mathbf{\Lambda}_{\mathbb{1}}$ are non-singular;
- the first $2N-1$ moments of the inter-arrival time completely determine its distribution;
- the first joint moments of 2 consecutive inter-arrival intervals, in particular, $\eta_{i,j}^{(k)}$, $i,j = 0,\dots,n-1, 1 \le k \le K$ define the whole process.

The vector $\underline{\sigma}$ and the matrix $\boldsymbol{S}$ is a non-redundant matrix exponential representation of the inter-arrival time distribution, i.e., $\underline{\sigma}\, e^{\boldsymbol{S}x}(-\boldsymbol{S})\mathbb{1} = f(x)$, and can be computed by the algorithm presented in [82].

It remains to show that the joint moments of the MRAP with representation $\boldsymbol{D_0}$, $\boldsymbol{D_k}$ ($k = 1,\dots,K$) are $\eta_{i,j}^{(k)}$, $i,j = 0,\dots,N-1, 1 \le k \le K$. The lag-1 joint moments of the MRAP given by $\boldsymbol{D_k}, k = 0,\dots,K$ are

$$\vartheta_{i,j}^{(k)} = i!j!\,\underline{\sigma}(-\boldsymbol{D_0})^{-i-1}\boldsymbol{D_k}(-\boldsymbol{D_0})^{-j}\mathbb{1} = \underbrace{i!\,\underline{\sigma}\boldsymbol{E}^i}_{\text{row of } \mathbf{\Lambda}_\sigma}\ \mathbf{\Lambda}_\sigma^{-1}\boldsymbol{N_k}\mathbf{\Lambda}_{\mathbb{1}}^{-1}\ \underbrace{j!\,\boldsymbol{E}^j\mathbb{1}}_{\text{column of } \mathbf{\Lambda}_{\mathbb{1}}}. \tag{80}$$

Matrix $\mathbf{\Theta_k}$ is defined such that its i,j element is $\vartheta_{i,j}^{(k)}$. Based on (80) we have

$$\mathbf{\Theta_k} = \mathbf{\Lambda}_\sigma\mathbf{\Lambda}_\sigma^{-1}\boldsymbol{N_k}\mathbf{\Lambda}_{\mathbb{1}}^{-1}\mathbf{\Lambda}_{\mathbb{1}} = \boldsymbol{N_k} \tag{81}$$

which implies that $\vartheta_{i,j}^{(k)} = \eta_{i,j}^{(k)}$ for $i,j = 0,\dots,N-1, 1 \le k \le K$. □

Theorem 13 makes it possible to construct a representation for a MRAP when its moments and joint moments are known.

An important consequence of Theorem 13 is that the number of parameters to define a non-redundant MRAP of order N is KN^2. The first $2N-1$ (marginal) moments of $\mathcal{F}_0$ define the distribution of the inter-arrival times ($\underline{\sigma}$ and $\boldsymbol{S}$) and the matrices of the joint moments $\boldsymbol{N_k}$ are given by their KN^2 elements. All together there are $KN^2 + 2N - 1$ parameters but they are not independent. For $i = 0,\dots,N-1$ we have

$$\begin{aligned} m_i &= i!\underline{\alpha}(-\boldsymbol{D_0})^{-i}\mathbb{1} = i!\underline{\alpha}(-\boldsymbol{D_0})^{-i-1}(-\boldsymbol{D_0})\mathbb{1} \\ &= i!\underline{\alpha}(-\boldsymbol{D_0})^{-i-1}\sum_{k=1}^{K}\boldsymbol{D_k}\mathbb{1} = \sum_{k=1}^{K}\eta_{i,0}^{(k)}, \end{aligned}$$

where we utilized that the row sum of $\boldsymbol{D}$ is zero. Similarly, for $j = 0,\dots,N-1$ we have

$$m_j = E\left(\mathcal{F}_1^j\right) = \sum_{k=1}^{K} Pr(\mathcal{F}_0^{(k)} > 0)E\left(\mathcal{F}_1^j\right) = \sum_{k=1}^{K}\eta_{0,j}^{(k)}.$$

These two sets of equations result in $2N-1$ additional linear relations among the moments and the joint moments reducing the number of independent parameters to NK^2.

3.2.4 *Obtaining Markovian representation with successive transformations*

Section 3.2.3 presents a method to create a MRAP from KN^2 parameters, from which $2N-1$ are marginal moments and the remaining ones are joint moments. The result of this method, however, is typically non-Markovian, and is not even necessarily a valid MRAP, there is no guarantee that the joint density function (62) is non-negative. There are no ways to check the non-negativity of *all* finite dimensional joint density functions. The only possibility to ensure that the result corresponds to a valid arrival process is trying to transform it to a Markovian representation (to a MMAP), which, by stochastic interpretation, always defines a valid arrival process.

In order to transform a non-Markovian representation $(\boldsymbol{D_0}, \dots, \boldsymbol{D_K})$ to a Markovian one $(\boldsymbol{G_0}, \dots, \boldsymbol{G_K})$ we need to find an appropriate non-singular transformation matrix $\boldsymbol{B}$, for which $\boldsymbol{B}\mathbb{1} = \mathbb{1}$ and $\boldsymbol{G_k} = \boldsymbol{B}^{-1}\boldsymbol{D_k}\boldsymbol{B}, k = 0, \dots, K$ holds.

Hence, the transformation matrix has to be such that the elements of $\boldsymbol{B}^{-1}\boldsymbol{D_k}\boldsymbol{B}, k = 1, \dots, K$, and the off-diagonal elements of $\boldsymbol{B}^{-1}\boldsymbol{D_0}\boldsymbol{B}$ are non-negative. The non-positive constraint on the diagonal elements of $\boldsymbol{B}^{-1}\boldsymbol{D_0}\boldsymbol{B}$ is automatically fulfilled in this case, since for a valid MAP $\sum_{k=0}^{K} \boldsymbol{D_k}\mathbb{1} = 0$ and from $\boldsymbol{B}\mathbb{1} = \mathbb{1}$ we have $\sum_{k=0}^{K} \boldsymbol{G_k}\mathbb{1} = \boldsymbol{B}^{-1}\sum_{k=0}^{K} \boldsymbol{D_k}\boldsymbol{B}\mathbb{1} = 0$.

We apply an iterative numerical optimization method to find such matrix $\boldsymbol{B}$. The cost function is defined to penalize the negative elements, hence we are looking for a representation that minimizes

$$\mathcal{E}(\boldsymbol{D_0}, \dots, \boldsymbol{D_K}) = -\sum_{i,j,i\neq j} \min\{0, \boldsymbol{D_0}_{i,j}\} - \sum_{k=1}^{K}\sum_{i,j} \min\{0, \boldsymbol{D_k}_{i,j}\}. \tag{82}$$

The minimization consists of the successive application of *elementary similarity transformations*. Elementary transformations are defined by matrix

$$\boldsymbol{B}_{i,j}(b) = \begin{matrix} \\ \\ \\ i \\ \\ \end{matrix} \begin{matrix} \begin{matrix} \quad & \quad & \quad & \quad & \quad \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \dots \\ 0 & b & 0 & 1-b & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \begin{matrix} \quad & \quad & \quad & \quad & \quad \end{matrix} \end{matrix}, \tag{83}$$

$$\begin{matrix} \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad \\ \quad & j & \quad & \quad & \quad \end{matrix}$$

and have three parameters: b is the step size controlling the coarseness of the transformation and parameters i, j determine which elements of the representation are affected by the transformation.

In each step of the optimization similarity transformations are applied to the current representation. The i, j parameters are selected to minimize the object function $\mathcal{E}$. A large step size b results in a faster convergence, but the algorithm may stop quickly without finding a Markovian solution. Hence, b parameter is decreased gradually in subsequent steps of the optimization.

The details of the procedure are described by Algorithm 3. This algorithm usually finds a solution if the input has a Markovian representation, and never finds a solution if it does not.

Algorithm 3 The algorithm to transform a non-Markovian representation to a Markovian one

procedure TRANSFORM-TO-MARKOVIAN($\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}$)
  $\boldsymbol{G_k} \leftarrow \boldsymbol{D_k}$, for $k = 0, \ldots, K$
  $b \leftarrow 0.5$
  while $b >$ *precision* **do**
    repeat
      $i^*, j^* \leftarrow \arg\min_{i,j} \mathcal{E}(\boldsymbol{B}_{i,j}(b)^{-1}\boldsymbol{G_0}\boldsymbol{B}_{i,j}(b), \ldots, \boldsymbol{B}_{i,j}(b)^{-1}\boldsymbol{G_K}\boldsymbol{B}_{i,j}(b))$
      $\boldsymbol{G_k} \leftarrow \boldsymbol{B}_{i^*,j^*}(b)^{-1}\boldsymbol{G_k}\boldsymbol{B}_{i^*,j^*}(b)$, for $k = 0, \ldots, K$
      $i^*, j^* \leftarrow \arg\min_{i,j} \mathcal{E}(\boldsymbol{B}_{i,j}(-b)^{-1}\boldsymbol{G_0}\boldsymbol{B}_{i,j}(-b), \ldots, \boldsymbol{B}_{i,j}(-b)^{-1}\boldsymbol{G_K}\boldsymbol{B}_{i,j}(-b))$
      $\boldsymbol{G_k} \leftarrow \boldsymbol{B}_{i^*,j^*}(-b)^{-1}\boldsymbol{G_k}\boldsymbol{B}_{i^*,j^*}(-b)$, for $k = 0, \ldots, K$
    until no further improvement
    $b \leftarrow b/2$
  end while
  return $(\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})$
end procedure

Let us consider a numerical example for the moment matching procedure. The marginal and lag-1 joint moments for this example were taken from the LBL trace (also used in Section 2.3.1, consisting of TCP traffic measurements), which was normalized to $m_1 = 1$:

$$\{m_1, m_2, m_3, m_4, m_5\} = \{1,\ 2.942,\ 16.84,\ 150.73,\ 1876.8\},$$

$$\boldsymbol{N} = \begin{bmatrix} 1 & m_1 & m_2 \\ m_1 & \eta_{1,1} & \eta_{1,2} \\ m_2 & \eta_{2,1} & \eta_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2.942 \\ 1 & 1.3013 & 4.5056 \\ 2.942 & 4.5 & 17.416 \end{bmatrix}.$$

First $(\underline{\sigma}, \boldsymbol{S})$ is obtained by procedure [82] based on the marginal moments $m_1, \ldots, m_5$, resulting

$$\underline{\sigma} = \begin{bmatrix} 0.33333 & 0.33333 & 0.33333 \end{bmatrix},$$

$$\boldsymbol{S} = \begin{bmatrix} -1.7356 & 0.34074 & -0.95214 \\ -0.18575 & -0.63031 & -0.042169 \\ -0.48092 & -0.036353 & -0.6245 \end{bmatrix}.$$

Then, matrices $\boldsymbol{\Lambda}_\sigma$, $\boldsymbol{\Lambda}_{\mathbb{1}}$ are built and $\boldsymbol{R}$ is computed from the matrix of joint moments. The (non-Markovian) representation of the resulting RAP is

$$\boldsymbol{D_0} = \begin{bmatrix} -1.7356 & 0.34074 & -0.95214 \\ -0.18575 & -0.63031 & -0.042169 \\ -0.48092 & -0.036353 & -0.6245 \end{bmatrix}, \boldsymbol{D_1} = \begin{bmatrix} 1.4612 & -0.24037 & 1.1263 \\ 0.1905 & 0.53436 & 0.13337 \\ 0.47616 & 0.1323 & 0.5333 \end{bmatrix}.$$

Finally, $(\boldsymbol{D_0}, \boldsymbol{D_1})$ was transformed to a valid MAP with Algorithm 3 (in less than a second), giving

$$\boldsymbol{G_0} = \begin{bmatrix} -2.0161 & 0.091134 & 0.029499 \\ 0.014955 & -0.63564 & 0.0079826 \\ 0.08378 & 0.062089 & -0.33873 \end{bmatrix}, \boldsymbol{G_1} = \begin{bmatrix} 1.8439 & 0.04524 & 0.0063419 \\ 0.063835 & 0.54292 & 0.0059518 \\ 0.047433 & 0.003395 & 0.14203 \end{bmatrix}.$$

3.3 OBTAINING AN APPROXIMATE MARKOVIAN REPRESENTATION

The procedure presented in Section 3.2.4 does not always succeed to return a valid Markovian representation. For instance, if the output of the moment matching procedure is such that the (joint) density of the inter-arrival times is negative, then the process itself is invalid, which can not be fixed by any similarity transformations.

In such cases the goal is to find a valid MMAP that is as "close" as possible to the invalid process. This section presents two algorithms developed for this purpose.

3.3.1 *The two-step fitting approach*

Compared to PH fitting methods, developing MMAP fitting algorithms is substantially more complex computationally. MMAPs have much more parameters, and it is not obvious either what kind of distance function to use during the optimization.

The idea appearing in [50], referred to as the *two-step fitting approach*, simplifies the fitting process significantly by cutting the complex fitting process into two much smaller non-linear optimization problems. Generally speaking, the main idea of the applied approach is that the $\boldsymbol{D_0}$ and the $\boldsymbol{D_1}$ matrices are constructed separately.

- In the first step, the inter-arrival time distribution is fitted by a PH distribution, which determines matrix $\boldsymbol{D_0}$ (the generator of the PH distribution) and vector $\underline{\alpha}$ (the initial probability vector of the PH distribution).
- Then matrix $\boldsymbol{D_1}$ is constructed, such that the inter-arrival time distribution of the resulting MAP is kept the same, and its correlation structure approximates the one of the target process as much as possible. In this step matrix $\boldsymbol{D_1}$ has to satisfy the following two constraints to maintain the inter-arrival time distribution determined in the first step:

  C1: $\boldsymbol{D_1}\mathbb{1} = -\boldsymbol{D_0}\mathbb{1}$,

  C2: $\underline{\alpha}(-\boldsymbol{D_0})^{-1}\boldsymbol{D_1} = \underline{\alpha}$.

The first step is a PH fitting problem for which any PH fitting procedure can be used.

In the second step a non-linear optimization program needs to be solved. In the next subsections the goal function of the optimization is the L_2 distance of the lag-1 joint moments (Section 3.3.3) and the lag-1 joint densities (Section 3.3.4). However, the two-phase MMAP fitting approach is more general, many alternative statistical quantities can be taken into account in the optimization to approximate the correlation of the inter-arrival times.

3.3.2 *Fitting the distribution of the inter-arrival times*

The inter-arrival time distribution is fitted by a PH distribution providing matrix $\boldsymbol{D_0}$ and vector $\underline{\alpha}$. However, it does matter what the structure of this PH distribution is: it determines how successful the second step of the two-step MMAP fitting method is going to be. The preferred PH distribution has a structure by which constraints **C1** and **C2** leave enough "degrees of freedom" for the optimization.

E.g., if the PH distribution is such that the absorbing state is reachable only from a single state, then due to constraint **C1** matrix $\boldsymbol{D_1}$ can have only a single non-zero row, implying that

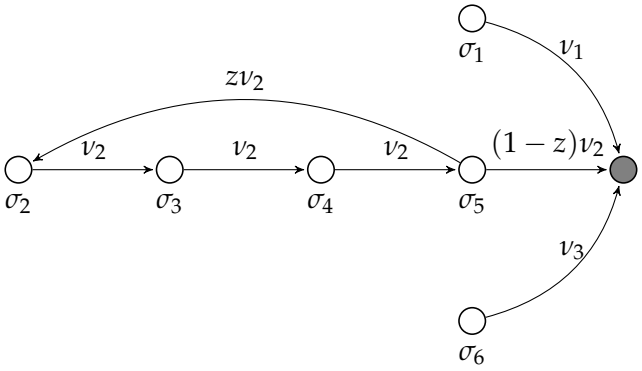


Figure 18.: Example for a hyper-FEB PH distribution

the resulting MMAP can have no correlation at all. This means that the canonical forms for APH distributions (see Figure 4) are not suitable for the two-step fitting approach.

Similarly, no correlation can be achieved due to **C2** if vector $\underline{\alpha}$ of the PH distribution has only a single non-zero element.

In the literature several articles investigate the optimal PH structure for MMAP fitting (see [17] or [50]) proposing various heuristic transformation methods, but exact solution for this problem does not exist yet.

According to our numerical experiments, special PH structures, e.g., hyper-exponential and hyper-Erlang distributions (Figure 3) perform very well in this algorithm, since they have many non-zero elements both in vector $\underline{\alpha}$ and in vector $\boldsymbol{D_0}\mathbb{1}$.

To solve this problem [23] develops a moment matching method that returns a hyper-exponential distribution of order N based on $2N-1$ moments, if it is possible. An other solution published in [51] is based on a hyper-Erlang distribution of common order, which always succeeds if an appropriately large Erlang order is chosen.

Our method of choice, however, is a slight modification of the algorithm presented in Section 2.4, which is the generalization of the former two. It constructs PH distributions from FEBs (see Section 2.1.4), where each FEB implements an eigenvalue of the target distribution. A FEB consisting of a single state represents a real eigenvalue. With FEBs it is possible to represent complex eigenvalues as well, as opposed to the previously mentioned methods. The original method in Section 2.4 puts the FEBs in a row (as in Figure 5), which is not appropriate for our goals, since there is only a single state connected to the absorbing state, implying that no correlation can be realized.

However, the original method can be modified in a straight forward way to return a hyper-FEB structure (as shown in Figure 18). In this modification the vector-matrix pair $(\underline{\sigma}, \boldsymbol{S})$ given by (50) is transformed to a hyper-FEB representation instead of the monocyclic representation.

3.3.3 *Approximate Markovian representation by fitting the joint moments*

As in case of PH distributions, the fitting problem can be made simpler with an appropriately defined subclass in case of MMAPs as well.

In this section we are going to use a sub-class of MMAPs called structured marked Markovian arrival processes (SMMAPs), that are the multi-type extensions of structured Markovian arrival processes (SMAPs) introduced in [8], which are the generalizations of the ER-CHMM structure introduced in [68].

In a SMMAP we have M PH distributed branches with branch i consisting of N_i phases. The phases of the entire system have a two-dimensional identifier: phase (i,n) identifies state n in branch i. Each inter-arrival time is PH distributed, and the choice determining which branch generates the next inter-arrival time is Markovian. The parameters characterizing this process are as follows.

- A size M set of PH distributions characterized by $\{(\underline{\sigma}^{(i)}, \boldsymbol{S}^{(i)}), i = 1,\ldots,M\}$, called *components*. Each inter-arrival time is generated by one of the component PH distributions.
- Probabilities $p_{i,j}^{(k)}$ with $k = 1,\ldots,K,\ i,j = 1,\ldots,M$. $p_{i,j}^{(k)}$ represents the probability that the next inter-arrival time will be generated by component j given that the previous one was generated by component i resulting in a type k arrival.

According to the definition matrix $\boldsymbol{D_0}$ is given by

$$\boldsymbol{D_0} = \begin{bmatrix} \boldsymbol{S}^{(1)} & & & \\ & \boldsymbol{S}^{(2)} & & \\ & & \ddots & \\ & & & \boldsymbol{S}^{(M)} \end{bmatrix}, \tag{84}$$

and matrices $\boldsymbol{D_k}, k = 1,\ldots,K$, are

$$\boldsymbol{D_k} = \begin{bmatrix} \underline{s}^{(1)}\underline{\sigma}^{(1)}p_{1,1}^{(k)} & \underline{s}^{(1)}\underline{\sigma}^{(2)}p_{1,2}^{(k)} & \ldots & \underline{s}^{(1)}\underline{\sigma}^{(M)}p_{1,M}^{(k)} \\ \underline{s}^{(2)}\underline{\sigma}^{(1)}p_{2,1}^{(k)} & \underline{s}^{(2)}\underline{\sigma}^{(2)}p_{2,2}^{(k)} & \ldots & \underline{s}^{(2)}\underline{\sigma}^{(M)}p_{2,M}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{s}^{(M)}\underline{\sigma}^{(1)}p_{M,1}^{(k)} & \underline{s}^{(M)}\underline{\sigma}^{(2)}p_{M,2}^{(k)} & \ldots & \underline{s}^{(M)}\underline{\sigma}^{(M)}p_{M,M}^{(k)} \end{bmatrix}, \tag{85}$$

where column vector $\underline{s}^{(i)} = (-\boldsymbol{S}^{(i)})\mathbb{1},\ i = 1,\ldots,M$.

Introducing matrices $\boldsymbol{P}^{(k)} = \{p_{i,j}^{(k)},\ i,j = 1,\ldots,M\}$ and $\boldsymbol{P} = \sum_{k=1}^{K}\boldsymbol{P}^{(k)}$, the row vector $\underline{\pi} = \{\pi_i,\ i = 1,\ldots,M\}$ describing the steady state probabilities of the components is the solution of linear system $\underline{\pi}\boldsymbol{P} = \underline{\pi}, \underline{\pi}\mathbb{1} = 1$. One of the major benefits of using SMMAPs is that the marginal and joint moments are given by simple formulas as

$$m_i = E\left(\mathcal{F}^i\right) = \sum_{a=1}^{M}\pi_a m_i^{(a)}, \tag{86}$$

$$\eta_{i,j}^{(k)} = E\left((\mathcal{F}_0^{(k)})^i(\mathcal{F}_1)^j\right) = \sum_{a=1}^{M}\sum_{b=1}^{M}\pi_a p_{a,b}^{(k)} m_i^{(a)} m_j^{(b)}, \tag{87}$$

where $m_i^{(a)}$ is the ith moment of component PH distribution a.

In the two-step fitting framework the inter-arrival times and the correlation of the arrival process are fitted consecutively. Section 3.3.2 proposes a hyper-FEB PH structure for the former step. The hyper-FEB PH structure makes it easy to derive the component distributions: each FEB branch constitutes one component. Matrix $\boldsymbol{S}^{(i)}$ is the generator of the ith branch, and vector $\underline{\sigma}^{(i)}$ is composed by the (normalized) initial probabilities of branch i. Furthermore, the inter-arrival time distribution imposes a constraint for correlation fitting: the steady state component probability π_i is the probability of starting from FEB branch i, thus π_i is the sum of the corresponding elements of vector $\underline{\sigma}$.

For fitting the correlations of the inter-arrival times, the subject function is the relative L_2 distance of the lag-1 joint moments, thus the optimization problem can be formulated as

$$\boldsymbol{P_1}, \ldots, \boldsymbol{P_K} = \underset{\boldsymbol{P_1},\ldots,\boldsymbol{P_K}}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{R} \sum_{j=1}^{R} \left(\frac{\eta_{i,j}^{(k)} - \hat{\eta}_{i,j}^{(k)}}{\hat{\eta}_{i,j}^{(k)}} \right)^2, \tag{88}$$

where $\hat{\eta}_{i,j}^{(k)}$ denotes the target joint moments and R is the number of joint moments to approximate.

Inserting (87) into (88) leads to

$$\boldsymbol{P_1}, \ldots, \boldsymbol{P_K} = \underset{\boldsymbol{P_1},\ldots,\boldsymbol{P_K}}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{R} \sum_{j=1}^{R} \left(\frac{\sum_{a=1}^{M} \sum_{b=1}^{M} \pi_a p_{a,b}^{(k)} m_i^{(a)} m_j^{(b)} - \hat{\eta}_{i,j}^{(k)}}{\hat{\eta}_{i,j}^{(k)}} \right)^2, \tag{89}$$

where the variables to optimize are only the elements of matrices $\boldsymbol{P_k}, k = 1, \ldots, K$, everything else is given. The constraints of the optimization are

$$\begin{aligned} \underline{\pi} \sum_{k=1}^{K} \boldsymbol{P_k} &= \underline{\pi}, \\ \sum_{k=1}^{K} \boldsymbol{P_k} \mathbb{1} &= \mathbb{1}, \\ \boldsymbol{P_k} &\geq \boldsymbol{0}, \quad \text{for } k = 1, \ldots, K. \end{aligned} \tag{90}$$

Observe that this nonlinear program is a non-negative least-squares (NNLS) problem with linear equality and inequality constraints ([37],[59]), which is a relatively easy to solve subclass of non-linear optimization problems[1].

A similar approach has been applied in [19] on the basis of the overall state space of the Markov chain where matrix $\boldsymbol{D_1}$ was fitted for given matrix $\boldsymbol{D_0}$. In [19] the equations for several joint moments were composed to a single NNLS favoring higher order joint moments, so that the fitting of lower order joint moments got worse. One possibility to cope with this problem is to introduce weight functions to privilege lower order moments, but there is no general rule which weight functions are appropriate [19]. To avoid weighting of the joint moments of different order we propose a *step-by-step* fitting algorithm here. In the nth step, joint moments $\eta_{i,j}^{(k)}$, $i + j = n$ are the subject of fitting while keeping the already fitted $\eta_{i,j}^{(k)}$, $i + j < n$ moments unchanged.

In the first step, only the joint moments of order $i = 1, j = 1$ are the subject of fitting, and the only linear constraints are the "standard" ones, given by (90). Suppose $\tilde{\eta}_{1,1}^{(k)}, k = 1, \ldots, K$ are the optimal solutions. In the next step the NNLS problem is formulated for joint moments of order $i + j = 3$, and new linear constraints are added that ensure that the lower order joint moments found to be optimal in the previous step are preserved. These additional constraints have a form of $\tilde{\eta}_{1,1}^{(k)} = \sum_{a=1}^{M} \sum_{b=1}^{M} \pi_a p_{a,b}^{(k)} m_1^{(a)} m_1^{(b)}$ for $k = 1, \ldots, K$. In each step, only the joint moments belonging to the same order are optimized and the optimal results for lower order joint moments are preserved with the appropriate additional linear constraints.

Algorithm 4 gives the formal description of the procedure. The inputs of the algorithm are the target marginal and joint moments to fit, and the algorithm returns the matrices characterizing a valid MMAP.

1 A possible implementation is available at http://suvrit.de/software.html

Algorithm 4 Step-by-step fitting of lag-1 joint moments

procedure FIT-JOINT-MOMS($\hat{m}_i,\ i = 1, \ldots, 2M-1,\ \hat{\eta}_{i,j}^{(k)},\ i,j = 1, \ldots, R,\ k = 1, \ldots, K$)
  $\mathcal{P} \leftarrow$ hyper-FEB solutions of moment matching based on $\hat{m}_1, \hat{m}_2, \ldots$
  $\{\boldsymbol{D}_0^*, \ldots, \boldsymbol{D}_K^*\} \leftarrow \emptyset$
  for each solution $(\underline{\sigma}, \boldsymbol{S})$ in $\mathcal{P}$ **do**
    obtain component parameters $(\underline{\sigma}^{(i)}, \boldsymbol{S}^{(i)})$ and π_i from $(\underline{\sigma}, \boldsymbol{S})$ for $i = 1, \ldots, M$
    $eqns \leftarrow \{\underline{\pi}\boldsymbol{P} = \underline{\pi}, \boldsymbol{P}\mathbb{1} = \mathbb{1}\}$
    for $n = 2, \ldots, 2R$ **do**
      Solve $\boldsymbol{Q_1}, \ldots, \boldsymbol{Q_K} \leftarrow \underset{\boldsymbol{P_1}, \ldots, \boldsymbol{P_K}}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{n-1} \sum_{j=1}^{n-i} \left(\frac{\sum_{a=1}^{M} \sum_{b=1}^{M} \pi_a p_{a,b}^{(k)} m_i^{(a)} m_j^{(b)} - \hat{\eta}_{i,j}^{(k)}}{\hat{\eta}_{i,j}^{(k)}} \right)^2$
      Subject to $eqns$ and $\boldsymbol{P}_k \geq \boldsymbol{0}, k = 1, \ldots, K$
      for $i, j = 1, \ldots, R$ with $i + j = n$ and $k = 1, \ldots, K$ **do**
        $\eta_{i,j}^{(k)} = \sum_{a=1}^{M} \sum_{b=1}^{M} \pi_a q_{a,b}^{(k)} m_i^{(a)} m_j^{(b)}$
        Add new constraint $eqns \leftarrow eqns \cup \left\{ \eta_{i,j}^{(k)} = \sum_{a=1}^{M} \sum_{b=1}^{M} \pi_a p_{a,b}^{(k)} m_i^{(a)} m_j^{(b)} \right\}$
      end for
    end for
    Build matrices $\{\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}\}$ based on (84) and (85)
    if solution is better than $\{\boldsymbol{D}_0^*, \ldots, \boldsymbol{D}_K^*\}$ **then**
      $\{\boldsymbol{D}_0^*, \ldots, \boldsymbol{D}_K^*\} \leftarrow \{\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}\}$
    end if
  end for
  return $\{\boldsymbol{D}_0^*, \ldots, \boldsymbol{D}_K^*\}$
end procedure

Alternatively, the step-by-step fitting of the joint moments of different orders can be omitted by performing the fitting in a single NNLS step involving all the joint moments up to order R, i.e. the for-loop of Algorithm 4 is replaced by a single NNLS problem with the initial equations as conditions. This (simpler) variant of the algorithm will be referred to as *one-step* method in the sequel.

To demonstrate the behavior of the algorithm, we are going to fit a MAP for the BC-pAug89 trace (also used in Section 2.3.2 and in Section 2.4.3) consisting of measurements on an Ethernet network. The moment matching procedure introduced in Section 3.2.2 returned matrices that we failed to transform to a Markovian representation. Executing the *step-by-step* variant of the presented fitting method, however, returned a valid MAP characterized by

$$\begin{aligned} \boldsymbol{D_0} &= \begin{bmatrix} -0.051901 & 0 & 0 \\ 0 & -0.15118 & 0 \\ 0 & 0 & -1.2404 \end{bmatrix}, \\ \boldsymbol{D_1} &= \begin{bmatrix} 0 & 5.19 \cdot 10^{-7} & 0.0519 \\ 0 & 0.092536 & 0.05864 \\ 0.0001297 & 0.016444 & 1.2238 \end{bmatrix}. \end{aligned} \tag{91}$$

The first five marginal moments of this MAP are exact, they are the same as the ones of the measurements. The joint moments are approximated reasonably well as well. The matrix of joint moments corresponding to the measurements ($\boldsymbol{N}_{BC}$) and to the step-by-step fitting procedure ($\boldsymbol{N}_{step-by-step}$) are

$$\boldsymbol{N}_{BC} = \begin{bmatrix} 1.6449 & 13.786 & 256.48 \\ 13.786 & 159.62 & 3116.1 \\ 256.48 & 3116.1 & 61177 \end{bmatrix}, \boldsymbol{N}_{step-by-step} = \begin{bmatrix} 1.6449 & 11.484 & 203.36 \\ 10.916 & 106.58 & 2169.4 \\ 191.23 & 2300.8 & 54775 \end{bmatrix}.$$

Observe that $\eta_{1,1}$, which determines the lag-1 auto-correlation, is exact. The *one-step* variant of the algorithm achieves slightly better results in the higher joint moments (although not everywhere), but fails to capture the lower ones. The corresponding joint moments are

$$\boldsymbol{N}_{one-step} = \begin{bmatrix} 1.5842 & 11.28 & 197.15 \\ 11.194 & 109.62 & 2254 \\ 192.6 & 2218.5 & 54349 \end{bmatrix}. \tag{92}$$

3.3.4 *Approximate Markovian representation by fitting the joint distribution*

The procedure presented in this section provides an alternative method to construct matrices $\boldsymbol{D}_k, k = 1, \ldots, K$ of the process to approximate (remember that matrix $\boldsymbol{D_0}$ and vector $\underline{\alpha}$ are already available due to Section 3.3.2).

While Section 3.3.3 created matrices $\boldsymbol{D}_k, k = 1, \ldots, K$ such that the L_2 distance of the joint moments are minimized, the goal here is to minimize the L_2 distance of the joint density functions up to a given lag k.

Before formalizing and solving the optimization problem we first provide an exact definition of the distance function and provide an efficient numerical procedure to evaluate it. The efficient evaluation of the distance function ensures the quick termination of the optimization algorithm.

Let us consider two MMAPs, $\mathcal{F} = (\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ and $\mathcal{G} = (\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})$. The squared difference (L_2 distance) of the joint density of the inter-arrival times up to lag-k is defined by

$$\begin{aligned}\mathcal{D}_k\{\mathcal{F},\mathcal{G}\} = \sum_{n_1=1}^{K} \cdots \sum_{n_{k-1}=1}^{K} \sum_{n_k=1}^{K} \int_0^\infty \cdots \int_0^\infty \int_0^\infty \\ \Big(\underline{\alpha}_{\mathcal{F}} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{n_1}} \cdots e^{\boldsymbol{D_0} x_{k-1}} \boldsymbol{D_{n_{k-1}}} \cdot e^{\boldsymbol{D_0} x_k} \boldsymbol{D_{n_k}} \mathbb{1} \\ - \underline{\alpha}_{\mathcal{G}} e^{\boldsymbol{G_0} x_1} \boldsymbol{G_{n_1}} \cdots e^{\boldsymbol{G_0} x_{k-1}} \boldsymbol{G_{n_{k-1}}} \cdot e^{\boldsymbol{G_0} x_k} \boldsymbol{G_{n_k}} \mathbb{1} \Big)^2 \\ dx_1 \ldots dx_{k-1}\, dx_k,\end{aligned} \tag{93}$$

where $\underline{\alpha}_{\mathcal{F}}$ and $\underline{\alpha}_{\mathcal{G}}$ denote the stationary phase distributions of MMAPs $\mathcal{F}$ and $\mathcal{G}$ at arrival instants. The squared distance is summed up for all combinations of arrival types up to lag k. The square term expands to

$$\mathcal{D}_k\{\mathcal{F},\mathcal{G}\} = L_k(\mathcal{F},\mathcal{F}) - 2L_k(\mathcal{F},\mathcal{G}) + L_k(\mathcal{G},\mathcal{G}), \tag{94}$$

where $L_k(\mathcal{F},\mathcal{G})$ represents the integral

$$\begin{aligned}L_k(\mathcal{F},\mathcal{G}) = \sum_{n_1=1}^{K} \cdots \sum_{n_{k-1}=1}^{K} \sum_{n_k=1}^{K} \int_0^\infty \cdots \int_0^\infty \int_0^\infty \\ \underline{\alpha}_{\mathcal{F}} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{n_1}} \cdots e^{\boldsymbol{D_0} x_{k-1}} \boldsymbol{D_{n_{k-1}}} \cdot e^{\boldsymbol{D_0} x_k} \boldsymbol{D_{n_k}} \mathbb{1} \\ \cdot \underline{\alpha}_{\mathcal{G}} e^{\boldsymbol{G_0} x_1} \boldsymbol{G_{n_1}} \cdots e^{\boldsymbol{G_0} x_{k-1}} \boldsymbol{G_{n_{k-1}}} \cdot e^{\boldsymbol{G_0} x_k} \boldsymbol{G_{n_k}} \mathbb{1}\, dx_1 \ldots dx_{k-1}\, dx_k.\end{aligned} \tag{95}$$

The following theorem provides a procedure to evaluate this integral with recursive solutions of k Sylvester equations.

Theorem 14. *$L_k(\mathcal{F},\mathcal{G})$ can be expressed by*

$$L_k(\mathcal{F},\mathcal{G}) = \sum_{n=1}^{K} \mathbb{1}^T \boldsymbol{G_n}^T \cdot \boldsymbol{Y_k} \cdot \boldsymbol{D_n} \mathbb{1}, \tag{96}$$

where matrix $\boldsymbol{Y_k}$ is the solution of the recursive Sylvester equation

$$\begin{cases} -\sum_{n=1}^{K} \boldsymbol{G_n}^T \boldsymbol{Y_{k-1}} \boldsymbol{D_n} = \boldsymbol{G_0}^T \boldsymbol{Y_k} + \boldsymbol{Y_k} \boldsymbol{D_0} & \text{for } k > 1, \\ -\underline{\alpha}_{\mathcal{G}}^T \underline{\alpha}_{\mathcal{F}} = \boldsymbol{G_0}^T \boldsymbol{Y_1} + \boldsymbol{Y_1} \boldsymbol{D_0} & \text{for } k = 1. \end{cases} \tag{97}$$

Proof. We start by transforming (95) as

$$\begin{aligned}&L_k(\mathcal{F},\mathcal{G}) \\ &= \sum_{n_k=1}^{K} \sum_{n_{k-1}=1}^{K} \cdots \sum_{n_1=1}^{K} \int_0^\infty \cdots \int_0^\infty \int_0^\infty \mathbb{1}^T \boldsymbol{G_{n_k}}^T e^{\boldsymbol{G_0}^T x_k} \boldsymbol{G_{n_{k-1}}}^T e^{\boldsymbol{G_0}^T x_{k-1}} \cdots \boldsymbol{G_{n_1}}^T e^{\boldsymbol{G_0}^T x_1} \underline{\alpha}_{\mathcal{G}}^T \\ &\cdot \underline{\alpha}_{\mathcal{F}} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{n_1}} \cdots e^{\boldsymbol{D_0} x_{k-1}} \boldsymbol{D_{n_{k-1}}} \cdot e^{\boldsymbol{D_0} x_k} \boldsymbol{D_{n_k}} \mathbb{1}\, dx_1 \ldots dx_{k-1}\, dx_k \\ &= \sum_{n_k=1}^{K} \mathbb{1}^T \boldsymbol{G_{n_k}}^T \Bigg(\sum_{n_{k-1}=1}^{K} \cdots \sum_{n_1=1}^{K} \int_0^\infty \cdots \int_0^\infty \int_0^\infty e^{\boldsymbol{G_0}^T x_k} \boldsymbol{G_{n_{k-1}}}^T e^{\boldsymbol{G_0}^T x_{k-1}} \cdots \boldsymbol{G_{n_1}}^T e^{\boldsymbol{G_0}^T x_1} \underline{\alpha}_{\mathcal{G}}^T \\ &\cdot \underline{\alpha}_{\mathcal{F}} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{n_1}} \cdots e^{\boldsymbol{D_0} x_{k-1}} \boldsymbol{D_{n_{k-1}}} \cdot e^{\boldsymbol{D_0} x_k}\, dx_1 \ldots dx_{k-1}\, dx_k \Bigg) \cdot \boldsymbol{D_{n_k}} \mathbb{1}.\end{aligned}$$

Let us denote the term in the parenthesis by $\boldsymbol{Y_k}$. For $k > 1$, separating the first and the last terms leads to the recursion

$$\begin{aligned}\boldsymbol{Y_k} &= \sum_{n_{k-1}=1}^{K} \int_0^\infty e^{\boldsymbol{G_0}^T x_k} \cdot \boldsymbol{G_{n_{k-1}}}^T \Bigg(\sum_{n_{k-2}=1}^{K} \cdots \sum_{n_1=1}^{K} \int_0^\infty \cdots \int_0^\infty e^{\boldsymbol{G_0}^T x_{k-1}} \boldsymbol{G_{n_{k-2}}}^T \cdots \boldsymbol{G_{n_1}}^T e^{\boldsymbol{G_0}^T x_1} \underline{\alpha}_{\mathcal{G}}^T \\ &\quad \cdot \underline{\alpha}_{\mathcal{F}} e^{\boldsymbol{D_0} x_1} \boldsymbol{D_{n_1}} \cdots e^{\boldsymbol{D_0} x_{k-1}} \, dx_1 \ldots dx_{k-1} \Bigg) \boldsymbol{D_{n_{k-1}}} \cdot e^{\boldsymbol{D_0} x_k} \, dx_k \\ &= \sum_{n_{k-1}=1}^{K} \int_0^\infty e^{\boldsymbol{G_0}^T x_k} \boldsymbol{G_{n_{k-1}}}^T \cdot \boldsymbol{Y_{k-1}} \cdot \boldsymbol{D_{n_{k-1}}} e^{\boldsymbol{D_0} x_k} \, dx_k, \end{aligned} \tag{98}$$

which is the solution of Sylvester equation $-\sum_{n=1}^{K} \boldsymbol{G_n}^T \boldsymbol{Y_{k-1}} \boldsymbol{D_n} = \boldsymbol{G_0}^T \boldsymbol{Y_k} + \boldsymbol{Y_k} \boldsymbol{D_0}$ (see Theorem 36 in Appendix A.2). The equation for $k = 1$ is obtained similarly. □

Note that the solution of (97) is always unique as matrices $\boldsymbol{D_0}$ and $\boldsymbol{G_0}$ are sub-generators.

Having defined the distance function, the optimization problem providing matrices $\boldsymbol{G_k}$, $k = 1, \ldots, K$ can be formulated by

$$(\boldsymbol{G_1}, \ldots, \boldsymbol{G_K}) = \underset{\boldsymbol{G_1}, \ldots, \boldsymbol{G_K}}{\arg\min} \, \mathcal{D}_k\{(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}), (\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})\}, \tag{99}$$

where matrices $(\boldsymbol{D_0}, \ldots, \boldsymbol{D_K})$ correspond to the MRAP to approximate (not having a Markovian representation or possibly not a valid process at all), and matrices $(\boldsymbol{G_0}, \ldots, \boldsymbol{G_K})$ are the matrices of a valid MMAP.

In the single-type case ($K = 1$), if the approximation is based on the lag-1 behavior only and ignores the distance of higher dimensional joint densities, then the optimization (99) reduces to a quadratic programming problem.

Theorem 15. *Given that $\underline{\alpha}_{\mathcal{G}}$ and $\boldsymbol{G_0}$ are available, matrix $\boldsymbol{G_1}$ minimizing $\mathcal{D}_2\{\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}\}$ is the solution of the quadratic program*

$$\min_{\boldsymbol{G_1}} \left\{ vec\langle \boldsymbol{G_1} \rangle^T (\boldsymbol{W_{BB}} \otimes \boldsymbol{Y_{BB}}) vec\langle \boldsymbol{G_1} \rangle - 2 vec\langle \boldsymbol{D_1} \rangle^T (\boldsymbol{W_{AB}} \otimes \boldsymbol{Y_{AB}}) vec\langle \boldsymbol{G_1} \rangle \right\} \tag{100}$$

subject to

$$\left(\boldsymbol{I} \otimes \underline{\alpha}_{\mathcal{G}} (-\boldsymbol{G_0})^{-1} \right) vec\langle \boldsymbol{G_1} \rangle = \underline{\alpha}_{\mathcal{F}}, \tag{101}$$

$$(\mathbb{1}^T \otimes \boldsymbol{I}) vec\langle \boldsymbol{G_1} \rangle = -\boldsymbol{G_0} \mathbb{1}. \tag{102}$$

Matrices $\boldsymbol{W_{AB}}, \boldsymbol{W_{BB}}, \boldsymbol{Y_{AB}}$ and $\boldsymbol{Y_{BB}}$ are the solutions to Sylvester equations

$$\boldsymbol{D_0} \boldsymbol{W_{AB}} + \boldsymbol{W_{AB}} \boldsymbol{G_0}^T = -\boldsymbol{D_0} \mathbb{1} \cdot \mathbb{1}^T \boldsymbol{G_0}^T, \tag{103}$$

$$\boldsymbol{G_0} \boldsymbol{W_{BB}} + \boldsymbol{W_{BB}} \boldsymbol{G_0}^T = -\boldsymbol{G_0} \mathbb{1} \cdot \mathbb{1}^T \boldsymbol{G_0}^T, \tag{104}$$

$$\boldsymbol{D_0}^T \boldsymbol{Y_{AB}} + \boldsymbol{Y_{AB}} \boldsymbol{G_0} = -\underline{\alpha}_{\mathcal{F}}^T \cdot \underline{\alpha}_{\mathcal{G}}, \tag{105}$$

$$\boldsymbol{G_0}^T \boldsymbol{Y_{BB}} + \boldsymbol{Y_{BB}} \boldsymbol{G_0} = -\underline{\alpha}_{\mathcal{G}}^T \cdot \underline{\alpha}_{\mathcal{G}}. \tag{106}$$

Proof. Let us first apply the vec$\langle\rangle$ operator (column stacking, see Appendix A.1) on (96) at $K = 1, k = 2$. Utilizing the identity (353) and the identity (354) we get

$$\text{vec}\langle L_2(\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}) \rangle = (\mathbb{1}^T \boldsymbol{D_0}^T \otimes \mathbb{1}^T \boldsymbol{G_0}^T) \cdot \text{vec}\langle \boldsymbol{Y_2} \rangle = \text{vec}\langle \boldsymbol{G_0} \mathbb{1} \cdot \mathbb{1}^T \boldsymbol{D_0}^T \rangle^T \cdot \text{vec}\langle \boldsymbol{Y_2} \rangle. \tag{107}$$

Applying the vec$\langle\rangle$ operator on both sides of (97) and using (353) again leads to

$$-(\boldsymbol{I}\otimes\boldsymbol{G_1}^T\boldsymbol{Y_1})\mathrm{vec}\langle\boldsymbol{D_1}\rangle=(\boldsymbol{I}\otimes\boldsymbol{G_0}^T)\mathrm{vec}\langle\boldsymbol{Y_2}\rangle+(\boldsymbol{D_0}^T\otimes\boldsymbol{I})\mathrm{vec}\langle\boldsymbol{Y_2}\rangle, \tag{108}$$

from which vec$\langle\boldsymbol{Y_2}\rangle$ is expressed by

$$\mathrm{vec}\langle\boldsymbol{Y_2}\rangle=(-\boldsymbol{D_0}^T\oplus\boldsymbol{G_0}^T)^{-1}(\boldsymbol{I}\otimes\boldsymbol{G_1}^T)(\boldsymbol{I}\otimes\boldsymbol{Y_{AB}})\mathrm{vec}\langle\boldsymbol{D_1}\rangle, \tag{109}$$

since $\boldsymbol{Y_1}=\boldsymbol{Y_{AB}}$. Thus we have

$$\mathrm{vec}\langle L_2(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}})\rangle=\underbrace{\mathrm{vec}\langle\boldsymbol{G_0}\mathbb{1}\cdot\mathbb{1}^T\boldsymbol{D_0}^T\rangle^T(-\boldsymbol{D_0}^T\oplus\boldsymbol{G_0}^T)^{-1}}_{\mathrm{vec}\langle\boldsymbol{W_{AB}}\rangle^T}(\boldsymbol{I}\otimes\boldsymbol{G_1}^T)(\boldsymbol{I}\otimes\boldsymbol{Y_{AB}})\mathrm{vec}\langle\boldsymbol{D_1}\rangle, \tag{110}$$

where we recognized that the transpose of vec$\langle\boldsymbol{W_{AB}}\rangle$ expressed from (103) matches the first two terms of the expression. Using the identities of the vec$\langle\rangle$ operator yields

$$\mathrm{vec}\langle\boldsymbol{W_{AB}}\rangle^T(\boldsymbol{I}\otimes\boldsymbol{G_1}^T)=\mathrm{vec}\langle\boldsymbol{G_1}^T\boldsymbol{W_{AB}}\rangle^T=\mathrm{vec}\langle\boldsymbol{G_1}\rangle^T(\boldsymbol{W_{AB}}\otimes\boldsymbol{I}). \tag{111}$$

Finally, putting together (110) and (111) gives

$$\mathrm{vec}\langle L_2(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}})\rangle=\mathrm{vec}\langle\boldsymbol{G_1}\rangle^T(\boldsymbol{W_{AB}}\otimes\boldsymbol{Y_{AB}})\mathrm{vec}\langle\boldsymbol{D_1}\rangle. \tag{112}$$

From the components of $\mathcal{D}_2\{\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}}\}$ (see (94)) $L_2(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{F}})$ plays no role in the optimization as it does not depend on $\boldsymbol{G_1}$, the term $L_2(\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}})$ yields the linear term in (100) according to (112), and $L_2(\boldsymbol{\mathcal{G}},\boldsymbol{\mathcal{G}})$ introduces the quadratic term, based on (112) after replacing $\boldsymbol{\mathcal{F}}$ by $\boldsymbol{\mathcal{G}}$.

According to the first constraint (101) and the second constraint (102) the solution must satisfy $\underline{\alpha}_{\mathcal{G}}(-\boldsymbol{G_0})^{-1}\boldsymbol{G_1}=\underline{\alpha}_{\mathcal{G}}$ and $\boldsymbol{G_1}\mathbb{1}=-\boldsymbol{G_0}\mathbb{1}$, respectively. □

Theorem 16. *Matrix $\boldsymbol{W_{BB}}\otimes\boldsymbol{Y_{BB}}$ is positive definite, thus the quadratic optimization problem of Theorem 15 is convex.*

Proof. If $\boldsymbol{W_{BB}}$ and $\boldsymbol{Y_{BB}}$ are positive definite, then their Kronecker product is positive definite as well. First we show that matrix $\boldsymbol{Y_{BB}}$ is positive definite, thus $\underline{z}\,\boldsymbol{Y_{BB}}\,\underline{z}^T>0$ holds for any non-zero row vector z. Since $\boldsymbol{Y_{BB}}$ is the solution of a Sylvester equation, we have that $\boldsymbol{Y_{BB}}=\int_0^\infty e^{\boldsymbol{G_0}^Tx}\underline{\alpha}_{\mathcal{G}}^T\cdot\underline{\alpha}_{\mathcal{G}}e^{\boldsymbol{G_0}x}\,dx$. Hence

$$\underline{z}\,\boldsymbol{Y_{BB}}\,\underline{z}^T=\int_0^\infty\underline{z}e^{\boldsymbol{G_0}^Tx}\underline{\alpha}_{\mathcal{G}}^T\cdot\underline{\alpha}_{\mathcal{G}}e^{\boldsymbol{G_0}x}\underline{z}^T\,dx=\int_0^\infty\left(\underline{\alpha}_{\mathcal{G}}e^{\boldsymbol{G_0}x}\underline{z}^T\right)^2\,dx, \tag{113}$$

which can not be negative, furthermore, apart from a finite number of x values $\underline{\alpha}_{\mathcal{G}}e^{\boldsymbol{G_0}x}\underline{z}^T$ can not be zero either. Thus, the integral is always strictly positive.

The positive definiteness of matrix $\boldsymbol{W_{BB}}$ can be proven similarly. □

Being able to formalize the optimization of $\mathcal{D}_2\{\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}}\}$ as a quadratic programming problem means that obtaining the optimal matrix $\boldsymbol{G_1}$ is efficient: it is fast, and there is a single optimum which is always found.

If we intend to take higher lag joint density differences also into account ($k>2$) and/or there are multiple arrival types ($K>1$), the objective function is not quadratic. However, our numerical experience indicates that the built-in non-linear optimization tool in MATLAB (fmincon) is able to return the solution quickly, independent of the initial point. We have a strong suspicion that the returned solution is the global optimum, however we can not prove the convexity of the objective function formally.

Algorithm 5 Algorithm for fitting the joint density up to lag k

procedure FIT-JOINT-DENSITY($\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}$)
  $\hat{m}_1, \hat{m}_2, \cdots \leftarrow$ marginal moments of the input MRAP
  $\mathcal{P} \leftarrow$ hyper-FEB solutions of moment matching based on $\hat{m}_1, \hat{m}_2, \ldots$
  $\{\boldsymbol{D_0^*}, \ldots, \boldsymbol{D_K^*}\} \leftarrow \emptyset$
  for each solution $(\underline{\sigma}, \boldsymbol{S})$ in $\mathcal{P}$ **do**
    $\boldsymbol{G_0} \leftarrow \boldsymbol{S}$
    $\underline{\alpha}_{\mathcal{G}} \leftarrow \underline{\sigma}$
    Solve $(\boldsymbol{G_1}, \ldots, \boldsymbol{G_K}) = \underset{\boldsymbol{G_1}, \ldots, \boldsymbol{G_K}}{\arg\min}\, \mathcal{D}_k((\boldsymbol{D_0}, \ldots, \boldsymbol{D_K}), (\boldsymbol{G_0}, \ldots, \boldsymbol{G_K}))$,
    if solution is better than $\{\boldsymbol{D_0^*}, \ldots, \boldsymbol{D_K^*}\}$ **then**
      $\{\boldsymbol{D_0^*}, \ldots, \boldsymbol{D_K^*}\} \leftarrow \{\boldsymbol{G_0}, \ldots, \boldsymbol{G_K}\}$
    end if
  end for
  return $\{\boldsymbol{D_0^*}, \ldots, \boldsymbol{D_K^*}\}$
end procedure

The pseudo-code of the procedure to transform a non-Markovian or invalid MRAP to a valid Markovian representation is presented by Algorithm 5.

In the first numerical example 7 marginal moments and 9 lag-1 joint moments are extracted from the BC trace (also used in Section 3.3.3) to create a RAP of order 4 with the moment matching method presented in Section 3.2.2. The obtained matrices are:

$$\boldsymbol{G_0} = \begin{bmatrix} -0.579 & -0.402 & -0.364 & -0.348 \\ -0.368 & -0.205 & -0.315 & -0.36 \\ 1.32 & -0.845 & 0.701 & 1.13 \\ -1.7 & 0.3 & -1.14 & -1.52 \end{bmatrix}, \boldsymbol{G_1} = \begin{bmatrix} 0.576 & 0.262 & 0.41 & 0.446 \\ 0.168 & 0.501 & 0.313 & 0.266 \\ 0.29 & -1.69 & -0.598 & -0.302 \\ 0.292 & 1.94 & 1.03 & 0.786 \end{bmatrix}.$$

The RAP characterized by $\boldsymbol{\mathcal{G}} = (\boldsymbol{G_0}, \boldsymbol{G_1})$, however, apart of being non-Markovian, is not a valid stochastic process as the joint density given by (52) is negative since $f_2(0.5, 8) = -0.000357$. This invalid RAP is the target of our approximation in this section.

Let us now construct a MAP $\boldsymbol{\mathcal{F}^{(1)}} = (\boldsymbol{D_0^{(1)}}, \boldsymbol{D_1^{(1)}})$ which minimizes the squared distance of the lag-1 joint density with $\boldsymbol{\mathcal{G}}$. The distribution of the inter-arrival times, characterized by $\underline{\alpha}_{\mathcal{F}}, \boldsymbol{D_0^{(1)}}$ are obtained by the moment matching method and matrix $\boldsymbol{D_1^{(1)}}$ has been determined by the quadratic program provided by Theorem 15. The matrices of the MAP are

$$\boldsymbol{D_0^{(1)}} = \begin{bmatrix} -0.074 & 0 & 0 & 0 & 0 \\ 0 & -0.27 & 0.27 & 0 & 0 \\ 0 & 0 & -0.27 & 0.27 & 0 \\ 0 & 0 & 0 & -0.27 & 0 \\ 0 & 0 & 0 & 0 & -1.2 \end{bmatrix}, \boldsymbol{D_1^{(1)}} = \begin{bmatrix} 0.0065 & 0.024 & 0 & 5.5{\cdot}10^{-8} & 0.044 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.017 & 0.086 & 0 & 0 & 0.17 \\ 0 & 0.012 & 0 & 0 & 1.2 \end{bmatrix},$$

and the squared distance in the lag-1 joint pdf is $\mathcal{D}_2\{\boldsymbol{\mathcal{F}^{(1)}}, \boldsymbol{\mathcal{G}}\} = 0.000105$. The quadratic program has been solved in less than a second. Next, we repeat the same procedure, but instead of focusing on the lag-1 distance, we optimize on the squared distance of the joint pdf up to lag-10. This can not be formalized as a quadratic program any more, but the optimization

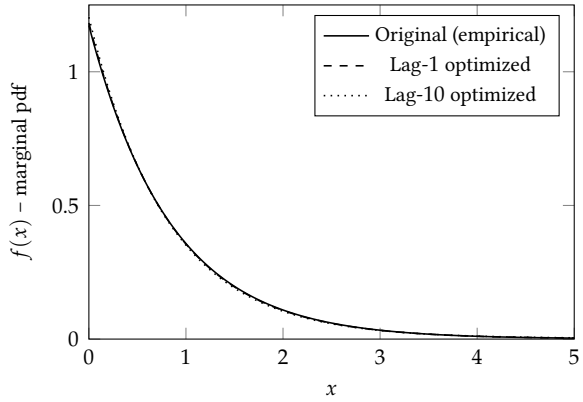


Figure 19.: Comparison of the density functions of the marginal distribution

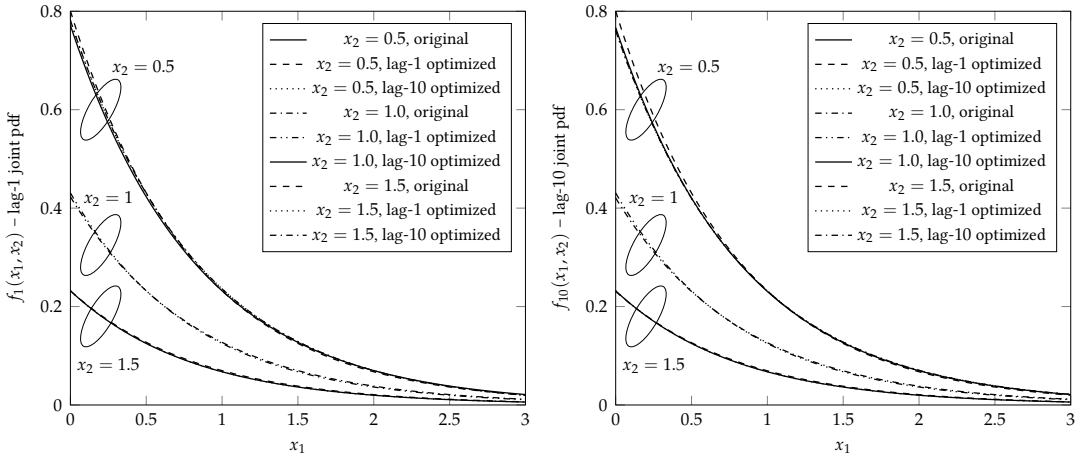


Figure 20.: Comparison of the lag-1 and lag-10 joint density functions

is still fast, lasting only 1-2 seconds. In this case the hyper-exponential marginal distribution provided the best results ($\mathcal{D}_{11}\{\boldsymbol{\mathcal{F}}^{(10)}, \boldsymbol{\mathcal{B}}\} = 4.37 \cdot 10^{-5}$). The matrices are

$$\boldsymbol{D_0^{(10)}} = \begin{bmatrix} -0.0519 & 0 & 0 \\ 0 & -0.151 & 0 \\ 0 & 0 & -1.24 \end{bmatrix}, \quad \boldsymbol{D_1^{(10)}} = \begin{bmatrix} 10^{-6} & 0.0519 & 10^{-6} \\ 10^{-6} & 0.151 & 0.000465 \\ 0.000129 & 10^{-6} & 1.24 \end{bmatrix}.$$

To evaluate the quality of the approximation Figure 19 compares the marginal density functions of $\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{F}}^{(1)}$ and $\boldsymbol{\mathcal{F}}^{(10)}$. The plots are hiding each other, the approximation is very accurate. To demonstrate that the lag-1 and the lag-10 joint densities are also accurate, Figure 20 depicts them at $x_2 = 0.5, 1$ and 1.5.

Part II

QUEUES

dc_1412_17

4

SKIP-FREE PROCESSES

This chapter introduces two important tools in Markovian modeling: the quasi birth-death processes and the Markovian fluid models. The subsequent chapters rely on these two tools heavily.

- In case of the MAP/MAP/1 queues in Chapter 5, quasi birth-death processes (QBDs) are used for the analysis of the queue length and the departure processes, while Markovian fluid models (MFMs) are used for the analysis of the sojourn time of customers.
- In case of MMAP[K]/PH[K]/1-FCFS queues in Chapter 6 all performance metrics are derived using the age process, which is transformed to a MFM.
- The analysis of the MMAP[K]/PH[K]/1 priority queue in Chapter 7 relies on the workload process, which is transformed to a MFM to make the solution numerically efficient.

4.1 QUASI BIRTH-DEATH PROCESSES

4.1.1 *Simple birth-death processes*

A CTMC $\{\mathcal{L}(t), t \geq 0\}$ over state space $\mathcal{N} = \{0, 1, 2, \dots\}$ is called an infinite *birth-death process* when it has only two kinds of state transitions: forward ("birth") and backward ("death") transitions moving the CTMC to the next and to the previous state, respectively. The generator matrix $\boldsymbol{Q}$ is tri-diagonal in this case, thus

$$\boldsymbol{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & & \\ \mu_1 & -\lambda_1 - \mu_1 & \lambda_1 & \\ & \mu_2 & -\lambda_2 - \mu_2 & \lambda_2 \\ & \ddots & \ddots & \ddots \end{bmatrix}. \tag{114}$$

The Markov chain $\mathcal{L}(t)$ is called a *homogeneous* birth-death process when all forward rates and all backward rates are identical, thus $\lambda_i = \lambda$, $\forall i \geq 0$ and $\mu_i = \mu$, $\forall i > 0$. Homogeneous birth-death processes have a great practical importance. Basic queueing systems, like the M/M/1-FCFS queue (used to model simple buffers with single server and first-come-first-served service) and the M/M/∞-PS queue (used to model infinite server systems where the total service capacity is fixed and shared among the servers) can both be modeled by such a Markov chain ([53]). An attractive feature of homogeneous birth-death processes is that their analysis is simple. Assuming stability (hence $\lambda < \mu$) the stationary distribution is geometric, given by

$$\pi_k = \lim_{t \to \infty} P(\mathcal{L}(t) = k) = (1 - \rho)\rho^k, \quad k = 0, \dots, \infty, \tag{115}$$

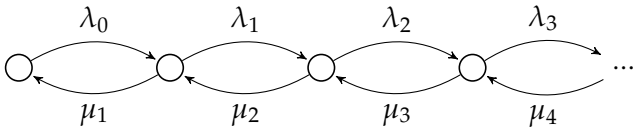


Figure 21.: A simple birth-death process

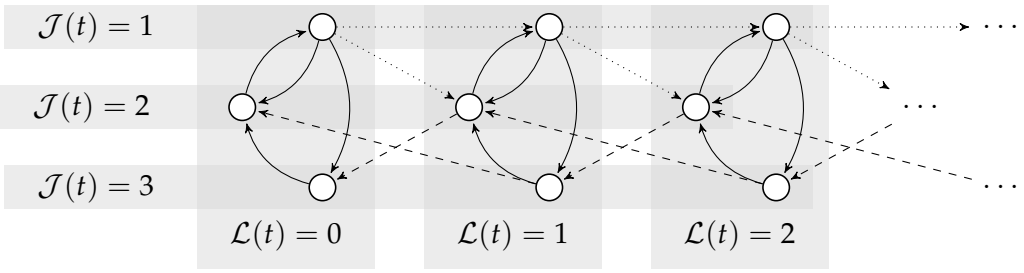


Figure 22.: A quasi birth-death process

where $\rho = \lambda/\mu$.

4.1.2 *Quasi birth-death processes*

In simple birth-death processes the sojourn time of the states is exponentially distributed (with parameter $\lambda_i + \mu_i$ in state i), and the distribution of the next state does not depend on the time spent in the previous state (it moves to the next state with probability $\lambda_i/(\lambda_i + \mu_i)$ and to the previous one with probability $\mu_i/(\lambda_i + \mu_i)$).

To model more general systems where these two properties do not hold, they have introduced *quasi birth-death processes* (QBDs, [65]). QBDs are two-dimensional continuous time Markov chains $\{\mathcal{L}(t), \mathcal{J}(t), t \geq 0\}$, where $\mathcal{L}(t)$ is referred to as the *level process* and $\mathcal{J}(t)$ is called the *phase process*. In a QBD only state transitions to the next and to the previous levels are allowed (see Figures 22 and 23). Due to this behavior QBDs are *skip-free to both to the left and to the right*.

In the sequel we restrict our attention to the case when the level process is infinite, $\mathcal{L}(t) \in [0, \infty]$ and the phase process is finite $\mathcal{J}(t) \in [1, N]$. As a consequence of the skip-free property the generator matrix is block tri-diagonal with the appropriate state ordering, thus

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F}_0 & & & \\ \boldsymbol{B}_1 & \boldsymbol{L}_1 & \boldsymbol{F}_1 & & \\ & \boldsymbol{B}_2 & \boldsymbol{L}_2 & \boldsymbol{F}_2 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{116}$$

where blocks $\boldsymbol{F}_i, \boldsymbol{L}_i$ and $\boldsymbol{B}_i$ denote the matrices containing the rates of level forward, local, and level backward transitions, respectively. If these matrix blocks are the same at each level,

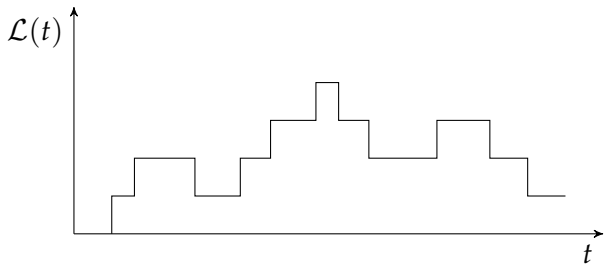


Figure 23.: The trajectory of a QBD process

thus $\boldsymbol{F}_i = \boldsymbol{F}, \forall i \geq 0$ and $\boldsymbol{L}_i = \boldsymbol{L}, \boldsymbol{B}_i = \boldsymbol{B}, \forall i > 0$ then the process is called a *homogeneous QBD*. In the forthcoming chapters homogeneous QBDs with generator

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F} & & & \\ \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & & \\ & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{117}$$

are applied many times for various modeling purposes.

4.1.3 *Stationary solution of QBDs*

In the regular part of the state space (above level 0) the generator of the CTMC characterizing the phase process is given by $\boldsymbol{A} = \boldsymbol{F} + \boldsymbol{L} + \boldsymbol{B}$. Assuming that $\boldsymbol{A}$ is irreducible, the stationary phase distribution vector $\underline{\nu}$ is the solution of $\underline{\nu}\boldsymbol{A} = \underline{0}, \underline{\nu}\mathbb{1} = 1$. The stationary drift of the QBD, which is the difference between the mean level forward and level backward transition rates, is given by

$$d = \underline{\nu}\boldsymbol{F}\mathbb{1} - \underline{\nu}\boldsymbol{B}\mathbb{1}. \tag{118}$$

The stability condition for QBDs is $d < 0$ (see [65]).

Let us denote the stationary distribution of the QBD by $\pi_{i,j} = \lim_{t\to\infty} P(\mathcal{L}(t) = i, \mathcal{J}(t) = j)$, and introduce vectors $\underline{\pi}$ and $\underline{\pi}_i$ such that $\underline{\pi}_i = \{\pi_{i,j}, j = 1, \ldots, N\}$ and $\underline{\pi} = \{\underline{\pi}_i, i = 0, \ldots, \infty\}$. Due to the block structure of the generator (117) and the structure of vector $\underline{\pi}$ the stationary equation $\underline{\pi}\boldsymbol{Q} = \underline{0}, \underline{\pi}\mathbb{1} = 1$ translates to

$$\underline{\pi}_0\boldsymbol{L}_0 + \underline{\pi}_1\boldsymbol{B} = \underline{0}, \tag{119}$$

$$\underline{\pi}_{i-1}\boldsymbol{F} + \underline{\pi}_i\boldsymbol{L} + \underline{\pi}_{i+1}\boldsymbol{B} = \underline{0}, \quad \text{for } i = 1, \ldots, \infty, \tag{120}$$

$$\sum_{i=0}^{\infty} \underline{\pi}_i\mathbb{1} = 1. \tag{121}$$

From [65] it is known that the solution of such matrix recursions is matrix-geometric, thus

$$\underline{\pi}_i = \underline{\pi}_0\boldsymbol{R}^i, \tag{122}$$

which has two parameters: matrix $\boldsymbol{R}$ and vector $\underline{\pi}_0$. Notice how similar the matrix-geometric solution (122) is to the geometric solution of simple birth-death systems given by (115).

To obtain matrix $\boldsymbol{R}$, the solution (122) is inserted to the stationary equation (120), from which it follows that $\boldsymbol{R}$ has to satisfy the matrix-quadratic equation

$$\boldsymbol{F} + \boldsymbol{R}\boldsymbol{L} + \boldsymbol{R}^2\boldsymbol{B} = \boldsymbol{0}. \tag{123}$$

This matrix quadratic equation has several different solutions. In [65] it is proven that the $\boldsymbol{R}$ we need is the minimal non-negative solution of this equation. The elements of matrix $\boldsymbol{R}$ have a stochastic interpretation as well: $(\boldsymbol{R})_{i,j}$ is the mean time spent at level $n+1$ and phase j, staring at level n and phase i, before returning to level $n-1$. Observe that due to the spatial homogeneity of the system this quantity does not depend on level n.

Vector $\underline{\pi}_0$ can be derived from the boundary equation (119). Making use of $\underline{\pi}_1 = \underline{\pi}_0\boldsymbol{R}$ (119) becomes

$$\underline{\pi}_0(\boldsymbol{L_0} + \boldsymbol{R}\boldsymbol{F}) = \boldsymbol{0}. \tag{124}$$

This linear system is, however, rank deficient. To make it full rank, the normalization condition

$$1 = \underline{\pi}_0 \sum_{i=0}^{\infty} \boldsymbol{R}^i \mathbb{1} = \underline{\pi}_0 (\boldsymbol{I} - \boldsymbol{R})^{-1} \mathbb{1} \tag{125}$$

must be added as well.

In the stationary analysis of QBDs the most challenging computational step is the solution of the matrix quadratic equation providing $\boldsymbol{R}$. Several approaches have been published, including the spectral solution in [63], the invariant subspace method in [2] and an iterative solution in [65].

The simplest algorithm for matrix $\boldsymbol{R}$ is a functional iteration based on simple substitutions. According to this method matrix $\boldsymbol{R}_0$ is set to $\boldsymbol{0}$ initially, and the steps

$$\boldsymbol{R}_n = (-\boldsymbol{L})^{-1} \left(\boldsymbol{F} + \boldsymbol{B}\boldsymbol{R}_{n-1}^2\right) \tag{126}$$

are repeated iteratively as long as the relative change in the matrix elements elements gets negligible. A drawback of this algorithm is that it suffers from linear convergence, thus the error term vanishes at a linear speed with the number of iterations taken. More advanced algorithms with quadratic convergence have been introduced as well, with the most popular one being the cyclic reduction (CR) algorithm ([12]). The iterations of the CR algorithm are more complex, but the solution is found in much fewer iterations than in case of the simple substitution method.

The CR algorithm is remarkably robust (as well as some more recent algorithms), it allows to solve QBDs with many thousands of phases in just a couple of seconds without numerical issues. In the rest of the dissertation, every time the solution of a QBD is required, the CR algorithm will be used to obtain matrix $\boldsymbol{R}$.

4.1.4 *Busy period analysis*

Matrix $\boldsymbol{R}$ plays a fundamental role in the analysis of QBDs, but there are some other important matrix quantities related to QBDs as well. One of those matrices is traditionally denoted by $\boldsymbol{G}$, and will be used several times in the upcoming chapters.

Matrix $\boldsymbol{G}$ contains the phase transition probabilities over the busy period. If random variable $\mathcal{B}$ denotes the duration of the busy period, thus $\mathcal{B} = \inf(t > 0, \mathcal{L}(t) = 0)$, the formal definition of the elements are

$$(\boldsymbol{G})_{i,j} = P(\mathcal{J}(\mathcal{B}) = j | \mathcal{L}(0) = 0, \mathcal{J}(0) = j). \tag{127}$$

Observe that due to the spatial homogeneity the scope of matrix $\boldsymbol{G}$ is not restricted to level 0 only. The entries $(\boldsymbol{G})_{i,j}$ also provide the probability that starting from any level n and phase i the phase will be j at the first return to level $n-1$.

The matrix equation for $\boldsymbol{G}$ can be derived along this interpretation. The QBD can either leave level n by moving right to level $n-1$, marking the end of the "time till first return" period, the corresponding phase transitions are given by $(-\boldsymbol{L})^{-1}\boldsymbol{B}$. Or it can leave level n by moving to the next level, which is accompanied by phase transitions $(-\boldsymbol{L})^{-1}\boldsymbol{F}$. In this case, however, level $n-1$ can be reached only by two subsequent "first return" periods. Hence, the equation for $\boldsymbol{G}$ is $\boldsymbol{G} = (-\boldsymbol{L})^{-1}\boldsymbol{B} + (-\boldsymbol{L})^{-1}\boldsymbol{F}\boldsymbol{G}^2$, that, pre-multiplying by $\boldsymbol{L}$ leads to

$$\boldsymbol{B} + \boldsymbol{L}\boldsymbol{G} + \boldsymbol{F}\boldsymbol{G}^2 = \boldsymbol{0}, \tag{128}$$

which is very similar to (123).

The algorithms developed for matrix $\boldsymbol{R}$ can be adopted to compute matrix $\boldsymbol{G}$. Actually, there are formulas available to express any of these matrices with the other one.

4.2 MARKOVIAN FLUID MODELS

QBDs are popular in the performance evaluation of a wide range of engineering systems, due to their algorithmic tractability which is the consequence of the skip-free behavior.

MFMs are the continuous counterparts of QBDs, where the level process $\mathcal{L}(t)$ is continuous. MFMs, similar to QBDs, are skip free both to the left and to the right. There are significant similarities in the solution methods for QBDs and MFMs as well, which have been discovered only in the last two decades (starting with [71]), and the effort towards adapting the research results available for QBDs to MFMs is still ongoing. The appearance of efficient stationary analysis algorithms for MFMs enabled the efficient solution of many multi-class queues as demonstrated in the next three chapters.

4.2.1 *Model definition*

MFMs (also known as Markovian fluid flows) are characterized by a two-dimensional Markov process $\{\mathcal{L}(t), \mathcal{J}(t), t > 0\}$, where $\mathcal{L}(t)$ represents the fluid level and $\mathcal{J}(t)$ is the underlying CTMC with state space $\mathcal{N}$ of size $|\mathcal{N}| = N$ and generator matrix $\boldsymbol{Q}$ that modulates the rate at which fluid is accumulated in the fluid buffer.

The rate at which the level of the buffer changes in state i of the background process is denoted by $r_i \in \mathbb{R}$. The diagonal matrix $\boldsymbol{R}$ is composed by fluid rates $r_i, i = 1, \ldots, N$. Formally, the behavior of the fluid buffer is as follows,

$$\frac{d}{dt}\mathcal{L}(t) = \begin{cases} r_{\mathcal{J}(t)}, & \text{if } \mathcal{L}(t) > 0, \\ \max\{0, r_{\mathcal{J}(t)}\}, & \text{if } \mathcal{L}(t) = 0. \end{cases} \tag{129}$$

In Figure 24 the solid line depicts a trajectory of the fluid level, and the dotted line represents the current state of the background process (the rates are $r_1 = r_2 = 1, r_3 = -1$).

4.2.2 *Stationary solution*

If the stationary distribution of the background CTMC is denoted by $\underline{\nu}$, the mean fluid drift is given by

$$d = \underline{\nu}\boldsymbol{R}\mathbb{1}. \tag{130}$$

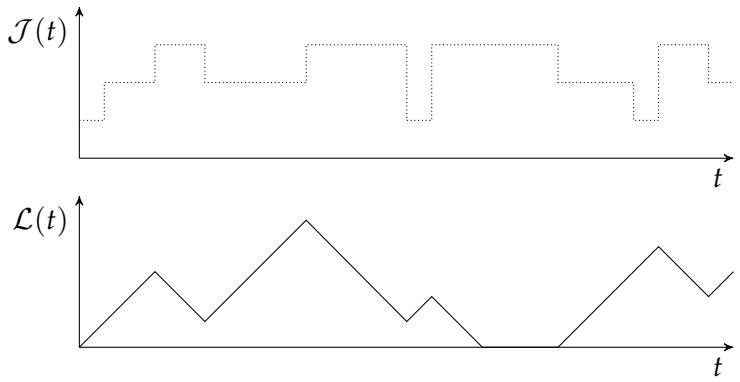


Figure 24.: The trajectory of a Markovian fluid model

The MFM is stable if $d < 0$ holds. In the sequel we are considering stable systems only.

Let us denote the row vector of the stationary distribution of the fluid level for $x > 0$ by $\underline{\pi}(x) = \{\pi_i(x), i \in \mathcal{N}\}$ with $\pi_i(x) = \lim_{t\to\infty} \lim_{\Delta\to 0}(1/\Delta)P(\mathcal{L}(t) \in (x, x+\Delta), \mathcal{J}(t) = i)$, and the row vector of the stationary probabilities of empty buffer by $\underline{p} = \{p_i, i \in \mathcal{N}\}$ with $p_i = \lim_{t\to\infty} P(\mathcal{J}(t) = i, \mathcal{L}(t) = 0)$.

Vectors $\underline{\pi}(x)$ and $\underline{p}$ are the solutions of the matrix differential equation [54]

$$\frac{d}{dx}\underline{\pi}(x)\boldsymbol{R} = \underline{\pi}(x)\boldsymbol{Q}, \tag{131}$$

with boundary conditions

$$\underline{\pi}(0)\boldsymbol{R} = \underline{p}\,\boldsymbol{Q}, \quad p_i = 0, \ \forall i : r_i > 0, \tag{132}$$

and normalizing condition

$$\underline{p}\mathbb{1} + \int_0^\infty \underline{\pi}(x)\mathbb{1}\,dx = 1. \tag{133}$$

These systems of equations are the continuous counterparts of (120), (119) and (121), respectively.

In the recent decades it has been recognized that the matrix-analytic approach allowing the efficient analysis of QBDs can be applied to MFMs as well, making it possible to solve fluid models with a large number of states (up to several thousand) in a numerically stable way (see [71],[76]). MFMs where $|r_i| = 1, \forall i \in \mathcal{N}$ are referred to as *canonical fluid models*, and are especially simple to analyze. Here we summarize the main steps of the analysis of canonical fluid models. We assume that the state space is partitioned according to the sign of the associated fluid rates to two sets $\mathcal{N}_+ = \{i \in \mathcal{N}, r_i = 1\}$ and $\mathcal{N}_- = \{i \in \mathcal{N}, r_i = -1\}$ ($N_+ = |\mathcal{N}_+|, N_- = |\mathcal{N}_-|$) as

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{++} & \boldsymbol{Q}_{+-} \\ \boldsymbol{Q}_{-+} & \boldsymbol{Q}_{--} \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{I} & \\ & -\boldsymbol{I} \end{bmatrix}. \tag{134}$$

The analysis is based on two fundamental matrices, matrix $\boldsymbol{\Psi}$ and $\boldsymbol{K}$ (see [71]). Matrix $\boldsymbol{\Psi}$ has a simple probabilistic interpretation: entry $(\boldsymbol{\Psi})_{i,j}$, $i \in \mathcal{N}_+, j \in \mathcal{N}_-$ is the probability that the background process is in state j when the fluid level returns to 0 given that it was in state i when the busy period (a non-empty period of the fluid queue) was initiated. Matrix $\boldsymbol{\Psi}$ is the solution to the non-symmetric algebraic Riccati equation (NARE)

$$\boldsymbol{\Psi}\boldsymbol{Q}_{-+}\boldsymbol{\Psi} + \boldsymbol{\Psi}\boldsymbol{Q}_{--} + \boldsymbol{Q}_{++}\boldsymbol{\Psi} + \boldsymbol{Q}_{+-} = \boldsymbol{0}. \tag{135}$$

The efficient solution of this NARE is as crucial for the analysis of MFMs as the solution of the matrix-quadratic equation is for the analysis of QBDs. The state-of-the art algorithms are the Structure-preserving Doubling Algorithm (SDA) [36] and the Alternating-Directional Doubling Algorithm (ADDA) [86], both of these algorithms have quadratic convergence speed.

Matrix $\boldsymbol{K}$ has an important role as well. Entry i, j of matrix $e^{\boldsymbol{K}x}$ is the expected number of crossings of fluid level x in phase $j \in \mathcal{N}_+$ starting from level 0 and phase $i \in \mathcal{N}_+$, before returning to level 0. If the mean fluid rate is negative, all eigenvalues of matrix $\boldsymbol{K}$ have negative real parts (thus it is full rank and invertible) and can be expressed from $\boldsymbol{\Psi}$ as

$$\boldsymbol{K} = \boldsymbol{Q}_{++} + \boldsymbol{\Psi}\boldsymbol{Q}_{-+}. \tag{136}$$

Based on these matrices the stationary fluid level density vector $\underline{\pi}$ and the stationary probability vector of the idle buffer $\underline{p}$ can be computed by the following theorem.

Theorem 17. *If the drift of the queue is negative, vector $\underline{\pi}(x)$ is given by*

$$\underline{\pi}(x) = \begin{bmatrix} \underline{\pi}_+(x) & \underline{\pi}_-(x) \end{bmatrix} = \underline{p}_-\boldsymbol{Q}_{-+}e^{\boldsymbol{K}x}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi} \end{bmatrix}, \quad x \geq 0, \tag{137}$$

and the probability mass vector p is equal to

$$\underline{p} = \begin{bmatrix} 0 & \underline{p}_- \end{bmatrix}, \tag{138}$$

where $\underline{p}_-$ is the solution to the set of linear equations

$$\underline{p}_-(\boldsymbol{Q}_{--} + \boldsymbol{Q}_{-+}\boldsymbol{\Psi}) = \underline{0}, \tag{139}$$

$$\underline{p}_-\boldsymbol{Q}_{-+}(-\boldsymbol{K})^{-1}\begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi} \end{bmatrix}\mathbb{1} + \underline{p}_-\mathbb{1} = 1. \tag{140}$$

Proof. The theorem is based on [76], especially on Theorem 2.2. □

Let us now investigate the similarities between QBDs and MFMs. While the stationary distribution is matrix-geometric for QBDs (122), it is matrix-exponential for MFMs (137). Matrices $\boldsymbol{K}$ and $\boldsymbol{\Psi}$ in MFMs play the same role as matrices $\boldsymbol{R}$ and $\boldsymbol{G}$ in QBDs. Furthermore, both for $\boldsymbol{R}$ (123) and $\boldsymbol{K}$ (136) the iterative solutions of some kind of quadratic equations are necessary.

4.2.3 *Busy period analysis of Markovian fluid models*

In this section we briefly summarize the most essential results of [1] and [72] on the busy period analysis of fluid models, they will be necessary in Chapter 7 for the solution of priority queues.

As mentioned above, $\boldsymbol{\Psi}$ is the phase transition probability matrix between the beginning and the end of the busy period. If the duration of the busy period is also of interest, we can introduce matrix $\boldsymbol{\Psi}(t)$, the time dependent counterpart of $\boldsymbol{\Psi}$. Entry $(\boldsymbol{\Psi}(t))_{i,j}$, $i \in \mathcal{N}_+, j \in \mathcal{N}_-, t > 0$ is the joint probability that the duration of the busy period is less than t and the underlying Markov chain is in state j when the fluid level returns to 0 given that it was in state i when the busy period was initiated.

According to Theorem 1 of [72], the LST of $\boldsymbol{\Psi}(t)$, denoted by $\boldsymbol{\Psi}^*(s)$ satisfies the NARE

$$\boldsymbol{\Psi}^*(s)\boldsymbol{Q}_{-+}\boldsymbol{\Psi}^*(s) + \boldsymbol{\Psi}^*(s)\boldsymbol{Q}_{--} + \boldsymbol{Q}_{++}\boldsymbol{\Psi}^*(s) + \boldsymbol{Q}_{+-} = 2s\boldsymbol{\Psi}^*(s). \tag{141}$$

(Note that setting $s \to 0$ gives back (135)).

Let the random variable $\mathcal{B}$ denote the length of the busy period of a canonical fluid queue characterized by matrix $\boldsymbol{Q}$ given that the state probability vector of the background CTMC is $\underline{\kappa} = \{\kappa_i, i = 1, \ldots, N_+\}$ when the busy period starts.

Theorem 18. *The LST of the busy period $f^*_{\mathcal{B}}(s) = E(e^{-s\mathcal{B}})$ is given by*

$$f^*_{\mathcal{B}}(s) = \underline{\kappa}\, \boldsymbol{\Psi}^*(s)\mathbb{1}. \tag{142}$$

Proof. The theorem follows from the probabilistic interpretation of $\boldsymbol{\Psi}(t)$. □

Theorem 19. *The kth moment of the busy period is given by*

$$E(\mathcal{B}^k) = \underline{\kappa}\,(-1)^k \boldsymbol{\Psi}^{(k)}\mathbb{1}, \tag{143}$$

where $\boldsymbol{\Psi}^{(0)} = \boldsymbol{\Psi}$ and matrices $\boldsymbol{\Psi}^{(k)}, k > 0$ are defined recursively as

$$\begin{aligned}(\boldsymbol{Q}_{++} + \boldsymbol{\Psi}\boldsymbol{Q}_{-+})\boldsymbol{\Psi}^{(k)} + \boldsymbol{\Psi}^{(k)}(\boldsymbol{Q}_{--} + \boldsymbol{Q}_{-+}\boldsymbol{\Psi}) \\ = 2k\boldsymbol{\Psi}^{(k-1)} - \sum_{i=1}^{k-1} \binom{k}{i} \boldsymbol{\Psi}^{(i)} \boldsymbol{Q}_{-+} \boldsymbol{\Psi}^{(k-i)}.\end{aligned} \tag{144}$$

Proof. (144) follows from routine derivations with $\boldsymbol{\Psi}^{(k)} = \frac{d^k}{ds^k}\boldsymbol{\Psi}^*(s)|_{s=0}$. □

Since (135) providing $\boldsymbol{\Psi}^{(0)}$ is a NARE and (144) providing $\boldsymbol{\Psi}^{(k)}, k > 0$ is a Sylvester equation, the LST of the busy period and the moments can be obtained in a numerically efficient way. The distribution function in time domain, $F_{\mathcal{B}}(t) = P(\mathcal{B} < t) = \underline{\kappa}\,\boldsymbol{\Psi}(t)\mathbb{1}$ is, however, more involved to calculate. One can rely on a generic numerical Laplace transform inversion procedure, but according to our experience they are not always reliable up to the machine precision, and need complex arithmetic. Instead, a simple and elegant procedure called *Erlangization* is available [72], according to which the order-n approximation $F^{(n)}_{\mathcal{B}}(t)$ is

$$F^{(n)}_{\mathcal{B}}(t) = \int_0^\infty f_{\mathcal{E}(n,n/t)}(u) \cdot F_{\mathcal{B}}(u)\, du, \tag{145}$$

where $f_{\mathcal{E}(n,n/t)}(u)$ is the density of an order-n Erlang distribution with rate parameter $\nu = n/t$ and we have that $F^{(n)}_{\mathcal{B}}(t) \to F_{\mathcal{B}}(t)$ as $n \to \infty$. $F^{(n)}_{\mathcal{B}}(t)$ is basically the probability that the busy period is shorter than an Erlang(n, ν) variable.

Specifically for the busy period analysis $F^{(n)}_{\mathcal{B}}(t)$ can be obtained according to the next theorem.

Theorem 20. *([72], Theorem 4) The order-n approximation of the busy period distribution is*

$$F^{(n)}_{\mathcal{B}}(t) = \underline{\kappa} \sum_{k=0}^{n-1} \boldsymbol{\Psi}^{\nu}_k \mathbb{1}, \tag{146}$$

where matrices $\boldsymbol{\Psi}^{\nu}_k$ are defined recursively as

$$\begin{aligned}(\boldsymbol{Q}_{++} + \boldsymbol{\Psi}^{\nu}_0\boldsymbol{Q}_{-+} - \nu\boldsymbol{I})\boldsymbol{\Psi}^{\nu}_k + \boldsymbol{\Psi}^{\nu}_k(\boldsymbol{Q}_{--} + \boldsymbol{Q}_{-+}\boldsymbol{\Psi}^{\nu}_0 - \nu\boldsymbol{I}) \\ = -2\nu\boldsymbol{\Psi}^{\nu}_{k-1} - \sum_{i=1}^{k-1} \boldsymbol{\Psi}^{\nu}_i \boldsymbol{Q}_{-+} \boldsymbol{\Psi}^{\nu}_{k-i},\end{aligned} \tag{147}$$

for $k > 0$, and $\boldsymbol{\Psi}^{\nu}_0$ is the solution to the NARE

$$\boldsymbol{\Psi}^{\nu}_0\boldsymbol{Q}_{-+}\boldsymbol{\Psi}^{\nu}_0 + \boldsymbol{\Psi}^{\nu}_0(\boldsymbol{Q}_{--} - \nu\boldsymbol{I}) + (\boldsymbol{Q}_{++} - \nu\boldsymbol{I})\boldsymbol{\Psi}^{\nu}_0 + \boldsymbol{Q}_{+-} = \boldsymbol{0}. \tag{148}$$

For the detailed proof of the theorem, see [72]. The idea is to construct a special fluid model which counts the number of Exp(ν) events during the busy period. Matrix $\mathbf{\Psi}_k^{\nu}$ is the probability that k such events occur before the end of busy period (with the usual phase-transition probabilities being the entries of the matrix). If the number of Exp(ν) events is less than n, then the busy period is shorter than an Erlang(n, ν) variable, providing (145).

dc_1412_17

5

ANALYSIS OF THE MAP/MAP/1 QUEUE

The MAP/MAP/1 queue is a FCFS queue where the arrivals of customers are given by a MAP characterized by matrices $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$, and the service process is described by a MAP as well, given by matrices $\boldsymbol{S_0}$ and $\boldsymbol{S_1}$.

Thus, both the inter-arrival and the service times can be non-exponential and correlated. The majority of queueing models consider iid. service times, which is a reasonable assumption in many practical systems. For this specific queue, however, modeling the service process by a MAP makes the discussion simpler, and since the PH renewal processes are the sub-classes of MAPs, this choice makes the queueing model more general (it is described in Section 3.1.1 how to represent PH service times with a MAP).

The performance measures in this system, including the queue length and the sojourn time distributions, can be derived by standard methods and are known for a couple of decades. Nevertheless, we are going to present them in Sections 5.1 and 5.2 as they provide an introduction to the apparatus applied for the analysis of more complex systems described in the subsequent chapters. In Section 5.3.3 several approximations for the departure process are discussed, including the joint moment based one, which plays an essential role in the queueing network analysis approach proposed in Chapter 8.

5.1 ANALYSIS OF THE NUMBER OF CUSTOMERS IN THE SYSTEM

If we denote the mean arrival rate by $\lambda = \underline{\theta}_A \boldsymbol{D_1} \mathbb{1}$ and the mean service rate by $\mu = \underline{\theta}_S \boldsymbol{S_1} \mathbb{1}$ (with $\underline{\theta}_A$ and $\underline{\theta}_S$ being the stationary phase distributions of the arrival and service processes, respectively), the utilization of the queue is given by $\rho = \lambda/\mu$. In this chapter it is assumed that the system is stable, thus $\rho < 1$.

To analyze the number of customers in the system, a CTMC is created to model the *queue length process*. While this seems to be a natural choice, it will be clear in Chapter 6 and 7 that in many systems this approach is either too complex or infeasible.

The CTMC characterizing the queue length process needs to keep track of 1) $\mathcal{Y}(t)$, the number of customers in the system, 2) $\mathcal{J}_A(t)$, the phase of the arrival process, and 3) $\mathcal{J}_S(t)$, the phase of the service process. By introducing the finite state CTMC $\mathcal{J}(t)$ as the direct product of $\mathcal{J}_A(t)$ and $\mathcal{J}_S(t)$, the queue length process leads to a homogeneous QBD (see Section 4.1.2), where the generator has a block tri-diagonal structure given by (117). The matrix blocks of the generator are defined by the following Kronecker operations:

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{D_1} \otimes \boldsymbol{I}, \\ \boldsymbol{L} &= \boldsymbol{D_0} \oplus \boldsymbol{S_0}, \\ \boldsymbol{B} &= \boldsymbol{I} \otimes \boldsymbol{S_1}, \\ \boldsymbol{L_0} &= \boldsymbol{D_0} \otimes \boldsymbol{I}. \end{aligned} \tag{149}$$

The discussion of the basic properties of the Kronecker operations and how to use them to create the generator of independent, parallel Markov chains is provided in Appendix A.1. The meaning of the Kronecker summation giving matrix $\boldsymbol{L}$ is that the arrival and the service MAPs are evolving in parallel moving along their internal transitions. The Kronecker product providing $\boldsymbol{F}$ (and $\boldsymbol{B}$) are those transition rates of the arrival (and service) MAPs evolving in parallel that lead to arrivals (and services), respectively. Arrival events are accompanied by level forward and service events by level backward transitions in the QBD. At level 0, where the system is empty, the service MAP gets frozen.

If the arrival MAP has N_A phases and the service MAP has N_S phases, the cardinality of the superposed phase process (which is the size of the blocks of the generator) is $N = N_A \cdot N_S$.

Let us denote the joint stationary distribution of the number of customers in the system and the phase process by vector $\underline{y}_i = \{\lim_{t\to\infty} P(\mathcal{Y}(t) = i, \mathcal{J}(t) = j), j = 1, \ldots, N\}$. According to (122) $\underline{y}_i$ has a matrix-geometric distribution, thus

$$\underline{y}_i = \underline{y}_0 \boldsymbol{R}^i, \quad i \geq 0. \tag{150}$$

The details to obtain vector $\underline{y}_0$ and matrix $\boldsymbol{R}$ are described in Section 4.1.2.

The simplicity of the matrix-geometric distribution enables the efficient computation of many performance measures. E.g., the kth factorial moment of the number of customers in the system $E(\mathcal{Y}^k)$ can be computed as

$$E(\mathcal{Y}^k) = \sum_{i=0}^{\infty} i(i-1)\cdots(i-k+1)\, \underline{y}_0 \boldsymbol{R}^i \mathbb{1} = k! \underline{y}_0 \boldsymbol{R}^k (\boldsymbol{I} - \boldsymbol{R})^{-(k+1)} \mathbb{1}, \tag{151}$$

and the generating function (GF) $Y(z) = \sum_{i=0}^{\infty} z^i \underline{y}_i \mathbb{1}$ is

$$Y(z) = \sum_{i=0}^{\infty} z^i \, \underline{y}_0 \boldsymbol{R}^i \mathbb{1} = \underline{y}_0 (\boldsymbol{I} - z\boldsymbol{R})^{-1} \mathbb{1}. \tag{152}$$

For the departure process analysis the distribution of the number of customers embedded just after the departures, $\underline{x}_i$, will be necessary. This distribution is computed by "weighting" the elements of the stationary distribution with the transition rates leading to a departure, thus

$$\underline{x}_i = \frac{\underline{y}_{i+1}\boldsymbol{B}}{\sum_{k=1}^{\infty} \underline{y}_k \boldsymbol{B} \mathbb{1}} = \frac{1}{\lambda} \underline{y}_{i+1} \boldsymbol{B}, \quad i \geq 0. \tag{153}$$

The normalization constant, the denominator of (153) is the mean departure intensity which equals the mean arrival intensity λ when the queue is stable.

5.2 SOJOURN TIME ANALYSIS

In this section we introduce two modeling tools, the age process and the workload process, that play an important role in modern queueing theory. Here we use them only to derive the sojourn time distribution of the MAP/MAP/1 queue, but in the next two chapters they will be the fundamental tools of the analysis. The entire solution of the MMAP[K]/PH[K]/1 queue (including the analysis of the queue length, the sojourn time and the departure process, in Chapter 6) is based on the age process analysis. Similarly, the workload process analysis will be essential to solve priority queues in Chapter 7.

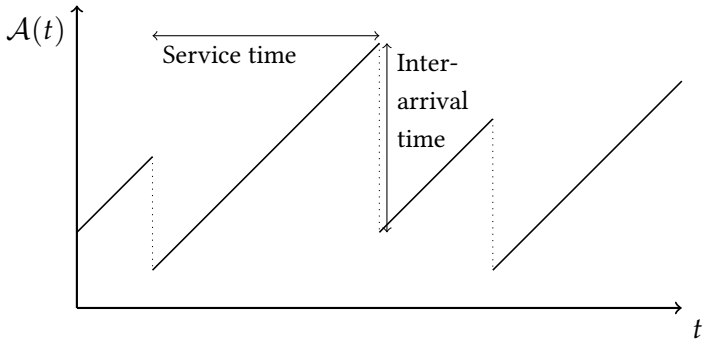


Figure 25.: Evolution of the age process

5.2.1 *Sojourn time analysis based on the age process*

The age process tracks the age of the customer under service [41]. The age of the customer in service increases linearly between the service instants and jumps downwards when the customer leaves the server (since the next customer is younger than the current one). The length of the downward jump is equal to the inter-arrival time between the customer leaving the system and the next customer who is about to enter the server (see Figure 25), unless the server becomes idle for a while.

There are various ways to deal with idle periods when using an age process: (i) negative values could be allowed for the age, the absolute value of which is the time until the server becomes busy again (ii) these idle periods could be skipped or (iii) the age is said to equal zero until the server becomes occupied again. We will make use of the latter approach and define the age process $\{\mathcal{A}(t), \mathcal{J}(t), t \geq 0\}$ as follows.

$\mathcal{A}(t)$ represents the age of the customer in service at time t, that is, $t - \mathcal{A}(t)$ represents the time of arrival of the customer in service, it is equal to zero in case the server is idle. If $\mathcal{A}(t) > 0$, then $\mathcal{J}(t)$ keeps track of (a) the current phase of the service MAP and (b) the phase of the arrival MAP at time $t - \mathcal{A}(t)$. If $\mathcal{A}(t) = 0$, then $\mathcal{J}(t)$ simply reflects the state of the arrival MAP at time t.

The direct analysis of the age process (exhibiting a skip free to the right behavior) seems hard due to the jumps. Observe that in our model the size of the jumps is not arbitrary, it is governed by the MAP generating the arrivals, which can be exploited to develop an efficient analysis method. We use the approach taken in [83], which was a generalization of [28]. The basic idea is to construct a canonical Markovian fluid model (Section 4.2) that is skip-free in both directions and to derive the steady state distribution of the age process from the steady state distribution of this fluid queue.

The background process of the fluid queue has two sets of phases according to the following considerations:

- The first set of phases corresponds to the evolution of the age process when $\mathcal{A}(t)$ increases. The fluid rates in this set of phases are equal to 1, since the age of the customer in the server increases according to a slope of one (see the solid line in Figure 26).
- The second set of phases is used whenever a customer leaves the system. In this case the age of the customer in service has to be decreased by the inter-arrival time between the customer leaving and the one who is about to enter the server.

  According to the definition of the age process, this decrease is immediate: it is a jump. As the inter-arrival time follows a MAP, the same amount of age decrease can be

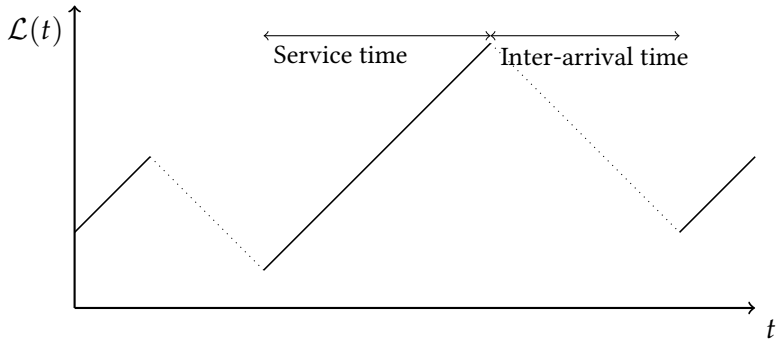


Figure 26.: Canonical MFM for the age process

achieved in an alternative way as well. Let us set the fluid rate to -1 and start the evolution of the MAP where it has been stopped before (the dotted line in Figure 26). When the MAP generates an arrival, the queue level representing the age process has been decreased appropriately, so the MAP can be frozen again and the fluid queue can go back to the first set of phases corresponding to the service periods.

To obtain the age process, the second set of phases (the dotted line in Figure 26) has to be censored out. Consequently, the age process will be analyzed using a canonical MFM: in the first set of phases the fluid rates are $+1$, in the second set of phases they are -1. The corresponding phase space partitions are denoted by $\mathcal{N}_+$ and $\mathcal{N}_-$, respectively, and their sizes are $|\mathcal{N}_+| = |\mathcal{N}_-| = N$. The first (second) set of phases will be referred to as positive (negative) phases in the sequel.

The generator matrix of the fluid model is

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{++} & \boldsymbol{Q}_{+-} \\ \boldsymbol{Q}_{-+} & \boldsymbol{Q}_{--} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S_0} \otimes \boldsymbol{I} & \boldsymbol{S_1} \otimes \boldsymbol{I} \\ \boldsymbol{I} \otimes \boldsymbol{D_1} & \boldsymbol{I} \otimes \boldsymbol{D_0} \end{bmatrix}. \tag{154}$$

In the positive phases (belonging to the intervals between jumps of the age process) the evolution of the background process is determined by the evolution of the service MAP, and the arrival MAP is frozen. After service completion there is a transition to the negative phases where the service MAP gets frozen, and the arrival MAP is resumed to decrease the age as much as younger the next customer is. When the arrival occurs, the background process goes back to the positive states and the service of the next customer starts.

The steady state pdf of the fluid model, $\underline{\pi}(x) = \underline{p}_- \boldsymbol{Q}_{-+} e^{\boldsymbol{K}x} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi} \end{bmatrix}$, is given by Theorem 17 in Section 4.2.2. The steady state distribution of the age process is obtained by censoring the results of the fluid queue on the positive phases. Let us denote the steady state joint density of the age process and the phase of the background process by $a_i(x)$ defined as

$$a_i(x) = \lim_{t\to\infty} \frac{d}{dx} P(\mathcal{A}(t) < x, \mathcal{J}(t) = i), \tag{155}$$

for $x \geq 0$ and $i = 1, \ldots, N$, and the corresponding vector by $\underline{a}(x) = \{a_i(x),\ i = 1, \ldots, N\}$.

Vector $\underline{a}(x)$ of the age process is obtained from the stationary solution of the fluid model as follows:

$$\underline{a}(x) = \frac{\underline{\pi}_+(x)}{\int_{y=0}^{\infty} \underline{\pi}_+(y)\mathbb{1}\, dy} = \frac{\underline{p}_-(\boldsymbol{I} \otimes \boldsymbol{D_1})e^{\boldsymbol{K}x}}{\underline{p}_-(\boldsymbol{I} \otimes \boldsymbol{D_1})(-\boldsymbol{K})^{-1}\mathbb{1}} = \underline{a}(0)e^{\boldsymbol{T}x}, \quad x \geq 0, \tag{156}$$

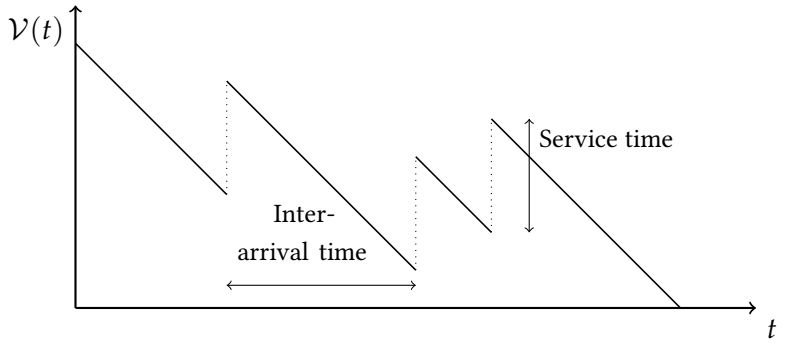


Figure 27.: The workload process of the MAP/MAP/1 queue

where in the last step we switched to the traditional representation of the age process. (The relation between the parameters of the MFM and the ones of the age process is $\boldsymbol{T} = \boldsymbol{K}$, $\underline{a}(0) = \frac{\underline{p}_-(\boldsymbol{I}\otimes\boldsymbol{D_1})}{\underline{p}_-(\boldsymbol{I}\otimes\boldsymbol{D_1})(-\boldsymbol{K})^{-1}\mathbb{1}}$).

Finally, the sojourn time of customers is equal to their age when they leave the system (this is the time instant right before the age process jumps downwards). Hence, the distribution of the sojourn time $\mathcal{T}$ is given by

$$F_{\mathcal{T}}(t) = P(\mathcal{T} < t) = \frac{\int_0^t \underline{a}(x)(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}\,dx}{\int_0^\infty \underline{a}(x)(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}\,dx} = 1 - \frac{\underline{a}(0)e^{\boldsymbol{T}t}(-\boldsymbol{T})^{-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}, \quad (157)$$

while the LST of the sojourn time $f^*_{\mathcal{T}}(s) = E(e^{-\mathcal{T}s})$ and kth moment $E(\mathcal{T}^k)$ are

$$f^*_{\mathcal{T}}(s) = \frac{\underline{a}(0)(s\boldsymbol{I}-\boldsymbol{T})^{-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}, \quad (158)$$

$$E(\mathcal{T}^k) = \frac{k!\,\underline{a}(0)(-\boldsymbol{T})^{-k-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\boldsymbol{S_1}\otimes\boldsymbol{I})\mathbb{1}}. \quad (159)$$

5.2.2 *Sojourn time analysis based on the workload process*

Besides the age process, an other useful tool to analyze the sojourn time of various queues is the workload process.

The workload process $\{\mathcal{V}(t), t \geq 0\}$ is the amount of work in the system at time t, thus the time needed to process all the customers in the queue if the arrival process was frozen. $\mathcal{V}(t)$ decreases by a slope of one between the arrival epochs (the server processes the workload), and jumps upwards at arrival epochs according to the service time requirement of the customer arrived (see Figure 27). Thus, $\mathcal{V}(t)$ is skip-free to the left. (As opposed to the age process, which is skip-free to the right).

As before, to characterize the sojourn time the two dimensional process $\{\mathcal{V}(t), \mathcal{J}(t), t \geq 0\}$ has to be studied. Here, $\mathcal{V}(t)$ is the workload of the queue at time t and $\mathcal{J}(t)$ is the phase process keeping track of the phase of the arrival MAP at time t and the phase of the service MAP right after the arrival of the last customer before t.

As the workload process has jumps, it is not straight forward to analyze its stationary behavior. However, the fact that the upward jumps are related to the service times and that the service times are determined by a MAP, it is possible to transform the skip-free to the left process to a MFM which is skip-free to both directions. The main idea is the same as in Section 5.2.1. At an arrival instant the amount of workload increment, which is given by a jump in

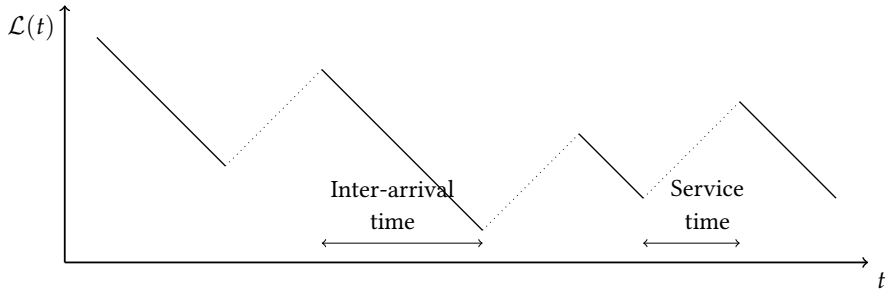


Figure 28.: The canonical MFM for the workload process

the original workload process, is accumulated in a time-continuous way in the transformed process.

For this transformation the state space is duplicated. In the first set of states (service periods) the workload decreases by a slope of one. When the MAP generates a customer arrival, the background Markov chain moves to the second set of states, which is responsible to increase the workload by the amount given by the service time requirement of the new customer. After increasing the workload the background process returns to the first set of states.

The transformed process is depicted by Figure 28. The generator of the associated canonical fluid model is

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{++} & \boldsymbol{Q}_{+-} \\ \boldsymbol{Q}_{-+} & \boldsymbol{Q}_{--} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_0 \otimes \boldsymbol{I} & \boldsymbol{S}_1 \otimes \boldsymbol{I} \\ \boldsymbol{I} \otimes \boldsymbol{D}_1 & \boldsymbol{I} \otimes \boldsymbol{D}_0 \end{bmatrix}, \tag{160}$$

which is exactly the same as the generator of the transformed age process (154).

In the transformed workload process the sojourn time of the customers is given by $\mathcal{V}(t)$ at $\mathcal{N}_+ \to \mathcal{N}_-$ transitions. These are the points where a new customer arrives and the workload has been incremented by its service time requirement. The time necessary to leave the system, hence the sojourn time of the customer, is given by $\mathcal{V}(t)$ at these points.

These particular points on the transformed workload process are the same as the ones on the transformed age process, hence the sojourn time distribution is the same as (157), we just arrived to it using a different modeling approach.

5.3 DEPARTURE PROCESS ANALYSIS

The exact departure process of a MAP/MAP/1 queue is a MAP with infinitely many phases, given by matrices $\boldsymbol{H}_0^{(\infty)}$ and $\boldsymbol{H}_1^{(\infty)}$. The phase process of this departure MAP is the QBD type Markov chain $\{\mathcal{Y}(t), \mathcal{J}(t)\}$ representing the joint evolution of the queue length and the phases of the arrival and service processes (defined in Section 5.1).

In this phase process the level backward transitions correspond to customer departures, which are the arrival events of the departure MAP, hence the corresponding transition rates are collected to matrix $\boldsymbol{H}_1^{(\infty)}$. All other transitions of the QBD (local and level forward tran-

sitions) are not accompanied by a departure event, hence their rates are comprised by matrix $\boldsymbol{H}_0^{(\infty)}$. Consequently, matrices $\boldsymbol{H}_0^{(\infty)}$ and $\boldsymbol{H}_1^{(\infty)}$ have the following structure:

$$\boldsymbol{H}_0^{(\infty)} = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} & \boldsymbol{0} & \dots \\ \vdots & \vdots & & \ddots & \ddots & \end{bmatrix}, \quad \boldsymbol{H}_1^{(\infty)} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \vdots & \vdots & \ddots & & \end{bmatrix}. \tag{161}$$

In [10] an approximation method is proposed for the departure process of MAP/PH/1 queues that is based on the truncation of the exact infinite MAP. Two further results appeared in the literature that are based on the same idea but can be applied to MAP/MAP/1 queues as well. Both of them truncate the infinite MAP at level n, but in different ways. The structure of the approximating departure process is the same

$$\boldsymbol{H}_0^{(n)} = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} & \boldsymbol{0} & \dots \\ \vdots & \vdots & & \ddots & \ddots & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \widehat{\boldsymbol{L}} \end{bmatrix}, \quad \boldsymbol{H}_1^{(n)} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots \\ \vdots & \vdots & \ddots & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \widehat{\boldsymbol{B}} & \widetilde{\boldsymbol{B}} \end{bmatrix}, \tag{162}$$

only the definition of the special matrix blocks, $\widehat{\boldsymbol{B}}$ and $\widetilde{\boldsymbol{B}}$ differs.

The next two subsection provide a short overview on these methods.

5.3.1 *Level probability based truncation method*

The basic idea of the truncation method of [74] is that all the levels $i \geq n$ of the exact model are merged into the last level (referred to as the clipping level) of the truncated model. All the level forward and local transitions of the infinite MAP correspond to local transitions in the truncated MAP, that gives $\widehat{\boldsymbol{L}} = \boldsymbol{F} + \boldsymbol{L}$.

However, in case of departures there are two cases when the truncated model is at the clipping level.

According to [74], the probability that the exact model is at level $i = n$ when the truncated process is at clipping level n is approximated using vector $\underline{y}_n$ and the probability that the exact model is at level $i > n$ when the truncated process is at clipping level n is approximated using vector $\underline{y}_n^+ = \sum_{k=n+1}^{\infty} \underline{y}_k$. Indeed, [74] approximates the probability that a departure of the truncated process at clipping level n and phase j moves the truncated process to level $n-1$ as $[\underline{y}_n]_j / ([\underline{y}_n^+]_j + [\underline{y}_n]_j)$, where $[\underline{y}_n]_j$ denotes the jth element of $\underline{y}_n$. Thus the related blocks of the truncated MAP are the following:

$$\begin{aligned} \widehat{\boldsymbol{B}} &= \mathrm{diag}\langle \underline{y}_n \rangle \mathrm{diag}\langle \underline{y}_n + \underline{y}_n^+ \rangle^{-1} \boldsymbol{B}, \\ \widetilde{\boldsymbol{B}} &= \mathrm{diag}\langle \underline{y}_n^+ \rangle \mathrm{diag}\langle \underline{y}_n + \underline{y}_n^+ \rangle^{-1} \boldsymbol{B}, \end{aligned} \tag{163}$$

where $\mathrm{diag}\langle vec \rangle$ denotes the diagonal matrix composed by the elements of vector vec. Since

$$\mathrm{diag}\langle \underline{y}_n \rangle \mathrm{diag}\langle \underline{y}_n + \underline{y}_n^+ \rangle^{-1} + \mathrm{diag}\langle \underline{y}_n^+ \rangle \mathrm{diag}\langle \underline{y}_n + \underline{y}_n^+ \rangle^{-1} = \boldsymbol{I}, \tag{164}$$

this definition ensures that $\widehat{\boldsymbol{B}} + \widetilde{\boldsymbol{B}} = \boldsymbol{B}$.

5.3.2 *ETAQA truncation method*

The method of [74] has been enhanced in [44]. The authors of [44] refer to their method as ETAQA truncation method. The blocks of the ETAQA truncated model are defined as

$$\begin{aligned} \widehat{\boldsymbol{L}} &= \boldsymbol{F} + \boldsymbol{L}, \\ \widehat{\boldsymbol{B}} &= \boldsymbol{B} - \boldsymbol{F}\boldsymbol{G}, \\ \widetilde{\boldsymbol{B}} &= \boldsymbol{F}\boldsymbol{G}, \end{aligned} \tag{165}$$

where matrix $\boldsymbol{G}$ is the phase transition probability matrix over the busy period of the QBD (see Section 4.1.4)

This construction (based on the idea of the ETAQA methodology) ensures that the steady state probabilities of the truncated process $\hat{\underline{y}}_k$ and of the exact model $\underline{y}_k$ are the same up to the clipping level, and for the clipping level $\hat{\underline{y}}_n = \sum_{k=n}^{\infty} \underline{y}_k$ holds. As a consequence, the inter-departure times and the lag correlations up to the truncation level n are preserved exactly.

5.3.3 *Joint moment based departure process approximation*

The joint moment based description of the departure process differs significantly from the truncation based approximation approaches described in Section 5.3.1 and 5.3.2. Those techniques construct an approximate departure process directly based on the behavior of the MAP/MAP/1 queue. Our proposed approach instead first computes dominant parameters of the departure process, namely the lag-1 joint moments of the consecutive inter-departure times, and then creates a MAP that realizes these parameters.

To describe the moments of the departure process we need the following notations. The row vector $\underline{d}_k^{(D)}$ ($\underline{s}_k^{(D)}$) denotes the phase distribution of the arrival (service) MAP after a departure which left k customers in the system. The ith elements of $\underline{d}_k^{(D)}$ and $\underline{s}_k^{(D)}$ are extracted from $\underline{x}_k$ (see (153)) as

$$[\underline{d}_k^{(D)}]_i = \underline{x}_k(\underline{e}_i^T \otimes \mathbb{1}) \text{ and } [\underline{s}_k^{(D)}]_i = \underline{x}_k(\mathbb{1} \otimes \underline{e}_i^T). \tag{166}$$

$\underline{e}_i$ is the row vector whose ith element is one and the others are zero. Matrix $\boldsymbol{U_0}$ of size $N_A \times N_S$ is composed by the elements of vector $\underline{x}_0$, such that $[\boldsymbol{U_0}]_{i,j} = [\underline{x}_0]_{(i-1)N_A+j}$. I.e., $[\boldsymbol{U_0}]_{i,j}$ is the probability that a departure leaves the MAP/MAP/1 queue empty, the phase of the arrival MAP is i and the phase of the departure MAP is j. Furthermore, $\underline{d}_0 = -\boldsymbol{D_0}\mathbb{1}$ and $\underline{s}_0 = -\boldsymbol{S_0}\mathbb{1}$ are the state dependent arrival and departure rates, respectively.

Theorem 21. *The stationary inter-departure time of a MAP/MAP/1 queue has a matrix exponential representation of order $N_A + N_S$ with initial vector $\underline{u}$, generator $\boldsymbol{M}$ and closing vector $\underline{w}$. That is, the cdf of the inter departure time distribution is $1 - \underline{u}e^{\boldsymbol{M}t}\underline{w}$, where*

$$\underline{u} = \begin{bmatrix} \underline{d}_0^T & \underline{s}_{1+}^{(D)} \end{bmatrix}, \tag{167}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{D_0}^T & \boldsymbol{U_0} \\ \boldsymbol{0} & \boldsymbol{S_0} \end{bmatrix}, \tag{168}$$

$$\underline{w} = \begin{bmatrix} (-\boldsymbol{D_0}^T)^{-1}\boldsymbol{U_0}\mathbb{1} \\ \mathbb{1} \end{bmatrix}, \tag{169}$$

$$\underline{s}_{1+}^{(D)} = \sum_{k=1}^{\infty} \underline{x}_k(\mathbb{1} \otimes \underline{e}_i^T) = \frac{1}{\lambda}\underline{y}_0\boldsymbol{R}^2(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{B}(\mathbb{1} \otimes \underline{e}_i^T), \tag{170}$$

where the subscript $1+$ *refers to the cases when there is at least one customer in the system and* $\underline{s}_{1+}^{(D)}$ *is obtained according to (166).*

Proof. If a departure leaves the MAP/MAP/1 queue busy such that the phase of the service MAP is j then the time to the next departure is phase type distributed time with initial vector $\underline{e}_j$ and generator $\boldsymbol{S_0}$.

If a departure leaves the MAP/MAP/1 queue empty such that the phase of the arrival MAP is i and service MAP is j then the time to the next departure is the sum of two phase type distributed times, the first one with initial vector $\underline{e}_i$ and generator $\boldsymbol{D_0}$ and the second one with initial vector $\underline{e}_j$ and generator $\boldsymbol{S_0}$.

The Laplace transform of the stationary inter departure time, $\mathcal{H}$, is

$$E(e^{-s\mathcal{H}}) = \sum_{j=1}^{m}[\underline{s}_{1+}^{(D)}]_j\,\underline{e}_j(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0 + \sum_{i=1}^{m}\sum_{j=1}^{m}[\boldsymbol{U_0}]_{i,j}\,\underline{e}_i(s\boldsymbol{I}-\boldsymbol{D_0})^{-1}\underline{d}_0\,\underline{e}_j(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0$$

where

$$\underline{e}_i(s\boldsymbol{I}-\boldsymbol{D_0})^{-1}\underline{d}_0 = \underline{d}_0^T((s\boldsymbol{I}-\boldsymbol{D_0})^{-1})^T\underline{e}_i^T = \underline{d}_0^T(s\boldsymbol{I}-\boldsymbol{D_0}^T)^{-1}\underline{e}_i^T\,.$$

Using this we have

$$E(e^{-s\mathcal{H}}) = \underline{s}_{1+}^{(D)}(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0 + \underline{d}_0^T((s\boldsymbol{I}-\boldsymbol{D_0}^T)^{-1}\boldsymbol{U_0}(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0.$$

Partitioning the Laplace transform of the matrix exponential distribution with representation $\underline{u}, \boldsymbol{M}, \underline{w}$ we have

$$\begin{aligned}
&\underline{u}(s\boldsymbol{I}-\boldsymbol{M})^{-1}(-\boldsymbol{M})\underline{w} = \\
&\begin{bmatrix} \underline{d}_0^T & \underline{s}_{1+}^{(D)} \end{bmatrix} \begin{bmatrix} s\boldsymbol{I}-\boldsymbol{D_0}^T & -\boldsymbol{U_0} \\ \boldsymbol{0} & s\boldsymbol{I}-\boldsymbol{S_0} \end{bmatrix}^{-1} \begin{bmatrix} -\boldsymbol{D_0}^T & -\boldsymbol{U_0} \\ \boldsymbol{0} & -\boldsymbol{S_0} \end{bmatrix} \begin{bmatrix} (-\boldsymbol{D_0}^T)^{-1}\boldsymbol{U_0}\mathbb{1} \\ \mathbb{1} \end{bmatrix} = \\
&\begin{bmatrix} \underline{d}_0^T & \underline{s}_{1+}^{(D)} \end{bmatrix} \begin{bmatrix} (s\boldsymbol{I}-\boldsymbol{D_0}^T)^{-1} & (s\boldsymbol{I}-\boldsymbol{D_0}^T)^{-1}\boldsymbol{U_0}(s\boldsymbol{I}-\boldsymbol{S_0})^{-1} \\ \boldsymbol{0} & (s\boldsymbol{I}-\boldsymbol{S_0})^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \underline{s}_0 \end{bmatrix} = \\
&\underline{d}_0^T(s\boldsymbol{I}-\boldsymbol{D_0}^T)^{-1}\boldsymbol{U_0}(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0 + \underline{s}_{1+}^{(D)}(s\boldsymbol{I}-\boldsymbol{S_0})^{-1}\underline{s}_0.
\end{aligned}$$

□

Corollary 2. *When in a MAP/MAP/1 queue the order of the arrival MAP is* N_A *and that of the service MAP is* N_S*, then the order of the phase type distributed inter-departure time distribution is at most* $N_A + N_S$ *and consequently the number of independent inter-departure time moments is at most* $2(N_A + N_S) - 1$.

Proof. The corollary is a straight forward consequence of Theorem 21. □

Theorem 22. *The stationary lag-1 joint moments of two consecutive inter-departure times $\mathcal{H}_0$ and $\mathcal{H}_1$ of a MAP/MAP/1 queue can be computed as*

$$E(\mathcal{H}_0^i \mathcal{H}_1^j) = \underline{z}\, i!(-\boldsymbol{H_0})^{-i-1}\boldsymbol{H_1}\, j!(-\boldsymbol{H_0})^{-j}\mathbb{1}, \tag{171}$$

where

$$\underline{z} = \begin{bmatrix} \underline{x}_0 & \underline{x}_1 & \underline{x}_{2+} \end{bmatrix}, \tag{172}$$

$$\underline{x}_{2+} = \sum_{k=2}^{\infty} \underline{x}_k = \frac{1}{\lambda}\underline{y}_0 \boldsymbol{R}^3(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{B}, \tag{173}$$

$$\boldsymbol{H_0} = \begin{bmatrix} \boldsymbol{L_0} & \boldsymbol{F} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{F} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}+\boldsymbol{F} \end{bmatrix}, \tag{174}$$

$$\boldsymbol{H_1} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}. \tag{175}$$

Proof. Since we focus on the joint moments of two consecutive inter-departure times we have to consider the following three cases:

- a departure leaves the queue empty, with probability $\underline{x}_0$;
- a departure leaves one customer in the queue, with probability $\underline{x}_1$;
- a departure leaves at least two customers in the queue, with probability $\underline{x}_{2+}$.

For all the three cases, the computation of the joint moments of inter-departure times is based on constructing the MAP that generates the departures and then computing the joint moments based on (55).

The process evolution up to the second departure is different in the three cases. Let us first consider the third case which is the simplest. If there are at least two customers in the queue at a departure, then the queue can not become empty before the next two departures. For this reason the joint moments of the next two inter-departure times do not depend on the arrivals. Consequently, in this case, it is enough to consider the state transitions which are assigned to a departure, $\boldsymbol{B}$, and the ones which are not, $\boldsymbol{L_0}+\boldsymbol{F}$. As a result, in this case the lag-1 joint moments can be computed as

$$\begin{aligned} E(\mathcal{H}_0^i \mathcal{H}_1^j I_{\{\mathcal{Y}(0)\geq 2\}}) &= \underline{x}_{2+}\, i!(-\boldsymbol{L}-\boldsymbol{F})^{-i}(-\boldsymbol{L}-\boldsymbol{F})^{-1}\boldsymbol{B}\, j!(-\boldsymbol{L}-\boldsymbol{F})^{-j}\mathbb{1} \\ &= \underline{x}_{2+}\, i!(-\boldsymbol{L}-\boldsymbol{F})^{-i-1}\boldsymbol{B}\, j!(-\boldsymbol{L}-\boldsymbol{F})^{-j}\mathbb{1}, \end{aligned} \tag{176}$$

where $\mathcal{Y}(t)$ denotes the number of customers at time t, we assume that a departure occurred at $t = 0$ and $I_{\{A\}}$ equals one when A is true and zero otherwise. In the second case, i.e., when a departure leaves one customer in the queue, we need to take into consideration one arrival as well in order to compute the joint moments of the next two inter-departure times.

This arrival can happen either before or after the first departure and is taken into account by the block $\boldsymbol{F}$ in position $(2,3)$ of $\boldsymbol{H_0}$ in (174).

Since in the third case the queue is left empty, for the calculation of the joint moments of the next two inter-departure times we have to consider two arrivals. The first happens before the first departure and is taken into account by the block $\boldsymbol{F}$ in position $(1,2)$ of $\boldsymbol{H_0}$ in (174). The second arrival can happen either before the first departure or after the first departure and is considered the same way as the arrival in the second case.

The three cases can be organized in a single compact form as presented in (171-175). □

Note that also the marginal moments of the inter-departure times can be computed based on Theorem 22 by setting j to 0 in (171). Having computed the marginal moments and the lag-1 joint moments of the departure process of a queue, we apply the method described in Section 3.2.4 to construct a MAP with such parameters and use this MAP as an approximation of the output process. If this method does not return a valid Markovian representation, the solutions recommended in Section 3.3.3 are the remedy.

It is important to note that

- the MAP defined by $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$ in (174) and (175) is not a good output process model of the MAP/MAP/1 queue,
- the embedded stationary distribution of the MAP defined by $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$ is different from $\underline{z}$,
- the finite dimensional matrix expression in (171) is exact, because vector $\underline{z}$ represents the effect of the infinite queue.

dc_1412_17

6

ANALYSIS OF THE MMAP[K]/PH[K]/1-FCFS QUEUE

The MMAP[K]/PH[K]/1-FCFS queue is the multi-class variant of the MAP/PH/1-FCFS queue in which the arrival process is a MMAP, the service times are phase-type distributed, and different classes of customers can have different service time distributions.

Let us denote the number of customer types by K. The matrices characterizing the MMAP of the arrivals are denoted by $\boldsymbol{D_k}, k = 0, \ldots, K$, and the arrival rate of type k customers is λ_k (see Section 3.1.2).

The initial vector and transient generator of the PH distribution representing the type k service times $\mathcal{S}_k$ are denoted by $\underline{\sigma}_k, \boldsymbol{S_k}, k = 1, \ldots, K$, respectively. Vector $\underline{\beta}_k$ is the stationary phase distribution of the service process, that is, the unique solution of $\underline{\beta}_k(\boldsymbol{S_k} - \boldsymbol{S_k}\mathbb{1}\underline{\sigma}_k) = \underline{0}, \underline{\beta}_k\mathbb{1} = 1$. The service rate of type k customers is then $\mu_k = \underline{\beta}_k(-\boldsymbol{S_k})\mathbb{1}$.

With these notations the load of the queue ρ is given by $\rho = \sum_{k=1}^{K} \lambda_k/\mu_k$, representing the fraction of time when the server is busy (provided that $\rho < 1$).

In case of the MAP/MAP/1 queue the distribution of the number of customers in the system was derived using the direct analysis of the queue length process. In the multi-class case, however, the corresponding Markov chain has a structure for which no explicit solutions are available in the literature. Therefore, in case of the MMAP[K]/PH[K]/1-FCFS queue all performance measures are derived by the analysis of the age process.

6.1 THE DISTRIBUTION OF THE AGE PROCESS

The class of MMAP[K]/PH[K]/1 queues forms a subclass of the semi-Markovian SM[K]/PH[K]/1 queues, the age process of which was considered in [41]. The matrix-exponential stationary solution of the distribution of the (skip-free to the right) age process is also derived in [41].

Nevertheless, we describe an alternative solution method here, which is based on the transformation of the age process to a fluid model (following [83], as we did in Section 5.2.1, too). This approach has several advantages over the direct solution [41], including that

- it is much easier to work with processes that are skip-free both to the left and to the right technically;
- it allows to make use of the mature, well proven numerical procedures for the stationary solution of MFMs in the age process analysis. In particular, to obtain the parameters of the matrix-exponentially distributed age distribution [41] describes only a linearly convergent functional iteration, while the MFMs based solution allows to compute these parameters by quadratically convergent iterative algorithms.

The rest of this section focuses on the stationary solution of the multi-dimensional process $\{\mathcal{A}(t), \mathcal{J}_A(t), \mathcal{J}_S(t), \mathcal{C}(t)\}$, that keeps track of 1) the age process $\mathcal{A}(t)$, 2) the phase of the

arrival process $\mathcal{J}_A(t)$, 3) the phase of the service process $\mathcal{J}_S(t)$, and the type (class) of the customer in the server $\mathcal{C}(t)$. Performance measures related to both the number of customers in the system and the sojourn time of customers can be derived from this multi-dimensional process.

In order to simplify the analysis, the two dimensional process $\{\mathcal{C}(t), \mathcal{J}_S(t), t \geq 0\}$ describing the type (class) of the customer currently in service and the current service phase will be represented by a (generalized) PH distribution of size $N_S = \sum_{k=1}^{K} N_k$ with generator

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S_1} & & & \\ & \boldsymbol{S_2} & & \\ & & \ddots & \\ & & & \boldsymbol{S_K} \end{bmatrix},$$

and the initial vector given that a type k customer is going to be served next is

$$\underline{\sigma}^{(k)} = [\underbrace{0, \ldots, 0}_{\sum_{j=1}^{k-1} N_j}, \underline{\sigma}_k, \underbrace{0, \ldots, 0}_{\sum_{j=k+1}^{K} N_j}], \quad k = 1, \ldots, K,$$

where N_k is the size of the PH representation of $\mathcal{S}_k$, the service time of type k customers.

Thus, the "large" generator $\boldsymbol{S}$ is composed by all the generators associated with the various customer classes, and the "large" PH distribution is initialized in a class dependent way. For later use, let $N = N_A \cdot N_S$, with N_A being the number of phases of the MMAP generating the arrivals.

The proposed representation of the service times allows to adapt the method described in Section 5.2.1 to the analysis of the above introduced multi-dimensional process. Hence, $\{\mathcal{A}(t), \mathcal{J}_A, \mathcal{J}_S(t)\}$ is transformed to a skip-free canonical fluid model with generator

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_{++} & \boldsymbol{Q}_{+-} \\ \boldsymbol{Q}_{-+} & \boldsymbol{Q}_{--} \end{bmatrix}. \tag{177}$$

Matrix $\boldsymbol{Q}_{++}$ describes the evolution of the system between arrivals. We have

$$\boldsymbol{Q}_{++} = \boldsymbol{S} \otimes \boldsymbol{I}, \tag{178}$$

thus the type of the current customer, its service phase and the phase of the MMAP at the last arrival instant are all encoded in the state space. When the service ends (with rates $(-\boldsymbol{S})\mathbb{1}$), the arrival process is resumed and the transformed process moves to the negative states responsible for generating the downward jump of the age process, thus

$$\boldsymbol{Q}_{+-} = (-\boldsymbol{S})\mathbb{1} \otimes \boldsymbol{I}. \tag{179}$$

In the negative states only the arrival process is active, leading to

$$\boldsymbol{Q}_{--} = \boldsymbol{D_0}. \tag{180}$$

Finally, when a customer arrives, a transition occurs to the positive states, and the PH distribution associated with the type of the new customer is initiated. Thus we have

$$\boldsymbol{Q}_{-+} = \sum_{k=1}^{K} \underline{\sigma}^{(k)} \otimes \boldsymbol{D_k}. \tag{181}$$

Like in Section 5.2.1, the stationary distribution of the age process is obtained from the solution of the fluid model by censoring to the positive states. If the solution of the canonical fluid model is given in form $\underline{\pi}(x) = \underline{p}_- \boldsymbol{Q}_{-+} e^{\boldsymbol{K}x} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi} \end{bmatrix}$, the pdf of the age process, $\underline{a}(x)$ can be expressed by

$$\underline{a}(x) = \frac{\underline{\pi}_+(x)}{\int_{y=0}^{\infty} \underline{\pi}_+(y)\mathbb{1}\, dy} = \frac{\underline{p}_- \boldsymbol{Q}_{-+}}{\underline{p}_- \boldsymbol{Q}_{-+}(-\boldsymbol{K})^{-1}\mathbb{1}} e^{\boldsymbol{K}x} = \underline{a}(0)e^{\boldsymbol{T}x}, \quad x \geq 0, \tag{182}$$

where in the last step we again switched to the traditional representation of the age process (used in the literature since [75]).

According to (182) the pdf at $x = 0$ is $\underline{a}(0) = \frac{\underline{p}_- \boldsymbol{Q}_{-+}}{\underline{p}_- \boldsymbol{Q}_{-+}(-\boldsymbol{K})^{-1}\mathbb{1}}$. In [41] the same quantity is expressed in the following closed form as well:

$$\underline{a}(0) = \frac{1}{\rho} \sum_{k=1}^{K} \frac{\lambda_k}{\mu_k} \left([0, \ldots, 0, \underline{\beta}_k, 0, \ldots, 0] \otimes \frac{\underline{\theta} \boldsymbol{D}_k}{\lambda_k} \right) (-\boldsymbol{T}), \tag{183}$$

In this formula the right-hand term of the Kronecker product is the phase distribution of the MMAP at the type k arrival epochs, and vector $\underline{\beta}_k$ denotes the steady state phase distribution of the type k service process.

6.2 DERIVING THE SOJOURN TIME FROM THE AGE PROCESS

The sojourn time of a customer is equal to its age when it leaves the system. Hence, the density function of the sojourn time of class k customers is

$$F_{\mathcal{T}_k}(t) = P(\mathcal{T}_k < t) = \frac{\int_0^t \underline{a}(x)(\underline{s}^{(k)} \otimes \mathbb{1})\, dx}{\int_0^{\infty} \underline{a}(x)(\underline{s}^{(k)} \otimes \mathbb{1})\, dx} = 1 - \frac{\underline{a}(0)e^{\boldsymbol{T}t}(-\boldsymbol{T})^{-1}(\underline{s}^{(k)} \otimes \mathbb{1})}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\underline{s}^{(k)} \otimes \mathbb{1})},$$

where column vector $\underline{s}^{(k)}$ consists of the class k service completion rates in various states of background process, thus

$$\underline{s}^{(k)} = \begin{bmatrix} 0 & \ldots & 0 & (-\boldsymbol{S}_k\mathbb{1})^T & 0 & \ldots & 0 \end{bmatrix}^T. \tag{184}$$

As we did in Section 5.2.1 in the single-class case, we provide the LST of the type k sojourn time $f^*_{\mathcal{T}_k}(s)$ and the nth moment $E(\mathcal{T}_k^n)$ for completeness:

$$f^*_{\mathcal{T}_k}(s) = \frac{\underline{a}(0)(s\boldsymbol{I} - \boldsymbol{T})^{-1}(\underline{s}^{(k)} \otimes \mathbb{1})}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\underline{s}^{(k)} \otimes \mathbb{1})}, \tag{185}$$

$$E(\mathcal{T}_k^n) = \frac{n!\, \underline{a}(0)(-\boldsymbol{T})^{-n-1}(\underline{s}^{(k)} \otimes \mathbb{1})}{\underline{a}(0)(-\boldsymbol{T})^{-1}(\underline{s}^{(k)} \otimes \mathbb{1})}. \tag{186}$$

6.3 ANALYSIS OF THE NUMBER OF CUSTOMERS

To express the distribution of the *total* number of customers in the system we first need to express the distribution of the number of customers *waiting* in the queue. At time t the age of the customer in the server is $\mathcal{A}(t)$, thus at time t those customers are waiting in the queue that arrived in $(0, t)$. The probability that $\underline{n} = \{n_k, k = 1, \ldots, K\}$ arrivals are generated by a MMAP in $(0, t)$, denoted by $\boldsymbol{P}(\underline{n}, t)$, is given by the differential equation (see [39])

$$\frac{d}{dt}\boldsymbol{P}(\underline{n}, t) = \boldsymbol{P}(\underline{n}, t)\boldsymbol{D}_0 + \sum_{k=1}^{K} \boldsymbol{P}(\underline{n} - \underline{e}_k, t)\boldsymbol{D}_k, \tag{187}$$

where $\underline{e}_k$ is a column vector containing zeros except the kth index, where it is 1.

Let us now introduce matrix $\boldsymbol{L}(\underline{n})$, which is related to the probability that $\underline{n}$ customers arrive over the (stationary) age of the current customer in the service, as follows:

$$\boldsymbol{L}(\underline{n}) = \int_0^\infty e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I})\,dx. \tag{188}$$

Here, $e^{\boldsymbol{T}x}$ is the density that the age of the current customer is x, and $\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I}$ is the probability that $\underline{n}$ customers arrived (including various classes) over time x.

Inserting the differential equation (187) into (188) and integrating by parts gives ([41], example 5.2)

$$\begin{aligned}
\boldsymbol{L}(\underline{n}) &= \int_0^\infty e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I}) \\
&= \boldsymbol{T}^{-1}\left(\left[e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I})\right]_0^\infty - \int_0^\infty e^{\boldsymbol{T}x}(\boldsymbol{P}'(\underline{n},x)\otimes\boldsymbol{I})\,dx\right) \\
&= \boldsymbol{T}^{-1}\left(\left[e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I})\right]_0^\infty - \underbrace{\int_0^\infty e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I})\,dx}_{\boldsymbol{L}(\underline{n})}(\boldsymbol{D_0}\otimes\boldsymbol{I})\right. \\
&\qquad \left. - \sum_{k=1,n_k>0}^{K}\underbrace{\int_0^\infty e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n}-\underline{e}_k,x)\otimes\boldsymbol{I})\,dx}_{\boldsymbol{L}(\underline{n}-\underline{e}_k)}(\boldsymbol{D_k}\otimes\boldsymbol{I})\right),
\end{aligned} \tag{189}$$

which, after pre-multiplying by $\boldsymbol{T}$, leads to a matrix recursion composed by Sylvester equations. Since $\left[e^{\boldsymbol{T}x}(\boldsymbol{P}(\underline{n},x)\otimes\boldsymbol{I})\right]_0^\infty$ is $-\boldsymbol{I}$ when $\underline{n}=\underline{0}$ and $\boldsymbol{0}$ otherwise, we get

$$\boldsymbol{T}\boldsymbol{L}(\underline{0}) + \boldsymbol{L}(\underline{0})(\boldsymbol{D_0}\otimes\boldsymbol{I}) = -\boldsymbol{I}, \tag{190}$$

$$\boldsymbol{T}\boldsymbol{L}(\underline{n}) + \boldsymbol{L}(\underline{n})(\boldsymbol{D_0}\otimes\boldsymbol{I}) = -\sum_{k=1,n_k>0}^{K}\boldsymbol{L}(\underline{n}-\underline{e}_k)(\boldsymbol{D_k}\otimes\boldsymbol{I}), \quad \underline{n}\neq\underline{0}. \tag{191}$$

Thus, the distribution of the number of *waiting* customers in the system (belonging to different types) is

$$w(\underline{n}) = \begin{cases} 1-\rho+\underline{a}(0)\boldsymbol{L}(\underline{0})\mathbb{1}, & \underline{n}=\underline{0}, \\ \underline{a}(0)\boldsymbol{L}(\underline{n})\mathbb{1}, & \underline{n}\neq\underline{0}, \end{cases} \tag{192}$$

where, according to the first case ($\underline{n}=\underline{0}$), there are two situations leading to zero waiting customers: with probability $1-\rho$ the entire system is empty, and with probability $\underline{a}(0)\boldsymbol{L}(\underline{0})\mathbb{1}$ there is a customer in the server (the age process is positive), but no further customers are waiting behind it.

To obtain the *total* number of customers, the customer in the server has to be taken into consideration as well. Let us introduce column vector $\underline{h}_k$ that sums over the states where a type k customer is in the server, thus

$$\underline{h}_k = \mathbb{1}_{N_A}\otimes\left[\underline{0}_{N_1}\ \cdots\ \underline{0}_{N_{k-1}}\ \ \mathbb{1}_{N_k}^T\ \ \underline{0}_{N_{k+1}}\ \cdots\ \underline{0}_{N_K}\right]^T. \tag{193}$$

Finally, the distribution of the number of customers in the system, denoted by $y(\underline{n}) = P(\mathcal{Y}_1 = n_1,\ldots,\mathcal{Y}_K = n_k)$, is given by

$$y(\underline{n}) = \begin{cases} 1-\rho, & \underline{n}=\underline{0}, \\ \underline{a}(0)\sum_{k=1,n_k>0}^{K}\boldsymbol{L}(\underline{n}-\underline{e}_k)h_k, & \underline{n}\neq\underline{0}, \end{cases} \tag{194}$$

where $\underline{a}(0)\boldsymbol{L}(\underline{n}-\underline{e}_k)h_k$ is the probability that there is a type k customer in the server and $\underline{n}-\underline{e}_k$ customers are waiting in the queue, hence there are $\underline{n}$ customers in the system in total.

6.4 ANALYSIS OF THE DEPARTURE PROCESS

6.4.1 *Distribution of the age process at departure instants*

We are focusing on the departure process in this section, hence we will be interested in the age process embedded at service completion instants. Since the state-dependent service completion rate is $-\boldsymbol{S}\mathbb{1}$, the density of the age process just before service completion instants $\underline{a}_D(x)=\{(\underline{a}_D(x))_i,\, i=1,\ldots,N_A\}$ can be expressed as

$$\underline{a}_D(x)=\frac{1}{\lambda}(\rho\underline{a}(0)e^{\boldsymbol{T}x})(-\boldsymbol{S}\mathbb{1}\otimes\boldsymbol{I}),\quad x>0, \tag{195}$$

where λ is the normalization constant. Integrating $\underline{a}_D(x)$ over x and making use of (183) for $\underline{a}(0)$ gives

$$\int_{x=0}^{\infty}\underline{a}_D(x)\,dx=\sum_{k=1}^{K}\frac{1}{\mu_k}\left([0,\ldots,0,\underline{\beta}_k,0,\ldots,0](-\boldsymbol{S}\mathbb{1})\otimes\frac{\underline{\theta}\boldsymbol{D}_k}{\lambda}\right)=\underline{\alpha}, \tag{196}$$

which is the stationary phase distribution of the MMAP at arrival instants. In (196) we utilized that $\mu_k=\underline{\beta}_k(-\boldsymbol{S}_k)\mathbb{1}$ and that $\underline{\alpha}=\sum_{k=1}^{K}\underline{\theta}\boldsymbol{D}_k/\lambda$.

6.4.2 *Phase transitions over the busy period of the age process*

In [75] it is proven that matrix $\boldsymbol{T}$ is the minimal solution of the matrix equation

$$\boldsymbol{T}=\boldsymbol{S}\otimes\boldsymbol{I}+\underbrace{\int_{x=0}^{\infty}e^{\boldsymbol{T}x}\,(-\boldsymbol{S}\mathbb{1}\otimes\boldsymbol{I})\,e^{\boldsymbol{D}_0x}dx}_{\boldsymbol{Y}_0}\sum_{k=1}^{K}\left(\underline{\sigma}^{(k)}\otimes\boldsymbol{D}_k\right). \tag{197}$$

Theorem 4.4 in [41] also indicates that all the eigenvalues of $\boldsymbol{T}$ lie in the open left half plane.

The integral term of the matrix equation (197) involving two matrix exponentials has a closed form solution. Theorem 36 (in Appendix A.2) implies that $\boldsymbol{Y}_0$ is the unique solution of the following Sylvester matrix equation:

$$\boldsymbol{T}\boldsymbol{Y}_0+\boldsymbol{Y}_0\boldsymbol{D}_0=(\boldsymbol{S}\mathbb{1})\otimes\boldsymbol{I}. \tag{198}$$

Matrix $\boldsymbol{Y}_0$ has an important role in the departure process analysis, that has a stochastic interpretation as well. Entry $(\boldsymbol{Y}_0)_{i,j}$ of this matrix is the probability that the age process returns to level 0 in phase j for the first time, given that it left level 0 in phase i.

One way to compute $\boldsymbol{Y}_0$ from matrix $\boldsymbol{T}$ is the solution of (198). However, matrix $\boldsymbol{\Psi}$ of the MFM corresponding to the transformed age process (Section 6.1) gives exactly this quantity, hence $\boldsymbol{Y}_0=\boldsymbol{\Psi}$.

6.4.3 *The lag-n joint transform of the departure process*

In this section we derive an expression for the joint LST of the 1st and $(n+1)$th inter-departure time. Let T_n denote the nth departure time with $T_0=0$ and let $\mathcal{H}_n=T_n-T_{n-1}$

denote the nth inter-departure time for $n \geq 1$. Furthermore, let $\mathcal{C}_n$ denote the type of the nth departing customer. Then $f^{(k,p)*}_{\mathcal{H}(n)}(s_1, s_2)$, the LST of the joint distribution of two inter-departure times and the corresponding customer types can be defined as

$$f^{(k,p)*}_{\mathcal{H}(n)}(s_1, s_2) = \int_{t_1=0}^{\infty} \int_{t_2=0}^{\infty} e^{-s_1 t_1 - s_2 t_2} dP(\mathcal{H}_1 < t_1, \mathcal{H}_{n+1} < t_2, \mathcal{C}_1 = k, \mathcal{C}_{n+1} = p), \tag{199}$$

for $p, k \in \{1, \ldots, K\}$ and $n \geq 1$.

Observe that the inter-departure times are

- either equal to service times (during the busy periods of the queue),
- or, if a customer arrives to an idle queue, they are equal to the service time of the customer plus the preceding idle time.

Due to this kind of relation between the busy periods and the inter-departure times we introduce several busy period related quantities before providing the solution for (199).

Denote $\boldsymbol{I_i} = [\boldsymbol{I}\ \boldsymbol{0}\ \ldots\ \boldsymbol{0}]$ and $\boldsymbol{J_i} = [\boldsymbol{0}\ \ldots\ \boldsymbol{0}\ \boldsymbol{I}]^T$ such that they have size $N_A \times i \cdot N_A$ and $i \cdot N_A \times N_A$, respectively. Further, let $\boldsymbol{J_{i,j}} = [\boldsymbol{J_j}^T\ \boldsymbol{0}\ \ldots\ \boldsymbol{0}]^T$ such that it is a size $i \cdot N_A \times N_A$ matrix.

We start by defining $(\boldsymbol{M_{k,i}}(t))_{j,j'}$, for $i \geq 1$, $k \in \{1, \ldots, K\}$ and $j, j' \in \{1, \ldots, N_A\}$, as the probability that i customers get served during a busy period that was initiated by a type k arrival, while the service time of the initial type k customer is at most t and the MMAP phase equals j at the start and j' at the end of the busy period. Let $\boldsymbol{M_{k,i}}(t)$ be the matrix with entry (j, j') equal to $(\boldsymbol{M_{k,i}}(t))_{j,j'}$. Let $\boldsymbol{M^*_{k,i}}(s)$ be the LST of $\boldsymbol{M_{k,i}}(t)$ and denote $\boldsymbol{M^*_{k,i}}(0) = \boldsymbol{M_{k,i}}(\infty)$ as $\boldsymbol{M_{k,i}}$.

The following lemma gives an expression for the matrices $\boldsymbol{M^*_{k,i}}(s)$. We note that the final results do note require the computation (nor the inversion) of the size $i \cdot N_A \times i \cdot N_A$ matrices $\boldsymbol{Q_i}$ defined in this lemma.

Lemma 2. *The matrices $\boldsymbol{M^*_{k,i}}(s)$ can be expressed recursively as*

$$\boldsymbol{M^*_{k,i}}(s) = (\underline{\sigma}_k \otimes \boldsymbol{I_i})((s\boldsymbol{I} - \boldsymbol{S_k}) \oplus \boldsymbol{Q_i})^{-1}(-\boldsymbol{S_k}\mathbb{1} \otimes \boldsymbol{J_i}). \tag{200}$$

where $\boldsymbol{Q_1} = \boldsymbol{D_0}$ and $\boldsymbol{Q_i}$ is the size $i \cdot N_A \times i \cdot N_A$ block Toeplitz matrix given by

$$\boldsymbol{Q_i} = \begin{bmatrix} \boldsymbol{D_0} & \sum_{q=1}^{K} \boldsymbol{D_q}\boldsymbol{M_{q,1}} & \cdots & \sum_{q=1}^{K} \boldsymbol{D_q}\boldsymbol{M_{q,i-1}} \\ & \ddots & \ddots & \vdots \\ & & \boldsymbol{D_0} & \sum_{q=1}^{K} \boldsymbol{D_q}\boldsymbol{M_{q,1}} \\ & & & \boldsymbol{D_0} \end{bmatrix}. \tag{201}$$

Proof. As $e^{\boldsymbol{A}\otimes\boldsymbol{I}} = e^{\boldsymbol{A}} \otimes \boldsymbol{I}$, $e^{\boldsymbol{I}\otimes\boldsymbol{B}} = \boldsymbol{I} \otimes e^{\boldsymbol{B}}$ and $e^{\boldsymbol{A}+\boldsymbol{B}} = e^{\boldsymbol{A}}e^{\boldsymbol{B}}$ if $\boldsymbol{A}$ and $\boldsymbol{B}$ commute, (200) can be rewritten as

$$\begin{aligned} \boldsymbol{M^*_{k,i}}(s) &= (\underline{\sigma}_k \otimes \boldsymbol{I_i}) \int_{y=0}^{\infty} \left(e^{(\boldsymbol{S_k} - s\boldsymbol{I})y} \otimes \boldsymbol{I}\right) \left(\boldsymbol{I} \otimes e^{\boldsymbol{Q_i}y}\right) dy (-\boldsymbol{S_k}\mathbb{1} \otimes \boldsymbol{J_i}) \\ &= \int_{y=0}^{\infty} \left(\underline{\sigma}_k e^{\boldsymbol{S_k}y}(-\boldsymbol{S_k})\mathbb{1}e^{-sy}\right) \otimes \left(\boldsymbol{I_i} e^{\boldsymbol{Q_i}y} \boldsymbol{J_i}\right) dy. \end{aligned}$$

(For the identities of the Kronecker operations see Appendix A.1).

As such it suffices to prove that

$$\boldsymbol{M}^*_{k,i}(s) = \int_{y=0}^{\infty} f_{S_k}(y) e^{-sy} \boldsymbol{I}_i e^{\boldsymbol{Q}_i y} \boldsymbol{J}_i \, dy, \tag{202}$$

with $\boldsymbol{Q}_i$ given by (201). The result for $i = 1$ is immediate as $\boldsymbol{I}_1 = \boldsymbol{I} = \boldsymbol{J}_1$, $\boldsymbol{Q}_1 = \boldsymbol{D}_0$ and

$$\boldsymbol{M}_{k,1}(t) = \int_{y=0}^{t} f_{S_k}(y) e^{\boldsymbol{D}_0 y} \, dy,$$

as there should be no arrivals during the service of the type k customer that initiated the busy period.

To establish the general case, we assume that the order of service is preemptive (resume) last-come-first-served instead of FCFS. Notice, the probabilities $(\boldsymbol{M}_{k,i}(t))_{j,j'}$ are not affected by the order of service and therefore the expressions are also valid for the FCFS order considered in this chapter.

Assume that the type k customer is in service and that the first arrival that occurs during the service is of type q. Then with probability $(\boldsymbol{M}_{q,r_1})_{j,j'}$ this arrival induces its own sub-busy period during which r_1 customers are served, while the MMAP phase changes from j to j'. Hence, when the type k customer resumes service the MMAP phase equals j'. If another customer arrives while the type k customer is served, this customer will induce another sub-busy period during which r_2 customers are served, etc. Hence, when the initial type k customer gets interrupted for the n-th time, the MMAP phase changes according to the matrix $\sum_{q=1}^{K} \boldsymbol{D}_q \boldsymbol{M}_{q,r_n}$. In order to have exactly i customers served, the sum of all the r_n values should equal $i-1$. Hence, if the service time of the initial type k customer equals y, we therefore find that $(\boldsymbol{I}_i e^{\boldsymbol{Q}_i y} \boldsymbol{J}_i)_{j,j'}$ represents the probability that i customers are served during the busy period initiated by the type k customer, while the MMAP phase equals j at the start and j' at the end of the busy period. This suffices to establish (202). □

Define the $(\boldsymbol{Z}_{k,i}(t))_{j,j'}$, for $i \geq 1, k \in \{1, \ldots, K\}$ and $j, j' \in \{1, \ldots, N_A\}$, as the probability that the ith customers leaves the server idle at departure time, given that a type k arrival (called the 1st customer) initiated a busy period, the service time of customer 1 is at most t and the MMAP phase equals j at the start of the busy period and j' when the ith customer leaves. Note, the ith customer marks the end of a busy period, but not necessarily the one initiated by customer 1 (unless $i = 1$). Let $\boldsymbol{Z}_{k,i}(t)$ be the matrix with entry (j, j') equal to $(\boldsymbol{Z}_{k,i}(t))_{j,j'}$. Let $\boldsymbol{Z}^*_{k,i}(s)$ be the LST of $\boldsymbol{Z}_{k,i}(t)$ and denote $\boldsymbol{Z}^*_{k,i}(0) = \boldsymbol{Z}_{k,i}(\infty)$ as $\boldsymbol{Z}_{k,i}$.

Lemma 3. *The $N_A \times N_A$ matrices $\boldsymbol{Z}^*_{k,i}(s)$ can be expressed recursively as*

$$\begin{aligned} \boldsymbol{Z}^*_{k,1}(s) &= \boldsymbol{M}^*_{k,1}(s), \\ \boldsymbol{Z}^*_{k,i}(s) &= \boldsymbol{M}^*_{k,i}(s) + \sum_{j=1}^{i-1} \boldsymbol{M}^*_{k,j}(s) \left(\sum_{q=1}^{K} \boldsymbol{P}_q \boldsymbol{Z}_{q,i-j} \right), \end{aligned} \tag{203}$$

for $i \geq 2$.

Proof. The equality $\boldsymbol{Z}^*_{k,1}(s) = \boldsymbol{M}^*_{k,1}(s)$ is immediate as $(\boldsymbol{Z}_{k,1}(t))_{j,j'}$ and $(\boldsymbol{M}_{k,1}(t))_{j,j'}$ represent the same probability. For $i \geq 2$, there are two options: either the busy period initiated by customer 1 ends when the i-th customer leaves (which corresponds to $\boldsymbol{M}^*_{k,i}(s)$) or it ends when the jth customer leaves with $j < i$. In the latter case assume the $j+1$th customer is of

type q, then this customer initiates another busy period and we still demand that customer $i > j$ leaves the server idle. Hence, in the latter case we find

$$\boldsymbol{Z}_{k,i}(t) = \sum_{j=1}^{i-1} \boldsymbol{M}_{k,j}(t) \left(\sum_{q=1}^{K} (\boldsymbol{D}_0)^{-1} \boldsymbol{D}_q \boldsymbol{Z}_{q,i-j} \right),$$

which implies (203). □

Define the $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$, for $n \geq 1$, $k \in \{1,\ldots,K\}$ and $j,j' \in \{1,\ldots,N_A\}$, as the following conditional probability: given that an age x customer (labeled customer 0) departs and the MMAP phase at its arrival time was j, $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$ holds the probability that (a) the next customer (labeled customer 1) is of type k, (b) the inter-departure time between customers 0 and 1 is at most t, (c) customer n leaves the server idle and (d) the MMAP phase equals j' when customer n departs. Let $\boldsymbol{H}_{k,n}(t,x)$ be the matrix with entry (j,j') equal to $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$. Let $\boldsymbol{H}^*_{k,n}(s,x)$ be the LST of $\boldsymbol{H}_{k,n}(t,x)$ and denote $\boldsymbol{H}^*_{k,n}(0,x) = \boldsymbol{H}_{k,n}(\infty,x)$ as $\boldsymbol{H}_{k,n}(x)$.

Lemma 4. *The $N_A \times N_A$ matrices $\boldsymbol{H}^*_{k,n}(s,x)$ can be computed as*

$$\begin{aligned} \boldsymbol{H}^*_{k,n}(s,x) = {} & e^{\boldsymbol{D}_0 x}(s\boldsymbol{I} - \boldsymbol{D}_0)^{-1}\boldsymbol{D}_k \boldsymbol{Z}^*_{k,n}(s) + \\ & \boldsymbol{I}_{n+1} e^{\boldsymbol{Q}^*_{k,n+1}(s)x} \left[\boldsymbol{J}_{n+1} + \sum_{i=1}^{n-1} \boldsymbol{J}_{n+1,i+1} \left(\sum_{q=1}^{K} \boldsymbol{P}_q \boldsymbol{Z}_{q,n-i} \right) \right], \end{aligned} \tag{204}$$

with

$$\boldsymbol{Q}^*_{k,n+1}(s) = \left[\begin{array}{c|ccc} \boldsymbol{D}_0 & \boldsymbol{D}_k \boldsymbol{M}^*_{k,1}(s) & \ldots & \boldsymbol{D}_k \boldsymbol{M}^*_{k,n}(s) \\ \hline & & & \\ & & \boldsymbol{Q}_n & \\ & & & \end{array}\right].$$

Proof. The probabilities $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$ are not affected by the amount of time that customer 0 had to wait, we may therefore assume that customer 0 initiated a busy period and his service time equals x.

We consider two cases. First, with probability $e^{\boldsymbol{D}_0 x}$, there are no arrivals while customer 0 is in the system. In this case the inter-departure time between customer 0 and 1 consists of an idle period plus the service time of customer 1. Hence, by the probabilistic interpretation of $\boldsymbol{Z}_{k,n}(t)$, the first case results in

$$\boldsymbol{H}^*_{k,n}(t,x, \text{cust. 0 leaves the queue empty}) = e^{\boldsymbol{D}_0 x} \int_{a=0}^{t} e^{\boldsymbol{D}_0 a} \boldsymbol{D}_k \boldsymbol{Z}_{k,n}(t-a)da,$$

which yields the first term appearing in (204).

Second, if there is at least one arrival while customer 0 is in the system, then the inter-departure time between customer 0 and 1 equals the service time of customer 1. Hence, in this case $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$ is also equal to the following conditional probability: given that a customer (labeled customer 0) initiates a busy period, requires service time x and the MMAP phase at its arrival time was j, $(\boldsymbol{H}_{k,n}(t,x))_{j,j'}$ holds the probability that (a) at least one arrival occurs during the service of customer 0 and the first arrival is of type k (labeled customer 1), (b) the service time of customer 1 is at most t, (c) customer n leaves the server idle and (d) the MMAP phase equals j' when customer n departs. Note, the above probability is not

affected by the order of service either. In this case we may therefore think in terms of a preemptive (resume) last-come-first-served system in which customer 0 has service time x. The first arrival during the service of customer 0 must be of type k, its service time should be at most t and the MMAP phase when customer 0 resumes service is determined by $\boldsymbol{D}_k\boldsymbol{M}_{k,r_1}(t)$, provided that customer 1 induces a sub-busy period during which r_1 customers are served. A possible second arrival of some type q will cause the MMAP phase to change according to $\boldsymbol{D}_q\boldsymbol{M}_{q,r_2}$, for some r_2, etc. Notice, in this case there is no restriction on the service time of the type q customer. Hence, for $i = 1, \ldots, n$

$$\boldsymbol{I}_{n+1}e^{\left[\begin{array}{c|ccc} \boldsymbol{D}_0 & \boldsymbol{D}_k\boldsymbol{M}_{k,1}(t) & \ldots & \boldsymbol{D}_k\boldsymbol{M}_{k,n}(t) \\ \hline & & \boldsymbol{Q}_n & \end{array}\right]x}\boldsymbol{J}_{n+1,i+1}.$$

is an $N_A \times N_A$ matrix with entry (j, j') equal to the following conditional probability: given that customer 0 initiates a busy period, has a service time of x and the MMAP phase at its arrival time was j, entry (j, j') holds the probability that (a) customer 1 is of type k and arrives while customer 0 is in the system, (b) the service time of customer 1 is at most t, (c) customer i leaves the server idle and (d) the MMAP phase equals j' when customer i departs. If $i = n$ this results in the term containing $\boldsymbol{J}_{n+1}$ in (204). Otherwise, we need another arrival of some type q (labeled customer $i + 1$) that initiates a busy period such that customer n leaves the server idle. This explains the terms containing $\sum_{q=1}^{K} \boldsymbol{P}_q\boldsymbol{Z}_{q,n-i}$ in (204), for $i = 1, \ldots, n-1$. □

Define $(\underline{v}_{k,n}(t))_j$, for $n \geq 1$, $k \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, N_A\}$, as the probability of the following event: assume we observe the system at an arbitrary departure instant, then the next inter-departure time is at most t and involves a type k customer (labeled customer 1), while customer n leaves the server idle and the phase of the MMAP is j when customer n departs. Let $\underline{v}_{k,n}(t)$ be the vector with entry j equal to $(\underline{v}_{k,n}(t))_j$. Let $\underline{v}^*_{k,n}(s)$ be the LST of $\underline{v}_{k,n}(t)$.

Finally, let $(\underline{v}_0)_j$, for $j \in \{1, \ldots, N_A\}$, be the probability that the server becomes idle at an arbitrary departure instant while the MMAP phase equals j. Denote $\underline{v}_0$ as the vector with entry j equal to $(\underline{v}_0)_j$.

Lemma 5. *The $1 \times N_A$ vectors $\underline{v}^*_{k,n}(s)$ can be expressed as*

$$\underline{v}^*_{k,n}(s) = \frac{\rho\underline{a}(0)}{\lambda}\int_{x=0}^{\infty} e^{\boldsymbol{T}x}(-\boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I})\boldsymbol{H}^*_{k,n}(s, x)dx, \tag{205}$$

while $\underline{v}_0 = \rho\underline{a}(0)\boldsymbol{Y}_0/\lambda$, where $\underline{a}(0)$ and $\boldsymbol{Y}_0$ are defined by (183) and (197).

Proof. From the probabilistic interpretation of $\boldsymbol{H}_{k,n}(t, x)$ it is clear that

$$\underline{v}_{k,n}(t) = \int_{x=0}^{\infty} \underline{a}_D(x)\boldsymbol{H}_{k,n}(t, x)dx,$$

where $\underline{a}_D(x)$ is the density of the age process at departure times. The expression in (205) therefore follows from (195). The expression for $\underline{v}_0$ is immediate from

$$\underline{v}_0 = \int_{x=0}^{\infty} \underline{a}_D(x)e^{\boldsymbol{D}_0x}dx,$$

and the definition of $\underline{a}(0)$ and $\boldsymbol{Y}_0$. □

Theorem 23. *The LST of the joint distribution of the 1st and $(n+1)$th inter-departure time with the first one being of type k and the $n+1$th of type p is given by*

$$f^{(k,p)*}_{\mathcal{H}(n)}(s_1,s_2) = \left[(\underline{\alpha}-\underline{v}_0)(-\boldsymbol{D_0})^{-1}+\underline{v}_0(s_1\boldsymbol{I}-\boldsymbol{D_0})^{-1}\right]\boldsymbol{D_k}\boldsymbol{P}^{n-1}\boldsymbol{P_p}\mathbb{1}f^*_{\mathcal{S}_k}(s_1)f^*_{\mathcal{S}_p}(s_2) + \underline{v}^*_{k,n}(s_1)\left[(s_2\boldsymbol{I}-\boldsymbol{D_0})^{-1}-(-\boldsymbol{D_0})^{-1}\right]\boldsymbol{D_p}\mathbb{1}f^*_{\mathcal{S}_p}(s_2). \tag{206}$$

Proof. We can write the joint LST as the sum of the joint LST in the case that the server is idle at the start of the $(n+1)$th inter-departure time and the joint LST in the case it is not. Due to the probabilistic interpretation of the vector $\underline{v}^*_{k,n}(s_1)$, the LST for the case where the server is idle at the start of the $(n+1)$th inter-departure time is given by

$$\underline{v}^*_{k,n}(s_1)(s_2\boldsymbol{I}-\boldsymbol{D_0})^{-1}\boldsymbol{D_p}\mathbb{1}f^*_{\mathcal{S}_p}(s_2). \tag{207}$$

The vector $\underline{v}_0$ and $\underline{\alpha}-\underline{v}_0$ correspond to the cases where the 1st inter-departure time starts with and without an idle period, respectively. Hence, the term

$$\left[(\underline{\alpha}-\underline{v}_0)(-\boldsymbol{D_0})^{-1}+\underline{v}_0(s_1\boldsymbol{I}-\boldsymbol{D_0})^{-1}\right]\boldsymbol{D_k}\boldsymbol{P}^{n-1}\boldsymbol{P_p}\mathbb{1}f^*_{\mathcal{S}_k}(s_1)$$

in (206) holds the LST of the first inter-departure time when the $(n+1)$th inter-departure time involves a type p customer, denoted by $f^{(k,p)*}_{\mathcal{H}(n)}(s_1,0)$. This implies that

$$f^{(k,p)*}_{\mathcal{H}(n)}(s_1,0)-\underline{v}^*_{k,n}(s_1)(-\boldsymbol{D_0})^{-1}\boldsymbol{D_p}\mathbb{1}$$

holds the LST of the first inter-departure time when the $(n+1)$th inter-departure time involves a type p customer in case the server is not idle at the start of the $(n+1)$th inter-departure time. If the server is not idle at the start of the $(n+1)$th inter-departure time, its LST is given by $f^*_{\mathcal{S}_p}(s_2)$, as it is equal to the LST of the service time of the $(n+1)$th customer, the type of which is p. Hence, the joint LST in case the server is busy at the start of the $(n+1)$th inter-departure time is given by

$$f^{(k,p)*}_{\mathcal{H}(n)}(s_1,0)f^*_{\mathcal{S}_p}(s_2)-\underline{v}^*_{k,n}(s_1)(-\boldsymbol{D_0})^{-1}\boldsymbol{D_p}\mathbb{1}f^*_{\mathcal{S}_p}(s_2). \tag{208}$$

Combining (207) and (208) establishes (206). □

6.4.4 *The inter-departure time distribution and lag-1 joint moments*

In this section we determine an expression for the moments of the inter-departure time distribution as well as the joint lag-1 moments via Theorem 23. Based on the lag-1 moments it is possible to plug the MMAP[K]/PH[K]/1 FCFS queue into the queueing network analysis framework introduced in Chapter 8.

We start by defining $(\underline{v}_1^{(k)})_j$, for $j \in \{1,\ldots,N_A\}$ and $k \in \{1,\ldots,K\}$, as the probability that an arbitrary departing customer leaves a single customer behind, the type of which is k, while the MMAP phase at the departure epoch is j. Denote $\underline{v}_1^{(k)}$ as the vector with entry j equal to $(\underline{v}_1^{(k)})_j$.

Lemma 6. *The $1\times N_A$ vectors $\underline{v}_1^{(k)}$ can be computed as $\underline{v}_1^{(k)} = \rho\underline{a}(0)\boldsymbol{Y_1^{(k)}}/\lambda$, where the matrices $\boldsymbol{Y_1^{(k)}}$, for $k=1,\ldots,K$, are the unique solutions to the Sylvester matrix equations*

$$\boldsymbol{T}\boldsymbol{Y_1^{(k)}}+\boldsymbol{Y_1^{(k)}}\boldsymbol{D_0} = -\boldsymbol{Y_0}\boldsymbol{D_k}. \tag{209}$$

Proof. A departure leaves a single (type k) customer behind if the MMAP generates a single (type k) arrival during the sojourn time of the departing customer. By conditioning on the arrival time of this type k customer we get

$$\underline{v}_1^{(k)} = \int_{x=0}^{\infty} \underline{a}_D(x) \int_{a=0}^{x} e^{\boldsymbol{D}_0 a} \boldsymbol{D}_k e^{\boldsymbol{D}_0 (x-a)} da\, dx, \tag{210}$$

which yields (due to Theorem 37)

$$\underline{v}_1^{(k)} = \frac{\rho \underline{a}(0)}{\lambda} \int_{x=0}^{\infty} e^{\boldsymbol{T}x} (-\boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I}) \boldsymbol{I}_2 e^{\begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{D}_k \\ & \boldsymbol{D}_0 \end{bmatrix} x} \boldsymbol{J}_2 dx,$$

due to (195). Hence, due to Theorem 36, $\underline{v}_1^{(k)} = \rho \underline{a}(0) \boldsymbol{X}^{(k)} \boldsymbol{J}_2 / \lambda$, where the $N_S \times 2N_A$ matrix $\boldsymbol{X}^{(k)}$ is the unique solution of the Sylvester matrix equation

$$\boldsymbol{T}\boldsymbol{X}^{(k)} + \boldsymbol{X}^{(k)} \begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{D}_k \\ & \boldsymbol{D}_0 \end{bmatrix} = \underbrace{(\boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I}) \boldsymbol{I}_2}_{\begin{bmatrix} \boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}}. \tag{211}$$

Due to (198), it is easy to verify that $\boldsymbol{X}^{(k)} = [\boldsymbol{Y}_0 \ \boldsymbol{Y}_1^{(k)}]$ satisfies (211) if $\boldsymbol{Y}_1^{(k)}$ is the unique solution of (209). □

Theorem 24. *The LST $f_{\mathcal{H}}^*(s)$ of the inter-departure time distribution is given by*

$$f_{\mathcal{H}}^*(s) = \left[(\underline{\alpha} - \underline{v}_0)(-\boldsymbol{D}_0)^{-1} + \underline{v}_0 (s\boldsymbol{I} - \boldsymbol{D}_0)^{-1} \right] \left(\sum_{k=1}^{K} \boldsymbol{D}_k \mathbb{1} f_{\mathcal{S}_k}^*(s) \right). \tag{212}$$

The joint LST $f_{\mathcal{H}}^{(k)*}(s_1, s_2)$ of two consecutive inter-departure times where the type of the first customer is k, can be expressed as

$$\begin{aligned} f_{\mathcal{H}}^{(k)*}(s_1, s_2) = {} & \left[(\underline{\alpha} - \underline{v}_0)(-\boldsymbol{D}_0)^{-1} + \underline{v}_0 (s_1\boldsymbol{I} - \boldsymbol{D}_0)^{-1} \right] \boldsymbol{D}_k \cdot \\ & \left(\sum_{p=1}^{K} \boldsymbol{P}_p \mathbb{1} f_{\mathcal{S}_p}^*(s_2) \right) f_{\mathcal{S}_k}^*(s_1) + \left[\underline{v}_0 (s_1\boldsymbol{I} - \boldsymbol{D}_0)^{-1} \boldsymbol{D}_k + \underline{v}_1^{(k)} \right] \boldsymbol{M}_{\boldsymbol{k,1}}^*(s_1) \cdot \\ & \left((s_2\boldsymbol{I} - \boldsymbol{D}_0)^{-1} - (-\boldsymbol{D}_0)^{-1} \right) \left(\sum_{p=1}^{K} \boldsymbol{D}_p \mathbb{1} f_{\mathcal{S}_p}^*(s_2) \right). \end{aligned} \tag{213}$$

Proof. As $f_{\mathcal{H}}^*(s) = \sum_{p=1}^{K} \sum_{k=1}^{K} f_{\mathcal{H}(n)}^{(k,p)*}(s, 0)$, (212) follows from (206). To establish (213), it suffices to sum (206) over p for $n = 1$ and to note that

$$\underline{v}_{k,1}^*(s) = (\underline{v}_0 (s\boldsymbol{I} - \boldsymbol{D}_0)^{-1} \boldsymbol{D}_k + \underline{v}_1^{(k)}) \boldsymbol{M}_{\boldsymbol{k,1}}^*(s),$$

due to the probabilistic interpretation of $\underline{v}_0$, $\underline{v}_1^{(k)}$, $\boldsymbol{M}_{\boldsymbol{k,1}}(t)$ and $\underline{v}_{k,1}(t)$. The above equality can also be proven algebraically as follows. Combining (205) and (204) yields

$$\begin{aligned} \underline{v}_{k,1}^*(s) = {} & \frac{\rho \underline{a}(0)}{\lambda} \int_{x=0}^{\infty} e^{\boldsymbol{T}x} (-\boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I}) e^{\boldsymbol{D}_0 x} dx (s\boldsymbol{I} - \boldsymbol{D}_0)^{-1} \boldsymbol{D}_k \boldsymbol{M}_{\boldsymbol{k,1}}^*(s) + \\ & \frac{\rho \underline{a}(0)}{\lambda} \int_{x=0}^{\infty} e^{\boldsymbol{T}x} (-\boldsymbol{S}\mathbb{1} \otimes \boldsymbol{I}) \boldsymbol{I}_2 e^{\begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{D}_k \boldsymbol{M}_{\boldsymbol{k,1}}^*(s) \\ & \boldsymbol{D}_0 \end{bmatrix} x} \boldsymbol{J}_2 dx. \end{aligned}$$

Equation (197) and Lemma 5 imply that the first term reduces to

$$\underline{v}_0(s\boldsymbol{I}-\boldsymbol{D_0})^{-1}\boldsymbol{D}_k\boldsymbol{M}^*_{k,1}(s),$$

while the second equals $\rho\underline{a}(0)\boldsymbol{X}\boldsymbol{J_2}/\lambda$ (due to Theorem 36), with $\boldsymbol{X}$ the unique solution of

$$\boldsymbol{T}\boldsymbol{X}+\boldsymbol{X}\begin{bmatrix}\boldsymbol{D_0} & \boldsymbol{D}_k\boldsymbol{M}^*_{k,1}(s)\\ & \boldsymbol{D_0}\end{bmatrix}=(\boldsymbol{S}\mathbb{1}\otimes\boldsymbol{I})\boldsymbol{I_2}.$$

It is easy to verify that $\boldsymbol{X}=[\boldsymbol{Y_0}\ \ \boldsymbol{Y}^{(k)}_1\boldsymbol{M}^*_{k,1}(s)]$ provided that

$$\boldsymbol{T}\boldsymbol{Y}^{(k)}_1\boldsymbol{M}^*_{k,1}(s)+\boldsymbol{Y}^{(k)}_1\boldsymbol{M}^*_{k,1}(s)\boldsymbol{D_0}=-\boldsymbol{Y_0}\boldsymbol{D}_k\boldsymbol{M}^*_{k,1}(s).$$

As the matrix $\boldsymbol{D_0}$ commutes with $\boldsymbol{M}^*_{k,1}(s)$, this equation follows from (209) and we may conclude that $\rho\underline{a}(0)\boldsymbol{X}\boldsymbol{J_2}/\lambda=\underline{v}^{(k)}_1\boldsymbol{M}^*_{k,1}(s)$ as required. □

The nth moment of the inter-departure times is given by $E(\mathcal{H}^n)=(-1)^n\frac{d^n}{ds^n}f^*_{\mathcal{H}}(s)|_{s=0}$. Instead of computing the moments directly, we introduce the so-called reduced moments

$$\hat{E}(\mathcal{H}^n)=E(\mathcal{H}^n)/n!,\quad \hat{E}(\mathcal{S}^n_k)=E(\mathcal{S}^n_k)/n!,$$

because they make the forthcoming expressions simpler.

Corollary 3. *The nth reduced moment of the inter-departure time distribution is given by*

$$\hat{E}(\mathcal{H}^n)=\sum_{k=1}^{K}\left(\frac{\lambda_k}{\lambda}\hat{E}(\mathcal{S}^n_k)+\underline{v}_0\sum_{\ell=1}^{n}(-\boldsymbol{D_0})^{-\ell-1}\boldsymbol{D}_k\mathbb{1}\hat{E}(\mathcal{S}^{n-\ell}_k)\right),$$

Proof. As $E(\mathcal{H}^n)=(-1)^n\frac{d^n}{ds^n}f^*_{\mathcal{H}}(s)|_{s=0}$, (212) implies

$$\hat{E}(\mathcal{H}^n)=\sum_{k=1}^{K}\left((\underline{\alpha}-\underline{v}_0)(-\boldsymbol{D_0})^{-1}\boldsymbol{D}_k\mathbb{1}\frac{E(\mathcal{S}^n_k)}{n!}+\frac{\underline{v}_0}{n!}\sum_{\ell=0}^{n}\binom{n}{\ell}(\ell!)(-\boldsymbol{D_0})^{-\ell-1}\boldsymbol{D}_k\mathbb{1}E(\mathcal{S}^{n-\ell}_k)\right),$$

which establishes the result as

$$\underline{\alpha}(-\boldsymbol{D_0})^{-1}\boldsymbol{D}_k\mathbb{1}=\underline{\theta}\left(\sum_{s=1}^{K}\boldsymbol{D}_s\right)(-\boldsymbol{D_0})^{-1}\boldsymbol{D}_k\mathbb{1}/\lambda=\underline{\theta}\boldsymbol{D}_k\mathbb{1}/\lambda=\lambda_k/\lambda.$$

□

Taking the derivatives of $f^{(k)*}_{\mathcal{H}}(s_1,s_2)$ gives the joint moments of two consecutive inter-departure times. Again, for simplicity we use the reduced moments instead of the standard ones. The (n_1,n_2)th reduced joint moment is denoted by $\hat{\eta}^{(k)}_{n_1,n_2}$ and is obtained from the LST as

$$\hat{\eta}^{(k)}_{n_1,n_2}=\frac{(-1)^{n_1+n_2}}{n_1!n_2!}\frac{\partial^{n_1}}{\partial s_1^{n_1}}\frac{\partial^{n_2}}{\partial s_2^{n_2}}f^{(k)*}_{\mathcal{H}}(s_1,s_2)|_{s_1=0,s_2=0}.$$

Corollary 4. *The (n_1, n_2)th reduced joint moment of the inter-departure times are given by*

$$\hat{\eta}^{(k)}_{n_1,n_2} = \left[\underline{\alpha}\boldsymbol{P_k}\hat{E}(\mathcal{S}_k^{n_1}) + \underline{v}_0\sum_{\ell=1}^{n_1}(-\boldsymbol{D_0})^{-\ell}\boldsymbol{P_k}\hat{E}(\mathcal{S}_k^{n_1-\ell})\right]\left(\sum_{q=1}^{K}\boldsymbol{P_q}\mathbb{1}\hat{E}(S_q^{n_2})\right)$$
$$+\left[\underline{v}_1^{(k)}\bar{\boldsymbol{M}}^{\boldsymbol{n_1}}_{\boldsymbol{k,1}} + \underline{v}_0\sum_{\ell=0}^{n_1}(-\boldsymbol{D_0})^{-\ell}\boldsymbol{P_k}\bar{\boldsymbol{M}}^{\boldsymbol{n_1-\ell}}_{\boldsymbol{k,1}}\right]\sum_{d=1}^{n_2}(-\boldsymbol{D_0})^{-d}\left(\sum_{q=1}^{K}\boldsymbol{P_q}\mathbb{1}\hat{E}(S_q^{n_2-d})\right), \tag{214}$$

where $\bar{\boldsymbol{M}}^{\boldsymbol{n}}_{\boldsymbol{k,1}}$ is defined and computed as follows:

$$\bar{\boldsymbol{M}}^{\boldsymbol{n}}_{\boldsymbol{k,1}} = \frac{(-1)^n}{n!}\frac{d^n}{ds^n}\boldsymbol{M}^{*}_{\boldsymbol{k,1}}(s)|_{s=0} = (\underline{\sigma}_k\otimes\boldsymbol{I})((-\boldsymbol{S_k})\oplus\boldsymbol{D_0})^{-n-1}(-\boldsymbol{S_k}\mathbb{1}\otimes\boldsymbol{I}). \tag{215}$$

dc_1412_17

7

ANALYSIS OF THE MMAP[K]/PH[K]/1 PRIORITY QUEUE

In the MMAP[K]/PH[K]/1 queue K types (classes) of customers are distinguished. The arrival process of customers is described by a MMAP, and the service times are PH distributed. There is a single server, which always picks the customer having the highest priority for service. If the ongoing service can not be interrupted when a higher priority customer arrives, the service is called to be *non-preemptive*. In the *preemptive resume* case (also referred to as the preemptive case for simplicity), however, the service of customers can be interrupted, and resumed later when all higher priority customers leave the system.

To introduce the analysis approach, the two-class case ($K = 2$) is considered throughout the chapter, and the extension to the general case ($K > 2$) is provided in [49].

Similar to Chapter 6, the MMAP characterizing the arrivals is given by the size $N_A \times N_A$ matrices $\boldsymbol{D_0}$, $\boldsymbol{D_H}$ and $\boldsymbol{D_L}$ containing the rates of internal transitions and transitions accompanied by high and low priority customers. The mean arrival rate of high (low) priority customers is denoted by λ_H (λ_L), and it is calculated by $\lambda_H = \underline{\theta}\boldsymbol{D_H}\mathbb{1}$ ($\lambda_L = \underline{\theta}\boldsymbol{D_L}\mathbb{1}$), respectively, where $\underline{\theta}$ is the stationary distribution of the phase process of the MMAP (see Section 3.1.2).

The random variable representing the service times of the low priority customers $\mathcal{S}_L$ is PH distributed with N_L phases, characterized by $\underline{\sigma}_L, \boldsymbol{S_L}$ and $\underline{s}_L$. Row vector $\underline{\sigma}_L$ is the initial vector, matrix $\boldsymbol{S_L}$ is the transient generator and column vector $\underline{s}_L$ holds the transition rates to the absorbing state, thus $\underline{s}_L = -\boldsymbol{S_L}\mathbb{1}$. The mean service rate is $\mu_L = 1/E(\mathcal{S}_L)$. The PH distribution corresponding to the high priority service times and its properties are defined similarly, using subscript H instead of L.

The load of the queue is $\rho = \lambda_H/\mu_H + \lambda_L/\mu_L$. In this chapter $\rho < 1$ is assumed.

7.1 ANALYSIS OF THE PREEMPTIVE RESUME PRIORITY QUEUE

Priority queues are extensively studied since the middle of the last century [61], starting with the most basic variant with Poisson arrival process and exponentially distributed service times. In the last two decades most research activity on priority queues has considered more general arrival processes like MAPs or MMAPs.

In [78] the MAP/G/1 preemptive priority queue is analyzed based on the workload process, and the LST of the sojourn time distribution of the customers is derived. The non-preemptive case is investigated in [79] and [80], where the LST of the sojourn time, the moments of the sojourn time, the GF of the queue length, the queue length moments and the queue length probabilities are provided.

After this overview one may think that not too much has left to be done in the field of MAP driven priority queues. However, all the aforementioned results assume a general distribution for the service times, which makes the solution complex and often difficult to implement in a proper way (in the numerical sense). To address this issue the generally distributed service

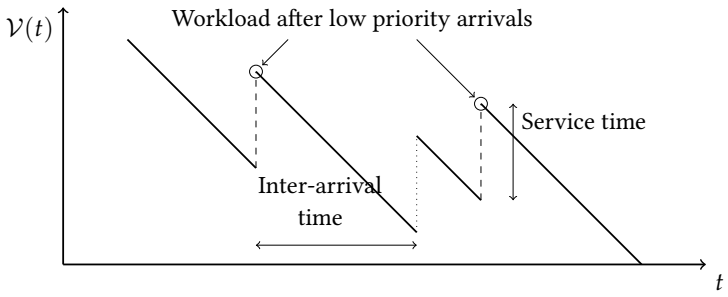


Figure 29.: The workload process of the priority queue

times can be replaced by PH distributed ones in the hope of the simpler and numerically more tractable solution.

In [3] the (discrete-time) MAP/PH/1 priority queue is considered by representing the state space with a QBD and exploiting the special structure of the related fundamental matrices. While this approach is elegant and seems promising, there are some computational bottlenecks (as pointed out in [48]). There have been efforts to make it more efficient (see [45] and [48]), but apart from the queue length moments all performance measures can be computed only in case of a very limited number of phases.

Our approach is based on the analysis of the workload process, just like [78] in the context of MAP/G/1 preemptive priority queues. However, by exploiting the technical simplicity of the PH distributed service times we are able to arrive to a more intuitive, simpler to implement and numerically more beneficial solution.

7.1.1 *The workload of the system just after low priority arrival instants*

For the analysis of the sojourn time we first need to derive the distribution of the workload a low priority arrival finds in the system (see Section 5.2.2).

The workload process $\{\mathcal{V}(t), t > 0\}$ is the amount of work in the system at time t, thus the time needed to process all the customers in the queue if the arrival process is frozen. $\mathcal{V}(t)$ decreases by a slope of one between the arrival epochs, and jumps up at arrival epochs according to the service time requirement of the arrival; thus, $\mathcal{V}(t)$ is skip-free to the left. An example to the workload process is depicted in Figure 29. As we have two customer classes, there are two kinds of jumps in the figure, the dotted one corresponds to the high, the dashed one to the low priority customers.

To completely characterize the situation an arriving low priority customer finds in the system, the stationary solution of $\{\mathcal{V}(t), \mathcal{J}(t)\}$, thus the joint distribution of the workload and the MMAP phase needs to be derived.

In our case the inter-arrival times are given by a MMAP and the size of the jumps is PH distributed, which makes it possible to apply the method of [28] to transform $\mathcal{V}(t)$, which is skip-free to the left, to $\mathcal{L}(t)$, which is skip-free to both directions (like it was done in the single class case in Section 5.2.2). More precisely, the continuous process with jumps $\{\mathcal{V}(t), \mathcal{J}(t)\}$, is transformed to $\{\mathcal{L}(t), \mathcal{Z}(t)\}$ from which the stationary distribution of $\{\mathcal{V}(t), \mathcal{J}(t)\}$ at low priority arrivals is computed.

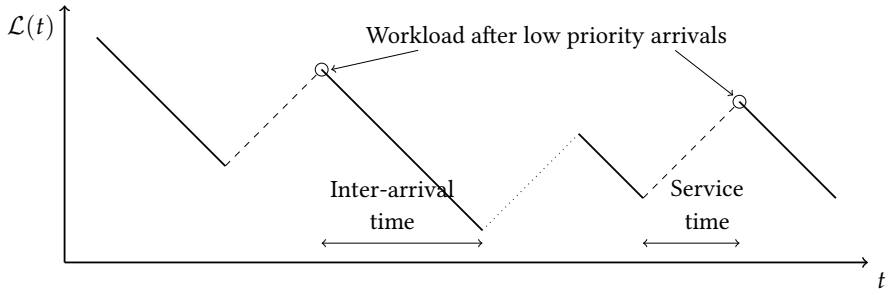


Figure 30.: The modified workload process of the queue

The transformation to the skip-free process is performed as follows. Let $\{\mathcal{L}(t), \mathcal{Z}(t)\}$ be a canonical MFM where $\mathcal{Z}(t)$ is the underlying CTMC with generator matrix $\boldsymbol{Q}$ given by

$$\begin{aligned} \boldsymbol{Q}_{++} &= \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{S_L} & \\ & \boldsymbol{I} \otimes \boldsymbol{S_H} \end{bmatrix}, \qquad \boldsymbol{Q}_{+-} = \begin{bmatrix} \boldsymbol{I} \otimes \underline{s}_L \\ \boldsymbol{I} \otimes \underline{s}_H \end{bmatrix}, \\ \boldsymbol{Q}_{-+} &= \begin{bmatrix} \boldsymbol{D_L} \otimes \underline{\sigma}_L & \boldsymbol{D_H} \otimes \underline{\sigma}_H \end{bmatrix}, \qquad \boldsymbol{Q}_{--} = \boldsymbol{D_0}. \end{aligned} \tag{216}$$

This fluid model behaves like $\{\mathcal{V}(t), \mathcal{J}(t)\}$ between arrivals, when it stays in the negative states $\mathcal{N}_-$. Whenever an arrival occurs, however, it switches to one of the positive state groups (depending on the class of the entering customer), and accumulates the workload increment with a slope of 1. Thus, the jumps are eliminated and replaced by progressive workload accumulations. The transformed process obtained from Figure 29 is depicted in Figure 30.

Observe that the joint stationary density of the workload and the MMAP phase at low priority arrivals are the same in the original and in the transformed process. The stationary solution $\underline{\pi}(x)$ of the transformed process (that is a canonical MFM) is given by Theorem 17, from which, by embedding at just after low priority arrivals we get a matrix-exponential solution

$$\begin{aligned} \underline{\hat{\pi}}(x) &= \frac{1}{\hat{c}} \underline{\pi}(x) \begin{bmatrix} \boldsymbol{I} \otimes \underline{s}_L \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} = \frac{1}{\hat{c}} \underline{p}_- \boldsymbol{Q}_{-+} e^{\boldsymbol{K}x} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Psi} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \otimes \underline{s}_L \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \\ &= \underbrace{\frac{1}{\hat{c}} \underline{p}_- \boldsymbol{Q}_{-+}}_{\underline{\hat{\beta}}} e^{\boldsymbol{K}x} \underbrace{\begin{bmatrix} \boldsymbol{I} \otimes \underline{s}_L \\ \boldsymbol{0} \end{bmatrix}}_{\hat{\boldsymbol{B}}} = \underline{\hat{\beta}} e^{\boldsymbol{K}x} \hat{\boldsymbol{B}}, \end{aligned} \tag{217}$$

where the normalization constant is $\hat{c} = \underline{p}_- \boldsymbol{Q}_{-+} (-\boldsymbol{K})^{-1} \hat{\boldsymbol{B}} \mathbb{1}$. Notice that from the three blocks in the last matrix term the upper two belong to $\mathcal{N}_+$ and the lower belongs to $\mathcal{N}_-$.

Due to technical reasons (which will be discussed later) the representation given by (217) will not be appropriate in the forthcoming derivations, because in general $\boldsymbol{K}\mathbb{1} + \hat{\boldsymbol{B}}\mathbb{1} \neq \underline{0}$. The following theorem provides the representation transformation that ensures the proper row-sums.

Theorem 25. *The joint density of the workload and the phase probability of the MMAP just after low priority arrivals $\underline{\hat{\pi}}(x)$ can be obtained by*

$$\underline{\hat{\pi}}(x) = \underline{\hat{\beta}}' e^{\boldsymbol{K}'x} \hat{\boldsymbol{B}}', \tag{218}$$

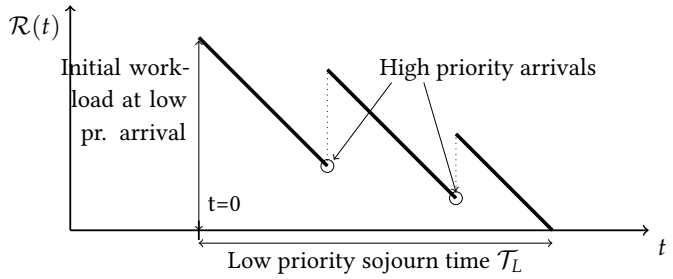


Figure 31.: The remaining sojourn time of a low priority customer

with $\hat{\underline{\beta}}' = \hat{\underline{\beta}} \cdot diag\langle\underline{\Delta}\rangle, \boldsymbol{K'} = diag\langle\underline{\Delta}\rangle^{-1} \cdot \boldsymbol{K} \cdot diag\langle\underline{\Delta}\rangle$ *and* $\hat{\boldsymbol{B}}' = diag\langle\underline{\Delta}\rangle^{-1} \cdot \hat{\boldsymbol{B}}$, *where* $\underline{\Delta} = (-\boldsymbol{K})^{-1}\hat{\boldsymbol{B}}\mathbb{1}$. *Furthermore, we have that*

$$\boldsymbol{K'}\mathbb{1} + \hat{\boldsymbol{B}}'\mathbb{1} = \underline{0}. \tag{219}$$

Proof. The fact that (218) equals (217) can be proven by

$$\begin{aligned}\hat{\underline{\pi}}(x) &= \hat{\underline{\beta}}' e^{\boldsymbol{K'}x}\hat{\boldsymbol{B}}' = \hat{\underline{\beta}} \cdot \mathrm{diag}\langle\underline{\Delta}\rangle \cdot e^{\mathrm{diag}\langle\underline{\Delta}\rangle^{-1}\cdot\boldsymbol{K}\cdot\mathrm{diag}\langle\underline{\Delta}\rangle x}\mathrm{diag}\langle\underline{\Delta}\rangle^{-1} \cdot \hat{\boldsymbol{B}} \\ &= \hat{\underline{\beta}} \cdot \mathrm{diag}\langle\underline{\Delta}\rangle \cdot \mathrm{diag}\langle\underline{\Delta}\rangle^{-1} \cdot e^{\boldsymbol{K}x} \cdot \mathrm{diag}\langle\underline{\Delta}\rangle\mathrm{diag}\langle\underline{\Delta}\rangle^{-1} \cdot \hat{\boldsymbol{B}} = \hat{\underline{\beta}} e^{\boldsymbol{K}x}\hat{\boldsymbol{B}}.\end{aligned} \tag{220}$$

To prove that (219) holds we have

$$\boldsymbol{K'}\mathbb{1} + \hat{\boldsymbol{B}}'\mathbb{1} = \mathrm{diag}\langle\underline{\Delta}\rangle^{-1}(\boldsymbol{K}(-\boldsymbol{K})^{-1}\hat{\boldsymbol{B}}\mathbb{1} + \hat{\boldsymbol{B}}\mathbb{1}) = \underline{0}. \tag{221}$$

□

7.1.2 *The sojourn time of low priority customers*

For the sojourn time analysis of low priority customers we introduce the remaining sojourn time process $\{\mathcal{R}(t), t \geq 0\}$. At $t = 0$, $\mathcal{R}(t)$ is the workload seen by a low priority customer when it arrives. For $t > 0$, $\mathcal{R}(t)$ decreases by a slope of one till a high priority arrival occurs, when $\mathcal{R}(t)$ has a jump with size given by a high priority service time. When $\mathcal{R}(t)$ reaches zero, it remains zero and the corresponding low priority customer leaves the system (see Figure 31). Hence, the sojourn time of low priority customers $\mathcal{T}_L$ is

$$\mathcal{T}_L = \inf\{t > 0 : \mathcal{R}(t) = 0\}. \tag{222}$$

Just like the workload process $\mathcal{V}(t)$, the remaining sojourn time process $\mathcal{R}(t)$ is skip-free to the left and has upward jumps. As we did with the workload process, we transform $\mathcal{R}(t)$ to a skip-free process which is easier to handle numerically, and derive the properties of $\mathcal{R}(t)$ from the transformed process.

Let us introduce a canonical fluid model $\{\tilde{\mathcal{R}}(t), \tilde{\mathcal{Z}}(t)\}$ where the generator $\tilde{\boldsymbol{Q}}$ of the underlying CTMC is defined by

$$\begin{aligned}\tilde{\boldsymbol{Q}}_{++} &= \begin{bmatrix}\boldsymbol{K'} & \\ & \boldsymbol{I} \otimes \boldsymbol{S_H}\end{bmatrix}, \quad \tilde{\boldsymbol{Q}}_{+-} = \begin{bmatrix}\hat{\boldsymbol{B}}' \\ \boldsymbol{I} \otimes \underline{s}_H\end{bmatrix}, \\ \tilde{\boldsymbol{Q}}_{-+} &= \begin{bmatrix}\boldsymbol{0} & \boldsymbol{D_H} \otimes \underline{\sigma}_H\end{bmatrix}, \quad \tilde{\boldsymbol{Q}}_{--} = \boldsymbol{D_0} + \boldsymbol{D_L},\end{aligned} \tag{223}$$

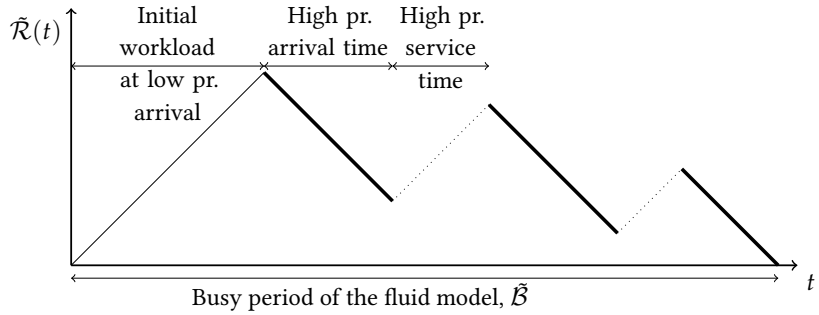


Figure 32.: The fluid model for the sojourn time analysis

furthermore, let the distribution of $\tilde{\mathcal{Z}}(t)$ at $t = 0$ be

$$\underline{\tilde{\kappa}} = \{P(\tilde{\mathcal{Z}}_+(0) = i)\} = \begin{bmatrix} \underline{\hat{\beta}}' & 0 \end{bmatrix}. \tag{224}$$

This fluid model has three state groups: there are two state groups in $\mathcal{N}_+$, and $\mathcal{N}_-$ is the third one.

The role of the first state group is the accumulation of the initial workload, experienced by a low priority customer when it enters the system. Observe that the sojourn time density of the first state group, when started from $\underline{\tilde{\kappa}}$, is exactly $\underline{\hat{\pi}}(x)$, which is the density of the initial workload. The second group of states is activated when an arrival occurs, and the corresponding workload increment is accumulated. The third group of states, the negative ones represent the periods when the server is processing the low priority workload and is decreasing the remaining sojourn time of the tagged low priority customer.

Note that due to Theorem 25 the usual property of Markovian generators $\tilde{\boldsymbol{Q}}\mathbb{1} = \underline{0}$ holds. The correctness of the solution with the non-Markovian components $\boldsymbol{K}'$ and $\hat{\boldsymbol{B}}'$ is ensured by [20].

The main idea in this section is that, by construction, the relation between the duration of the busy period $\tilde{\mathcal{B}}$ of the fluid model characterized by $(\underline{\tilde{\kappa}}, \tilde{\boldsymbol{Q}})$ and the sojourn time of low priority customers $\mathcal{T}_L$ is

$$\mathcal{T}_L = \tilde{\mathcal{B}}/2. \tag{225}$$

This relation is clearly visible when looking at Figures 31 and 32.

Finally, the following corollary expresses the properties of the sojourn time with the properties of the busy period (detailed in Section 4.2.3).

Corollary 5. *The distribution of $\mathcal{T}_L$ in time domain, in LST domain, and its moments can be expressed by*

$$F_{\mathcal{T}_L}(t) = P(\mathcal{T}_L < t) = F_{\tilde{\mathcal{B}}}(2t), \tag{226}$$

$$f^*_{\mathcal{T}_L}(s) = E(e^{-s\mathcal{T}_L}) = f^*_{\tilde{\mathcal{B}}}(s/2), \tag{227}$$

$$E(\mathcal{T}_L^k) = E(\tilde{\mathcal{B}}^k)/2^k. \tag{228}$$

7.1.3 *Number of low priority customers in the system*

First we derive the distribution of the number of low priority customers at low priority departure epochs (the corresponding random variable is denoted by $\mathcal{X}_L$), then the one at a random point in time (denoted by $\mathcal{Y}_L$).

When a low priority customer leaves the system, the number of customers behind it equals the number of low priority arrivals during its sojourn in the system. To analyze this quantity, let us go back to the remaining sojourn time introduced in Section 7.1.2, and modify the background process of the related fluid model such that it counts the number of low priority arrivals. Instead of $\tilde{Q}$ we get $\tilde{Q}'$ defined by

$$\tilde{Q}' = \begin{bmatrix} F_0 & F_1 & & \\ & F_0 & F_1 & \\ & & F_0 & \ddots \\ & & & \ddots \end{bmatrix}, \tag{229}$$

where matrices F_0 and F_1 are

$$F_0 = \begin{bmatrix} \tilde{Q}_{++} & \tilde{Q}_{+-} \\ \tilde{Q}_{-+} & D_0 \end{bmatrix}, \qquad F_1 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & D_L \end{bmatrix}. \tag{230}$$

With this generator, matrix $\tilde{\Psi}'$ of the corresponding canonical Markovian fluid model has an upper block-Toeplitz structure like

$$\tilde{\Psi}' = \begin{bmatrix} \tilde{\Psi}_0 & \tilde{\Psi}_1 & \tilde{\Psi}_2 & \cdots \\ & \tilde{\Psi}_0 & \tilde{\Psi}_1 & \cdots \\ & & \tilde{\Psi}_0 & \cdots \\ & & & \ddots \end{bmatrix}, \tag{231}$$

where the entry $(\tilde{\Psi}_i)_{k,\ell}$ is the probability that i low priority arrivals occur during the sojourn time of a low priority customer and the phase of the MMAP is ℓ at the departure given that the phase was k when it entered the system.

The reason of the upper block-Toeplitz structure is that the number of low-priority arrivals during the sojourn time can only increase, and that the MMAP generating the arrivals is independent of the queue length.

Theorem 26. *Matrix $\tilde{\Psi}_0$ is the solution to the NARE*

$$\tilde{\Psi}_0 \tilde{Q}_{-+} \tilde{\Psi}_0 + \tilde{\Psi}_0 D_0 + \tilde{Q}_{++} \tilde{\Psi}_0 + \tilde{Q}_{+-} = \mathbf{0}, \tag{232}$$

and for $i > 0$ matrices $\tilde{\Psi}_i$ can be obtained recursively by solving the Sylvester equation

$$(\tilde{Q}_{++} + \tilde{\Psi}_0 \tilde{Q}_{-+}) \tilde{\Psi}_i + \tilde{\Psi}_i (D_0 + \tilde{Q}_{-+} \tilde{\Psi}_0) = -\tilde{\Psi}_{i-1} D_L - \sum_{j=1}^{i-1} \tilde{\Psi}_j \tilde{Q}_{-+} \tilde{\Psi}_{i-j}. \tag{233}$$

Proof. Let us partition matrix $\tilde{Q}'$ according to the positive and negative states. We get

$$\tilde{Q}'' = \begin{bmatrix} \tilde{Q}''_{++} & \tilde{Q}''_{+-} \\ \tilde{Q}''_{-+} & \tilde{Q}''_{--} \end{bmatrix} = \left[\begin{array}{ccc|ccc} \tilde{Q}_{++} & & & \tilde{Q}_{+-} & & \\ & \tilde{Q}_{++} & & & \tilde{Q}_{+-} & \\ & & \ddots & & & \ddots \\ \hline \tilde{Q}_{-+} & & & D_0 & D_L & \\ & \tilde{Q}_{-+} & & & D_0 & D_L \\ & & \ddots & & & \ddots \end{array}\right]. \tag{234}$$

Substituting (234) and (231) into the NARE (135) provides the theorem after some algebraic manipulation. □

The probabilities for the number of low priority customers at low priority departures $x_i^L = P(\mathcal{X}_L = i)$ are obtained from $\tilde{\mathbf{\Psi}}_i$ by taking into consideration the initial probability vector of the busy period $\underline{\tilde{\kappa}}$. For later use, we also introduce row vector $\underline{x}_i^L = \{P(\mathcal{X}_L = i, \mathcal{J} = j), j = 1, \ldots, N_A\}$, the joint probability of the number of customers and the phase of the MMAP at departures (obviously, $x_i^L = \underline{x}_i^L \mathbb{1}$).

Corollary 6. *For the distribution of the number of low priority customers at low priority departures we have*

$$\underline{x}_i^L = \underline{\tilde{\kappa}} \tilde{\mathbf{\Psi}}_i. \tag{235}$$

The significance of (235) lies in the fact that the consecutive queue length probabilities can be obtained by consecutive solutions of Sylvester equations calculating $\tilde{\mathbf{\Psi}}_i$. The prior procedures of the related literature are far more expensive computationally.

Corollary 7. *The GF of the distribution of the number of customers at departures* $\underline{X}_L(z) = \sum_{i=0}^{\infty} z^i \underline{x}_i^L$ *can be obtained by*

$$\underline{X}_L(z) = \underline{\tilde{\kappa}} \tilde{\mathbf{\Psi}}(z), \tag{236}$$

where matrix $\tilde{\mathbf{\Psi}}(z)$ *satisfies the NARE*

$$\tilde{\mathbf{\Psi}}(z)\tilde{\boldsymbol{Q}}_{-+}\tilde{\mathbf{\Psi}}(z) + \tilde{\mathbf{\Psi}}(z)(\boldsymbol{D_0} + z\boldsymbol{D_L}) + \tilde{\boldsymbol{Q}}_{++}\tilde{\mathbf{\Psi}}(z) + \tilde{\boldsymbol{Q}}_{+-} = \mathbf{0}. \tag{237}$$

Proof. Multiplying (233) by z^i, summing it from 1 to infinity, then adding (232) provides (237). □

Finally, the factorial moments of $\mathcal{X}_L$ can be calculated by taking the derivatives of the generating function, hence

$$E(\mathcal{X}_L^k) = \sum_{i=0}^{\infty} i^k x_i^L = \frac{d^k}{dz^k} \underline{X}_L(z)|_{z=1} \mathbb{1}, \tag{238}$$

yielding a recursion introduced by the next corollary.

Corollary 8. *For the kth factorial moment of* $\mathcal{X}_L$ *we have*

$$\underline{E(\mathcal{X}_L^k)} = \underline{\tilde{\kappa}} \tilde{\mathbf{\Psi}}^{(k)}, \quad E(\mathcal{X}_L^k) = \underline{E(\mathcal{X}_L^k)} \mathbb{1}, \tag{239}$$

where $\tilde{\mathbf{\Psi}}^{(k)} = \frac{d^k}{dz^k} \tilde{\mathbf{\Psi}}(z)|_{z=1}$. *Matrix* $\tilde{\mathbf{\Psi}}^{(0)} = \tilde{\mathbf{\Psi}}$ *and for* $k > 0$ *matrices* $\tilde{\mathbf{\Psi}}^{(k)}$ *are obtained recursively by solving the following Sylvester equations*

$$\begin{aligned} &(\tilde{\boldsymbol{Q}}_{++} + \tilde{\mathbf{\Psi}}^{(0)}\tilde{\boldsymbol{Q}}_{-+})\tilde{\mathbf{\Psi}}^{(k)} + \tilde{\mathbf{\Psi}}^{(k)}(\tilde{\boldsymbol{Q}}_{--} + \tilde{\boldsymbol{Q}}_{-+}\tilde{\mathbf{\Psi}}^{(0)}) \\ &\qquad = -k\tilde{\mathbf{\Psi}}^{(k-1)}\boldsymbol{D_L} - \sum_{i=1}^{k-1} \binom{k}{i} \tilde{\mathbf{\Psi}}^{(i)} \tilde{\boldsymbol{Q}}_{-+} \tilde{\mathbf{\Psi}}^{(k-i)}. \end{aligned} \tag{240}$$

In the rest of the section we calculate various properties of the number of low priority customers at random point in time denoted by $\mathcal{Y}_L$. Our contribution in this subsection ends here, since the relations between $\mathcal{X}_L$ and $\mathcal{Y}_L$ are extensively studied in [79], that we provide here for the sake of completeness.

Let us introduce row vector $\underline{y}_i^L = \{P(\mathcal{Y}_L = i, \mathcal{J} = j), j = 1, \ldots, N_A\}$.

Theorem 27. *([79], Theorem 4.6) The GF of $\underline{y}_i^L$, denoted by $\underline{Y}_L(z) = \sum_{i=0}^{\infty} z^i \underline{y}_i^L$ is related to $\underline{X}_L(z)$ as*

$$\underline{Y}_L(z)(\boldsymbol{D_0} + \boldsymbol{D_H} + z\boldsymbol{D_L}) = \lambda_L(z-1)\underline{X}_L(z). \tag{241}$$

Corollary 9. *([79], Corollary 3.11) Vectors $\underline{y}_i^L, i \geq 0$ are recursively obtained by*

$$\begin{aligned} \underline{y}_0^L &= \lambda_L \underline{x}_0^L(-\boldsymbol{D_0} - \boldsymbol{D_H})^{-1}, \\ \underline{y}_i^L &= (\underline{y}_{i-1}^L \boldsymbol{D_L} + \lambda_L \underline{x}_i^L - \lambda_L \underline{x}_{i-1}^L)(-\boldsymbol{D_0} - \boldsymbol{D_H})^{-1}, \qquad i > 0. \end{aligned} \tag{242}$$

Corollary 10. *([79], Corollary 3.10) The factorial moments of the number of low priority customers at random point in time are obtained recursively as*

$$\begin{aligned} E(\mathcal{Y}_L^k) &= E(\mathcal{X}_L^k) + k\big(\underline{E(\mathcal{X}_L^{k-1})} - \underline{E(\mathcal{Y}_L^{k-1})}\boldsymbol{D_L}/\lambda_L\big)(\mathbb{1}\underline{\theta} - \boldsymbol{D})^{-1}\boldsymbol{D_L}\mathbb{1}, \\ \underline{E(\mathcal{Y}_L^k)} &= E(\mathcal{Y}_L^k)\underline{\theta} + k\big(\underline{E(\mathcal{Y}_L^{k-1})}\boldsymbol{D_L} - \lambda_L \underline{E(\mathcal{X}_L^{k-1})}\big)(\mathbb{1}\underline{\theta} - \boldsymbol{D})^{-1}, \end{aligned} \tag{243}$$

for $k > 0$, and $\underline{E(\mathcal{Y}_L^0)} = \underline{\theta}$.

7.1.4 *The analysis of the high priority class*

In case of the preemptive resume service policy the high priority class can be analyzed in separation, as a single-class MAP/PH/1 queue with arrival process given by matrices $(\boldsymbol{D_0} + \boldsymbol{D_L}, \boldsymbol{D_H})$ and service time distribution given by $(\underline{\sigma}_H, \boldsymbol{S_H})$. The details of the analysis of this queue can be found in Chapter 5.

7.2 ANALYSIS OF THE NON-PREEMPTIVE PRIORITY QUEUE

In the non-preemptive case the service of a low priority customer can not be interrupted. It turns out that the analysis approach developed in Section 7.1 can still be used with a small difference. Instead of analyzing the sojourn time and the number of customers in the system, in the non-preemptive case we will focus on the *waiting time* (which can be interrupted by a high priority arrival any time) and the *number of waiting customers* in the system. The non-interruptible service time and the number of arrivals during it will be added afterwards to obtain the sojourn time and the number of customers in the system.

7.2.1 *The workload of the system just before low priority arrival instants*

When a low priority customer enters the system, its waiting time equals the workload of the system just before its arrival (thus without its own service time) plus the service times of all high priority customers arrived during waiting in the queue. To find out the workload just before the arrival in the example of Figure 29 this means that we need the distribution of $\mathcal{V}(t)$ just before the jumps, instead of just after the jumps.

This distribution can be obtained by applying the same transformation procedure which results in a canonical Markovian fluid model with stationary fluid density $\underline{\pi}(x)$ and probability mass at level zero $\underline{p}$. Embedding right before low priority arrivals we get the density

$$\begin{aligned}\check{\underline{\pi}}(x) &= \frac{1}{\check{c}}\underline{\pi}(x)\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \boldsymbol{D_L}\end{bmatrix} = \frac{1}{\check{c}}\underline{p}_-\boldsymbol{Q}_{-+}e^{\boldsymbol{K}x}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{\Psi}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \boldsymbol{D_L}\end{bmatrix} \\ &= \underbrace{\frac{1}{\check{c}}\underline{p}_-\boldsymbol{Q}_{-+}}_{\check{\underline{\beta}}}e^{\boldsymbol{K}x}\underbrace{\boldsymbol{\Psi}\boldsymbol{D_L}}_{\check{\boldsymbol{B}}} = \check{\underline{\beta}}e^{\boldsymbol{K}x}\check{\boldsymbol{B}}.\end{aligned} \tag{244}$$

Notice that the workload just before the arrival can be exactly zero as well, with probability mass

$$\check{\underline{p}} = \frac{1}{\check{c}}\underline{p}_-\boldsymbol{D_L}. \tag{245}$$

The normalization constant is $\check{c} = \underline{p}_-\boldsymbol{D_L}\mathbb{1} + \underline{p}_-\boldsymbol{Q}_{-+}(-\boldsymbol{K})^{-1}\check{\boldsymbol{B}}\mathbb{1}$.

Similar to Theorem 25, it is again possible to similarity transform the representation $\check{\underline{\beta}}, \boldsymbol{K}$ and $\check{\boldsymbol{B}}$ to $\check{\underline{\beta}}', \boldsymbol{K}'$ and $\check{\boldsymbol{B}}'$ such that $\boldsymbol{K}'\mathbb{1} + \check{\boldsymbol{B}}'\mathbb{1} = \underline{0}$ holds.

7.2.2 *The sojourn time of low priority customers*

As mentioned before, first the waiting time (denoted by $\mathcal{W}_L$) is characterized, then the service time is added afterwards to get the sojourn time.

As done in Section 7.1.2, it is possible to introduce the remaining waiting time process $\mathcal{W}(t)$ and construct a canonical fluid model $\{\bar{\mathcal{W}}(t), \bar{\mathcal{Z}}(t)\}$ whose busy period $\bar{\mathcal{B}}$ is closely related to the waiting time. The blocks of the generator of this fluid model are

$$\bar{\boldsymbol{Q}}_{++} = \begin{bmatrix}\boldsymbol{K}' & \\ & \boldsymbol{I}\otimes\boldsymbol{S_H}\end{bmatrix}, \quad \bar{\boldsymbol{Q}}_{+-} = \begin{bmatrix}\check{\boldsymbol{B}}'\\ \boldsymbol{I}\otimes\underline{s}_H\end{bmatrix}, \tag{246}$$

$$\bar{\boldsymbol{Q}}_{-+} = \begin{bmatrix}\mathbf{0} & \boldsymbol{D_H}\otimes\underline{\sigma}_H\end{bmatrix}, \quad \bar{\boldsymbol{Q}}_{--} = \boldsymbol{D_0} + \boldsymbol{D_L}, \tag{247}$$

and the distribution of $\bar{\mathcal{Z}}(t)$ at $t = 0$ (that defines the initial distribution of the busy period) is

$$\bar{\underline{\kappa}} = \{P(\bar{\mathcal{Z}}_+(0) = i)\} = \begin{bmatrix}\check{\underline{\beta}}' & 0\end{bmatrix}. \tag{248}$$

Notice that everything is the same as in Section 7.1.2, except the parameters of the initial workload distribution. Hence, it is not surprising that $\mathcal{W}_L = \bar{\mathcal{B}}/2$.

Corollary 11. *The distribution of $\mathcal{W}_L$ in time domain, in LST domain, and its moments can be expressed by*

$$F_{\mathcal{W}_L}(t) = F_{\bar{\mathcal{B}}}(2t), \quad f^*_{\mathcal{W}_L}(s) = f^*_{\bar{\mathcal{B}}}(s/2), \quad E(\mathcal{W}_L^k) = E(\bar{\mathcal{B}}^k)/2^k. \tag{249}$$

As $\mathcal{T}_L = \mathcal{W}_L + \mathcal{S}_L$ holds, it is straight forward to obtain the LST of the distribution of $\mathcal{T}_L$ and its moments.

Corollary 12. *The LST of the distribution of $\mathcal{T}_L$ is given by*

$$f^*_{\mathcal{T}_L}(s) = f^*_{\mathcal{W}_L}(s)f^*_{\mathcal{S}_L}(s). \tag{250}$$

Taking the derivatives of $f^*_{\mathcal{T}_L}(s)$ with respect to s and tending $s \to 0$ yields the moments of the sojourn time.

Corollary 13. *The moments of the sojourn time $\mathcal{T}_L$ are given by*

$$E(\mathcal{T}_L^k) = \sum_{i=0}^{k} \binom{k}{i} E(\mathcal{W}_L^i) E(\mathcal{S}_L^{k-i}). \tag{251}$$

The distribution function of the sojourn time is more involved to obtain. One could directly express it as a continuous time convolution of $F_{\mathcal{W}_L}(t)$ and $f_{\mathcal{S}_L}(t)$, but it would involve an integral which can be evaluated only numerically. Remind that both $F_{\mathcal{T}_L}(t)$ in the preemptive resume case and $F_{\mathcal{W}_L}(t)$ in the non-preemptive case are derived from the distribution of the busy period of an appropriate fluid model, which is computed in terms of Erlangization (see Section 4.2.3), meaning that an order-n approximation is applied where increasing n improves the accuracy. For the preemptive resume case we had that the order-n approximation is

$$F^{(n)}_{\mathcal{T}_L,preemp.}(t) = P(\tilde{\mathcal{B}}/2 < \text{Erlang}(n, \frac{n}{t})) = P(\tilde{\mathcal{B}} < \text{Erlang}(n, \frac{n}{2t})) = \underline{\tilde{\kappa}} \sum_{k=0}^{n-1} \tilde{\mathbf{\Psi}}_k^{\nu} \mathbb{1},$$

with $\nu = n/(2t)$ and $\underline{\tilde{\kappa}}\tilde{\mathbf{\Psi}}_k^{\nu}\mathbb{1}$ holding the probabilities that k Exp(ν) events occur during the busy period.

In the non-preemptive case, however, busy period $\bar{\mathcal{B}}$ corresponds to the waiting time only. Thus, we have that the sojourn time distribution is

$$F^{(n)}_{\mathcal{T}_L}(t) = P(\bar{\mathcal{B}}/2 + \mathcal{S}_L < \text{Erlang}(n, \frac{n}{t})).$$

Theorem 28. *The order-n approximation of the distribution function of $\mathcal{T}_L$ is*

$$F^{(n)}_{\mathcal{T}_L}(t) = \underline{\bar{\kappa}} \sum_{k=0}^{n-1} \bar{\mathbf{\Psi}}_k^{\nu} \mathbb{1} d_{n-k} + \underline{\check{p}} \mathbb{1} d_n, \tag{252}$$

where $\nu = n/(2t)$, matrices $\bar{\mathbf{\Psi}}_k^{\nu}$ are defined by Theorem 20 with using $\bar{\boldsymbol{Q}}$ instead of $\boldsymbol{Q}$, and probabilities d_n are given by

$$d_n = 1 - \underline{\sigma}_L \left(\boldsymbol{I} - \boldsymbol{S_L}/(2\nu)\right)^{-n} \mathbb{1}. \tag{253}$$

Proof. We have that

$$F^{(n)}_{\mathcal{T}_L}(t) = P(\bar{\mathcal{B}}/2 + \mathcal{S}_L < \text{Erlang}(n, \frac{n}{t})) = P(\bar{\mathcal{B}} + 2\mathcal{S}_L < \text{Erlang}(n, \nu))$$
$$= \underline{\bar{\kappa}} \sum_{k=0}^{n-1} \bar{\mathbf{\Psi}}_k^{\nu} \mathbb{1} \cdot \underbrace{P(2\mathcal{S}_L < \text{Erlang}(n-k, \nu))}_{d_{n-k}} + \underline{\check{p}} \mathbb{1} \cdot \underbrace{P(2\mathcal{S}_L < \text{Erlang}(n, \nu))}_{d_n},$$

where the second term corresponds to the case when $\mathcal{W}_L = 0$. The d_ℓ probabilities can be derived as

$$\begin{aligned} d_n &= P(2\mathcal{S}_L < \text{Erlang}(n, \nu)) = P(\mathcal{S}_L < \text{Erlang}(n, 2\nu)) \\ &= 1 - \int_{u=0}^{\infty} \frac{(2\nu u)^{n-1}}{(n-1)!} 2\nu e^{-2\nu u} \underline{\sigma}_L e^{\boldsymbol{S}_L u} \mathbb{1} du \\ &= 1 - \frac{(-\nu)^{n-1}}{(n-1)!} 2\nu \underline{\sigma}_L \int_{u=0}^{\infty} \frac{d^{n-1}}{d\nu^{n-1}} e^{-2\nu u} e^{\boldsymbol{S}_L u} \mathbb{1} du \\ &= 1 - \frac{(-\nu)^{n-1}}{(n-1)!} 2\nu \underline{\sigma}_L \frac{d^{n-1}}{d\nu^{n-1}} (2\nu \boldsymbol{I} - \boldsymbol{S_L})^{-1} \mathbb{1} = 1 - (2\nu)^n \underline{\sigma}_L (2\nu \boldsymbol{I} - \boldsymbol{S_L})^{-n} \mathbb{1}, \end{aligned}$$

that equals to (253). □

7.2.3 *The number of low priority customers*

As in the preemptive resume case, first the number of low priority customers at low priority departures is analyzed, from which the results corresponding to a random point in time are derived.

To obtain the number of low priority customers at low priority departures ($\mathcal{X}_L$) a tagged low priority customer is picked, and the number of low priority arrivals is counted during its stay in the system. This quantity consists of two components: the number of arrivals during the waiting time, and the number of additional arrivals during the service time.

The number of arrivals during the waiting time can be derived from the fluid model representing the remaining waiting time process introduced in Section 7.2.2. We follow the exactly same recipe as in Section 7.1.3 with the preemptive case, thus we modify the background process of the fluid model $\bar{\boldsymbol{Q}}$ such that it counts the number of arrivals during the busy period and get $\bar{\boldsymbol{Q}}'$. The blocks of the corresponding $\bar{\boldsymbol{\Psi}}'$ matrix, $\bar{\boldsymbol{\Psi}}_k$ are holding the probabilities that k arrivals occurred during the busy period (that is, during the waiting time) given the initial phase of the MMAP. These matrices can be calculated as Theorem 26 does in the preemptive resume case, the only difference is that matrix $\bar{\boldsymbol{Q}}$ needs to be used instead of matrix $\tilde{\boldsymbol{Q}}$.

As for the second component, let us introduce matrices $\boldsymbol{A_i}, i \geq 0$ whose (k,ℓ)th entry is the probability that the MMAP generates i low priority arrivals during a low priority service time starting from phase k and the MMAP phase at the end of service is ℓ. Matrices $\boldsymbol{A_i}$ are matrix-geometric

$$\boldsymbol{A_i} = \boldsymbol{\alpha} \cdot \boldsymbol{A}^i \boldsymbol{a}, \quad i \geq 0, \tag{254}$$

where

$$\boldsymbol{\alpha} = \boldsymbol{I} \otimes \underline{\sigma}_L, \tag{255}$$

$$\boldsymbol{A} = \left(-(\boldsymbol{D_0} + \boldsymbol{D_H}) \oplus \boldsymbol{S_L}\right)^{-1} (\boldsymbol{D_L} \otimes \boldsymbol{I}), \tag{256}$$

$$\boldsymbol{a} = \left(-(\boldsymbol{D_0} + \boldsymbol{D_H}) \oplus \boldsymbol{S_L}\right)^{-1} (\boldsymbol{I} \otimes \underline{s}_L). \tag{257}$$

Theorem 29. *The joint probability of the number of low priority customers in the system and the phase of the MMAP at low priority departure instants is*

$$\underline{x}_i^L = \underline{h}_i \cdot \boldsymbol{a} + \check{\underline{p}} \boldsymbol{A_i}, \tag{258}$$

where vector $\underline{h}_0 = \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_{\boldsymbol{0}}$ *and* $\underline{h}_i, i > 0$ *is defined recursively as*

$$\underline{h}_i = \underline{h}_{i-1} \cdot \boldsymbol{A} + \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_{\boldsymbol{i}} \boldsymbol{\alpha}. \tag{259}$$

Proof. Let us sum the number of arrivals during the waiting time and during the service time by convolution, yielding

$$\underline{x}_i^L = \sum_{k=0}^{i} \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_k \boldsymbol{A_{i-k}} + \check{\underline{p}} \boldsymbol{A_i} = \underbrace{\sum_{k=0}^{i} \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_k \boldsymbol{\alpha} \boldsymbol{A}^{i-k}}_{\underline{h}_i} \boldsymbol{a} + \check{\underline{p}} \boldsymbol{A_i}. \tag{260}$$

The recursion for h_i can be shown by

$$\underline{h}_i = \sum_{k=0}^{i} \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_k \boldsymbol{\alpha} \boldsymbol{A}^{i-k} = \underbrace{\sum_{k=0}^{i-1} \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_k \boldsymbol{\alpha} \boldsymbol{A}^{i-1-k}}_{\underline{h}_{i-1}} \cdot \boldsymbol{A} + \bar{\underline{\kappa}} \bar{\boldsymbol{\Psi}}_{\boldsymbol{i}} \boldsymbol{\alpha}. \tag{261}$$

□

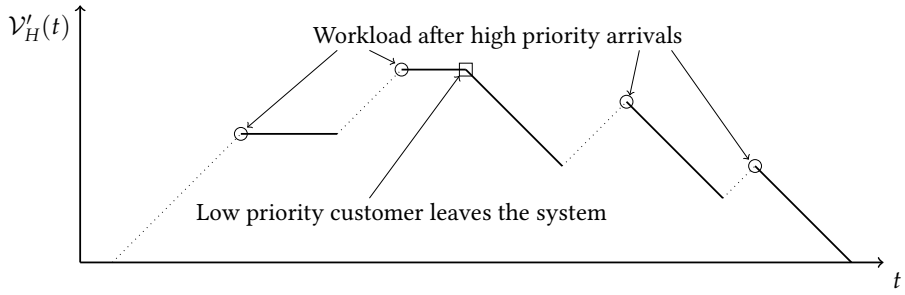


Figure 33.: The modified workload process of the high priority class

By introducing the GFs $\bar{\mathbf{\Psi}}(z) = \sum_{i=0}^{\infty} z^i \bar{\mathbf{\Psi}}_i$ and $\boldsymbol{A}(z) = \sum_{i=0}^{\infty} z^i \boldsymbol{A}_i$, the GF $\underline{X}_L(z) = \sum_{i=0}^{\infty} z^i \underline{x}_i^L$ is easy to obtain from (260) and (254).

Corollary 14. *$\underline{X}_L(z)$ is expressed by*

$$\underline{X}_L(z) = \underline{\bar{\kappa}} \bar{\mathbf{\Psi}}(z) \boldsymbol{A}(z) + \underline{\check{p}} \boldsymbol{A}(z), \tag{262}$$

where matrix $\boldsymbol{A}(z) = \sum_{i=0}^{\infty} z^i \boldsymbol{A}_i$ has the following closed form formula

$$\boldsymbol{A}(z) = \boldsymbol{\alpha}(\boldsymbol{I} - z\boldsymbol{A})^{-1}\boldsymbol{a}. \tag{263}$$

Based on (238) the factorial moments at departures are calculated by routine derivations of (262).

Corollary 15. *For the kth factorial moment of the number of low priority customers at low priority departures we have*

$$\underline{E(\mathcal{X}_L^k)} = \sum_{i=0}^{k} \binom{k}{i} \underline{\bar{\kappa}} \bar{\mathbf{\Psi}}^{(i)} \boldsymbol{A}^{(k-i)} + \underline{\check{p}} \boldsymbol{A}^{(k)}, \tag{264}$$

where matrices $\bar{\mathbf{\Psi}}^{(i)} = \frac{d^i}{dz^i} \bar{\mathbf{\Psi}}(z)|_{z=1}$ are obtained similar to (240) and matrices $\boldsymbol{A}^{(i)} = \frac{d^i}{dz^i} \boldsymbol{A}(z)|_{z=1}$ have the following closed form:

$$\boldsymbol{A}^{(i)} = i! \boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{A})^{-i-1} \boldsymbol{A}^i \boldsymbol{a}. \tag{265}$$

Having characterized the number of low priority customers at low priority departure epochs, the properties of the number of low priority customers at a random point in time are given by Theorem 27 and Corollaries 9 and 10.

7.2.4 *The analysis of the high priority class*

In the non-preemptive case the high priority class can not be analyzed in separation, since a high priority customer can not be served immediately when a low priority customer is in the server.

We use the workload process of the high priority class denoted by $\{\mathcal{V}_H(t), t > 0\}$ to derive the performance measures[1]. The trajectory of $\mathcal{V}_H(t)$ contains intervals where the slope is

1 Contrary to Sections 7.1.1 and 7.2.1, where the workload process of the entire system is discussed, the workload process considered here applies only to the high priority class.

zero corresponding to the periods when the server serves low priority customers. As before, $\mathcal{V}_H(t)$ is transformed to a fluid model (see Figure 33 for an example).

The blocks of the generator matrix of this fluid model are defined by

$$
\begin{aligned}
&Q^H_{++} = \begin{bmatrix} \boldsymbol{I}\otimes\boldsymbol{I}\otimes\boldsymbol{S_H} & \\ & \boldsymbol{I}\otimes\boldsymbol{S_H}\end{bmatrix}, \quad Q^H_{+-} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\otimes\underline{s}_H\end{bmatrix}, \quad Q^H_{+0} = \begin{bmatrix}\boldsymbol{I}\otimes\boldsymbol{I}\otimes\underline{s}_H\\ \boldsymbol{0}\end{bmatrix},\\
&Q^H_{-+} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{D_H}\otimes\underline{\sigma}_H\end{bmatrix}, \qquad Q^H_{--} = \boldsymbol{D_0}+\boldsymbol{D_L}, \quad Q^H_{-0} = \boldsymbol{0},\\
&Q^H_{0+} = \begin{bmatrix}\boldsymbol{D_H}\otimes\boldsymbol{I}\otimes\underline{\sigma}_H & \boldsymbol{0}\end{bmatrix}, \qquad Q^H_{0-} = \boldsymbol{I}\otimes\underline{s}_L, \qquad Q^H_{00} = (\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L}.
\end{aligned}
$$

Four state groups can be identified in the generator. The two state groups of $\mathcal{N}_+$ both correspond to the workload accumulation due to a new high priority arrival. The difference is that in the first state group the server works on a low priority customer, thus the phase of its service needs to be maintained during the workload accumulation. In the negative states $\mathcal{N}_-$ the server is working on a high, in the zero states $\mathcal{N}_0$ the server is working on a low priority customer.

The probability of the phases when the workload process leaves level zero, denoted by vector $\underline{\kappa}^H$, is not easy to obtain. Regarding this vector we are relying on the results of [79], which we re-formulate and simplify at several points due to the PH distributed service times.

Let us investigate the system at the departures that leave the high priority queue empty, and introduce two probability vectors, $\underline{\phi}$ and $\underline{\phi}_0$ associated to this embedded process. The ith entry of $\underline{\phi}_0$ is the probability that the whole system is empty at the embedded instant and the phase of the MMAP is i. Entry i of vector $\underline{\phi}$ is the probability that the embedded process is in state i in the product space of the MMAP phase and the phase of the low priority service time.

Theorem 30. *Vector $\underline{\phi}_0$ is given by*

$$
\underline{\phi}_0 = \frac{(1-\rho)\underline{p}_-(-\boldsymbol{D_0})}{\lambda_L\underline{p}_-\mathbb{1}+(1-\rho)\underline{p}_-\boldsymbol{D_H}\mathbb{1}}, \tag{266}
$$

where $\underline{p}_-$ is the probability mass vector of the fluid queue representing the workload process of the whole system (see Sections 7.1.1 and 7.2.1).

Vector $\underline{\phi}$ is the unique solution to the linear system

$$
\begin{aligned}
\underline{\phi} = {}& (\underline{\phi}-\underline{\phi}_0)(\boldsymbol{I}\otimes\underline{\sigma}_L)(-(\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L})^{-1}\begin{bmatrix}\boldsymbol{D_H}\otimes\boldsymbol{I}\otimes\underline{\sigma}_H & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Psi}^H\\
&+(\underline{\phi}-\underline{\phi}_0)(\boldsymbol{I}\otimes\underline{\sigma}_L)(-(\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L})^{-1}(\boldsymbol{I}\otimes\underline{s}_L)\\
&+\underline{\phi}_0(-\boldsymbol{D_0})^{-1}(\boldsymbol{D_L}\otimes\underline{\sigma}_L)(-(\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L})^{-1}\begin{bmatrix}\boldsymbol{D_H}\otimes\boldsymbol{I}\otimes\underline{\sigma}_H & \boldsymbol{0}\end{bmatrix}\boldsymbol{\Psi}^H\\
&+\underline{\phi}_0(-\boldsymbol{D_0})^{-1}(\boldsymbol{D_L}\otimes\underline{\sigma}_L)(-(\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L})^{-1}(\boldsymbol{I}\otimes\underline{s}_L)\\
&+\underline{\phi}_0(-\boldsymbol{D_0})^{-1}\begin{bmatrix}\boldsymbol{0} & \boldsymbol{D_H}\otimes\underline{\sigma}_H\end{bmatrix}\boldsymbol{\Psi}^H,
\end{aligned} \tag{267}
$$

$$
\underline{\phi}\mathbb{1} = 1, \tag{268}
$$

where $\boldsymbol{\Psi}^H$ is the solution of the NARE

$$
\begin{aligned}
\boldsymbol{\Psi}^H Q^H_{-+}\boldsymbol{\Psi}^H + \boldsymbol{\Psi}^H Q^H_{--} + (Q^H_{++} + Q^H_{+0}(-Q^H_{00})^{-1}Q^H_{0+})\boldsymbol{\Psi}^H\\
+ Q^H_{+-} + Q^H_{+0}(-Q^H_{00})^{-1}Q^H_{0-} = \boldsymbol{0}.
\end{aligned} \tag{269}
$$

Proof. Eq. (266) follows from [79], Theorem 3.1 and [79], Lemma 3.2.

Eq. (267) has 5 terms. The first one corresponds to the case when there are low priority customers in the system when the last high priority customer leaves. The server starts to serve a low priority customer. The PH of the service process and the MMAP evolve together, and the MMAP generates a high priority arrival before the current service is completed, and initiates the workload process (see Figure 33). The next departure leaving the high priority class empty occurs when the workload of the high priority class returns to level zero, with the corresponding phase transitions given by $\mathbf{\Psi}^H$ (which satisfies the usual NARE after censoring out the zero states). According to the second term the low priority service is completed before the MMAP generates a high priority customer, providing the phase of the next embedded point. In the third and fourth term the last high priority customer leaves the system empty, and the next arriving customer is a low priority one, while in the last term the next arriving customer is a high priority one. □

Let us introduce vectors q_L^H and q_0^H as the stationary phase probabilities that the server is working on a low priority customer and that the system is idle when there are no high priority customers in the system, respectively. These probability vectors can be obtained from ϕ and ϕ_0 by taking into account the mean amount of time spent in various phases in the system, yielding

$$\begin{aligned} q_L^H &= \frac{1}{c^H}(\phi - \phi_0 + \phi_0(-\boldsymbol{D_0})^{-1}\boldsymbol{D_L})(\boldsymbol{I}\otimes\underline{\sigma}_L)(-(\boldsymbol{D_0}+\boldsymbol{D_L})\oplus\boldsymbol{S_L})^{-1}, \\ q_0^H &= \frac{1}{c^H}\phi_0(-\boldsymbol{D_0})^{-1}, \end{aligned} \tag{270}$$

where c^H is a normalization constant. From these vectors the initial phase distribution vector for the high priority workload process denoted by $\underline{\kappa}^H$ is given by

$$\underline{\kappa}^H = q_L^H \begin{bmatrix} \boldsymbol{D_H}\otimes\boldsymbol{I}\otimes\underline{\sigma}_H & \mathbf{0} \end{bmatrix} + q_0^H \begin{bmatrix} \mathbf{0} & \boldsymbol{D_H}\otimes\underline{\sigma}_H \end{bmatrix} = q_L^H \boldsymbol{Q}_{0+}^H + q_0^H \boldsymbol{Q}_{-+}^H. \tag{271}$$

Finally, the next two theorems provide the performance measures for the high priority customers.

Theorem 31. *The pdf of the sojourn time of high priority customers $f_{\mathcal{T}_H}(t)$ is matrix-exponential*

$$f_{\mathcal{T}_H}(t) = \underline{\zeta} e^{\boldsymbol{Z}t}\underline{v}, \tag{272}$$

with parameters

$$\underline{\zeta} = \begin{bmatrix} \underline{\kappa}^H & 0 \end{bmatrix}/c, \quad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{K}^H & \begin{bmatrix} \mathbb{1}\otimes\boldsymbol{I}\otimes\underline{s}_H \\ \mathbf{0} \end{bmatrix} \\ \mathbf{0} & \boldsymbol{S_L} \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} 0 \\ \mathbb{1}\otimes\underline{s}_H \\ \underline{s}_L \end{bmatrix}, \tag{273}$$

where $\boldsymbol{K}^H = \boldsymbol{Q}_{++}^H + \boldsymbol{Q}_{+0}^H(-\boldsymbol{Q}_{00}^H)^{-1}\boldsymbol{Q}_{0+}^H + \mathbf{\Psi}^H\boldsymbol{Q}_{-+}^H$ and c is the normalization constant.

Proof. The density of the workload at high priority arrival including the service time requirement the customer brought to the system is $\underline{\kappa}^H e^{\boldsymbol{K}^H x}\boldsymbol{Q}_{+0}^H$ if the server works on a low priority customer and it is $\underline{\kappa}^H e^{\boldsymbol{K}^H x}\boldsymbol{Q}_{+-}^H$ otherwise (see the points marked by circles in Figure 33). In the latter case the sojourn time of the entering customer is x. In the former case, however,

the remaining service time of the low priority customer has to be taken into account as well. The phase of the low priority service is also encoded in the background process, hence we have

$$f_{\mathcal{T}_H}(t) = \left(\underline{\kappa}^H \int_{x=0}^{\infty} e^{\boldsymbol{K}^H x} \boldsymbol{Q}^H_{+0} (\mathbb{1} \otimes \boldsymbol{I}) e^{\boldsymbol{S}_L (t-x)} \underline{s}_L \, dt + \underline{\kappa}^H e^{\boldsymbol{K}^H x} \boldsymbol{Q}^H_{+-} \mathbb{1} \right) / c. \tag{274}$$

The convolution of the two matrix exponentials with parameters $\boldsymbol{K}^H$ and $\boldsymbol{S}_L$ can be represented by a single matrix exponential with parameter $\boldsymbol{Z}$ according to Theorem 37. The second term can be expressed using $\underline{\zeta} e^{\boldsymbol{Z}t}$ as well, by adding transitions from the first matrix block to the absorbing state with rates $\boldsymbol{Q}^H_{+-}\mathbb{1} = \begin{bmatrix} 0 \\ \mathbb{1} \otimes \underline{s}_H \end{bmatrix}$. Putting together the two terms provides the theorem. □

Corollary 16. *The LST of the distribution function and the moments of $\mathcal{T}_H$ are given by*

$$f^*_{\mathcal{T}_H}(s) = \underline{\zeta}(s\boldsymbol{I} - \boldsymbol{Z})^{-1}\underline{v}, \qquad E(\mathcal{T}_H^k) = k!\underline{\zeta}(-\boldsymbol{Z})^{-k-1}\underline{v}. \tag{275}$$

For the analysis of the number of high priority customers in the system we introduce a QBD, where the matrices corresponding to level backward, local and level forward transitions (denoted by $\boldsymbol{B}$, $\boldsymbol{L}$ and $\boldsymbol{F}$, respectively) are

$$\boldsymbol{L} = \begin{bmatrix} (\boldsymbol{D_0} + \boldsymbol{D_L}) \oplus \boldsymbol{S_L} & \boldsymbol{I} \otimes \underline{s}_L \underline{\sigma}_H \\ & (\boldsymbol{D_0} + \boldsymbol{D_L}) \oplus \boldsymbol{S_H} \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} & \\ & \boldsymbol{I} \otimes \underline{s}_H \underline{\sigma}_H \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{D_H} \otimes \boldsymbol{I} & \\ & \boldsymbol{D_H} \otimes \boldsymbol{I} \end{bmatrix}.$$

In the first group of states the server is working on a low, in the second one it is working on a high priority customer. It is possible to move from the first state group to second one (see matrix $\boldsymbol{L}$), but not the way around at levels > 0.

The entries of vector $\underline{y}^H_i$ are the probabilities that there are i high priority customers in the system and the background process is in different phases. It is well known that QBDs have a matrix geometric distribution.

Theorem 32. *Vectors $\underline{y}^H_i$ have the following matrix geometric form:*

$$\underline{y}^H_i = \underline{y}^H_0 \boldsymbol{R}^i, \tag{276}$$

where matrix $\boldsymbol{R}$ is the minimal non-negative solution to the matrix-quadratic equation

$$\boldsymbol{F} + \boldsymbol{R}\boldsymbol{L} + \boldsymbol{R}^2\boldsymbol{B} = \boldsymbol{0}, \tag{277}$$

and the probability of level 0 is $\underline{y}^H_0 = \begin{bmatrix} q^H_L & q^H_0 \end{bmatrix} / c'$. The normalization constant is $c' = \begin{bmatrix} q^H_L & q^H_0 \end{bmatrix} (\boldsymbol{I} - \boldsymbol{R})^{-1}\mathbb{1}$.

Proof. By definition in (270), vectors q^H_L and q^H_0 are the stationary phase probability vectors given that there are no high priority customers in the system. The matrix-geometric stationary distribution is a standard property of QBDs (see Section 4.1.3). □

Corollary 17. *The GF of the number of high priority customers $Y_H(z) = \sum_{i=0}^{\infty} z^i \underline{y}^H_i \mathbb{1}$ and the factorial moments $E(\mathcal{Y}_H^k)$ are given by*

$$Y_H(z) = \underline{y}^H_0 (\boldsymbol{I} - z\boldsymbol{R})^{-1}\mathbb{1}, \qquad E(\mathcal{Y}_H^k) = k!\underline{y}^H_0 \boldsymbol{R}^k (\boldsymbol{I} - \boldsymbol{R})^{-k-1}\mathbb{1}. \tag{278}$$

7.3 NUMERICAL BEHAVIOR

The steps of the presented analysis procedure are significantly less computationally demanding than the past methods published in the literature considering the same queueing system.

In this section we compare our procedure with three prior methods: the method of [3] (transformed to continuous time), its improved version published in [45], and the procedure of [48]. Note that the latter two procedures are far less general than [3] or the proposed one. They can handle only preemptive resume service, they do not analyze the sojourn time at all, and [48] is only able to provide the moments of the number of customers.

Since all involved procedures are exact, only the scalability is investigated, that is, the analysis time as the function of the number of phases.

For this purpose let us define the MMAP matrices as

$$\boldsymbol{D_0}^{(K)} = \begin{bmatrix} \bullet & K\nu & & & \\ \gamma & \bullet & (K-1)\nu & & \\ & \ddots & \ddots & \ddots & \\ & & (K-1)\gamma & \bullet & \nu \\ & & & K\gamma & \bullet \end{bmatrix}, \quad \boldsymbol{D_L}^{(K)} = \begin{bmatrix} 0 & & & \\ & r_L/K & & \\ & & 2r_L/K & \\ & & & \ddots \\ & & & & r_L \end{bmatrix},$$

and matrix $\boldsymbol{D_H}^{(K)}$ is defined similarly. The diagonal entries denoted by $\bullet$ are determined uniquely such that the row sums of $\boldsymbol{D_0}^{(K)} + \boldsymbol{D_L}^{(K)} + \boldsymbol{D_H}^{(K)}$ are zeroes.

The service times are characterized by order-2 PH distributions with parameters

$$\underline{\sigma}_H = \begin{bmatrix} 0.16667 & 0.83333 \end{bmatrix}, \quad \underline{\sigma}_L = \begin{bmatrix} 0.58824 & 0.41176 \end{bmatrix},$$

$$\boldsymbol{S_H} = \begin{bmatrix} -0.66667 & 0.66667 \\ 0 & -4 \end{bmatrix}, \quad \boldsymbol{S_L} = \begin{bmatrix} -3.2941 & 3.2941 \\ 0 & -5.6 \end{bmatrix},$$

having service rates $\mu_L = 2.8$ and $\mu_H = 2$. The utilization depends on K, it varies between 0.6 and 0.75.

Figure 34 depicts the analysis time required to obtain the first 10 moments of the number of low priority customers in the system in the preemptive case as the function of K^2. (This is the only performance measure that is supported by all the procedures). It is clearly visible that the presented method is at least an order of magnitude faster than the prior ones, and is able to solve systems with a large number of phases. No numerical problems were encountered even with the largest model. Additionally, as opposed to [45] and [48], the presented procedure can provide sojourn time related performance measures, and is able to handle the case of non-preemptive service as well.

7.4 DEPARTURE PROCESS ANALYSIS OF PRIORITY QUEUES

This section describes a method to obtain the multi-class lag-1 joint moments of the inter-departure times of priority queues. From these moments it is possible to create a MMAP (based on the results of Section 3.2.3) in order to approximate the departure process in a Markovian way.

The approach to derive the multi-class joint moments of the inter-departure times is similar to the one presented in 5.3.3 for the single-class case. For simplicity, we discuss only the

2 In our MATLAB implementation the NARE problems are solved by the ADDA procedure [86] and the Sylvester equations are solved by the lyap function of MATLAB, which is based on the Hessenberg-Schur algorithm [33].

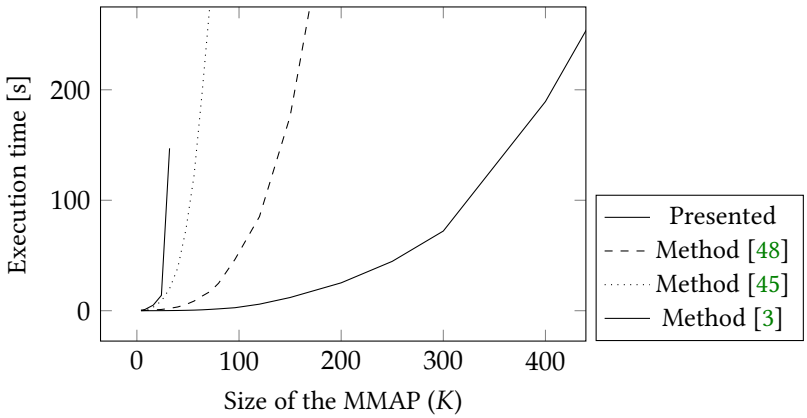


Figure 34.: Comparison of the execution times of various procedures

two-class case with preemptive resume priority, but the procedure itself can be extended to handle more general systems as well.

The main idea is that the stochastic behavior of two consecutive departure intervals is independent on the number of customers in the queue when there are at least two customers. The reason is that the system cannot become idle during the two consecutive departure intervals in this case. As a consequence, we have to distinguish just six cases, as follows:

- $0,0$: the last departure left the system empty,
- $1,0$: at the last departure one high and zero low priority customers are left in the system,
- $1,1+$: at the last departure one high and at least one low priority customers are left in the system,
- $2+,0+$: at the last departure at least two high priority customers are left in the system,
- $0,1$: at the last departure zero high and one low priority customers are left in the system,
- $0,2+$: at the last departure zero high and at least two low priority customers are left in the system.

(Note that there were only three cases to distinguish in the single-class MAP/MAP/1 system.)

The analysis method presented in Sections 7.1 and 7.2 provides only per-class performance measures and does not allow the analysis of the joint behavior of the priority classes. Hence, a different approach is needed to obtain the probabilities of the above listed these six cases.

The approach we are going to use is based on [3], where the analysis of the discrete time DMAP/PH/1 priority queue is presented. In contrast to [3], here we consider the continuous time model and extend the results of that paper in several ways. We provide new closed formulas and more efficient algorithms than the existing ones.

7.4.1 *The MMAP[K]/PH[K]/1 preemptive priority queue as a QBD process*

It is possible to define a three dimensional CTMC to model the queue length behavior. One dimension keeps track of the length of the high priority queue, the second one the length of the low priority queue, and the third dimension describes the phase of the arrival MMAP together with the phases of the low and high priority service PH distributions.

With proper numbering of the states the structure of the generator of this Markov chain is

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F} & & & \\ \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & & \\ & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{279}$$

where the blocks of the generator are infinite matrices corresponding to the same number of high priority customers but different number of low priority customers and different phases of the arrival and service processes. The blocks of $\boldsymbol{Q}$ are defined as

$$\boldsymbol{F} = \mathrm{diag}\langle \boldsymbol{E}_H \rangle \tag{280}$$

$$\boldsymbol{B} = \mathrm{diag}\langle \boldsymbol{J}_1^{(H)} \rangle \tag{281}$$

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} & \boldsymbol{E}_L & & & \\ & \boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} & \boldsymbol{E}_L & & \\ & & \boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} & \boldsymbol{E}_L & \\ & & & \ddots & \ddots \end{bmatrix}, \tag{282}$$

$$\boldsymbol{L}_0 = \begin{bmatrix} \boldsymbol{E}_0 & \boldsymbol{E}_L & & & \\ \boldsymbol{J}_1^{(L)} & \boldsymbol{E}_0 + \boldsymbol{J}_0^{(L)} & \boldsymbol{E}_L & & \\ & \boldsymbol{J}_1^{(L)} & \boldsymbol{E}_0 + \boldsymbol{J}_0^{(L)} & \boldsymbol{E}_L & \\ & & & \ddots & \ddots \end{bmatrix}, \tag{283}$$

with the notation

$$\begin{aligned} \boldsymbol{E}_i &= \boldsymbol{D}_i \otimes \boldsymbol{I} \otimes \boldsymbol{I}\,, \quad i = \{0, L, H\}, \\ \boldsymbol{J}_0^{(H)} &= \boldsymbol{I} \otimes \boldsymbol{S}_H \otimes \boldsymbol{I} \quad , \boldsymbol{J}_1^{(H)} = \boldsymbol{I} \otimes \underline{s}_H \underline{\sigma}_H \otimes \boldsymbol{I}\,, \\ \boldsymbol{J}_0^{(L)} &= \boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{S}_L \quad , \boldsymbol{J}_1^{(L)} = \boldsymbol{I} \otimes \boldsymbol{I} \otimes \underline{s}_L \underline{\sigma}_L\,. \end{aligned} \tag{284}$$

With these definitions matrices $\boldsymbol{E}_L$ (and $\boldsymbol{E}_H$) contain the transition rates accompanied by low (and high) priority arrivals, while $\boldsymbol{J}_1^{(L)}$ (and $\boldsymbol{J}_1^{(H)}$) are the ones accompanied by low (and high) priority services, respectively.

Since the generator is a QBD (with infinite number of phases), the solution is matrix-geometric (see Section 4.1.3), thus we have

$$\underline{y}_k = \underline{y}_0 \boldsymbol{R}^k\,, \quad k \geq 0, \tag{285}$$

where $\underline{y}_k$ is the vector of the steady state probability of the states with k high priority customers. This vector can be partitioned according the number of low priority customers, $\underline{y}_k = \{\underline{y}_{k,j},\, j \geq 0\}$, where $\underline{y}_{k,j}$ denotes the vector of steady state probabilities for the states with k high and j low priority customers. Furthermore, we denote the marginal steady state probability vectors of the classes as

$$y_i^{(H)} = \sum_{j=0}^{\infty} \underline{y}_{i,j}, \quad y_i^{(L)} = \sum_{j=0}^{\infty} \underline{y}_{j,i}\,.$$

Due to the definition of the blocks of the generator, both matrices $\boldsymbol{R}$ and $\boldsymbol{G}$ exhibit an upper-block-Toeplitz structure, since the number of low priority arrivals and the number of customers in the system are independent given the phase of the arrival and service processes (shown in [3]). Thus we have:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0^{(L)} & \boldsymbol{G}_1^{(L)} & \boldsymbol{G}_2^{(L)} & \cdots \\ & \boldsymbol{G}_0^{(L)} & \boldsymbol{G}_1^{(L)} & \cdots \\ & & \boldsymbol{G}_0^{(L)} & \cdots \\ & & & \ddots \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_0^{(L)} & \boldsymbol{R}_1^{(L)} & \boldsymbol{R}_2^{(L)} & \cdots \\ & \boldsymbol{R}_0^{(L)} & \boldsymbol{R}_1^{(L)} & \cdots \\ & & \boldsymbol{R}_0^{(L)} & \cdots \\ & & & \ddots \end{bmatrix}. \tag{286}$$

As in every homogeneous QBD, these matrices are the minimal non-negative solutions to the matrix quadratic equations (see (123) and (128))

$$\boldsymbol{0} = \boldsymbol{B} + \boldsymbol{L}\boldsymbol{G} + \boldsymbol{F}\boldsymbol{G}^2. \tag{287}$$

$$\boldsymbol{0} = \boldsymbol{F} + \boldsymbol{R}\boldsymbol{L} + \boldsymbol{R}^2\boldsymbol{B}, \tag{288}$$

Applying the definitions of matrices $\boldsymbol{B}$, $\boldsymbol{L}$ and $\boldsymbol{F}$, and exploiting the upper-block-Toeplitz structure of matrices $\boldsymbol{R}$ and $\boldsymbol{G}$ we can derive relationships for matrices $\boldsymbol{R}_i^{(L)}$ and $\boldsymbol{G}_i^{(L)}, i \geq 0$ (see [3]).

The equations for matrices $\boldsymbol{G}_i^{(L)}$ are as follows:

$$\text{for } i = 0: \quad \boldsymbol{0} = \boldsymbol{J}_1^{(H)} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)})\boldsymbol{G}_0^{(L)} + \boldsymbol{E}_H {\boldsymbol{G}_0^{(L)}}^2, \tag{289}$$

$$\text{for } i > 0: \quad \boldsymbol{0} = \boldsymbol{E}_L \boldsymbol{G}_{i-1}^{(L)} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)})\boldsymbol{G}_i^{(L)} + \boldsymbol{E}_H \sum_{k=0}^{i} \boldsymbol{G}_k^{(L)} \boldsymbol{G}_{i-k}^{(L)}. \tag{290}$$

Matrices $\boldsymbol{G}_i^{(L)}$ have important probabilistic interpretations. Entry (a, b) of matrix $\boldsymbol{G}_i^{(L)}$ is the conditional probability that starting from level n with the background process being in phase a, (1) the first visit to level $n-1$ occurs in phase b, (2) i low probability customers arrive during the first passage time.

The following expressions can be obtained for matrices $\boldsymbol{R}_i^{(L)}$:

$$\text{for } i = 0: \quad \boldsymbol{0} = \boldsymbol{E}_H + \boldsymbol{R}_0^{(L)}(\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)}) + {\boldsymbol{R}_0^{(L)}}^2 \boldsymbol{J}_1^{(H)}, \tag{291}$$

$$\text{for } i > 0: \quad \boldsymbol{0} = \boldsymbol{R}_{i-1}^{(L)} \boldsymbol{E}_L + \boldsymbol{R}_i^{(L)}(\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)}) + \sum_{k=0}^{i} \boldsymbol{R}_k^{(L)} \boldsymbol{R}_{i-k}^{(L)} \boldsymbol{J}_1^{(H)}. \tag{292}$$

Interestingly, summing up equations (290) from $i = 1$ to ∞ and adding (289) leads to a matrix quadratic equation for $\hat{\boldsymbol{G}}^{(L)} = \sum_{i=0}^{\infty} \boldsymbol{G}_i^{(L)}$, since

$$\boldsymbol{0} = \boldsymbol{J}_1^{(H)} + (\boldsymbol{E}_0 + \boldsymbol{E}_L + \boldsymbol{J}_0^{(H)})\hat{\boldsymbol{G}}^{(L)} + \boldsymbol{E}_H {\hat{\boldsymbol{G}}^{(L)}}^2. \tag{293}$$

Similarly, the sum of matrices $\boldsymbol{R}_i^{(L)}$, denoted by $\hat{\boldsymbol{R}}^{(L)} = \sum_{i=0}^{\infty} \boldsymbol{R}_i^{(L)}$, can be obtained as the minimal non-negative solution of a matrix quadratic equation as well, as

$$\boldsymbol{0} = \boldsymbol{E}_H + \hat{\boldsymbol{R}}^{(L)}(\boldsymbol{E}_0 + \boldsymbol{E}_L + \boldsymbol{J}_0^{(H)}) + {\hat{\boldsymbol{R}}^{(L)}}^2 \boldsymbol{J}_1^{(H)}. \tag{294}$$

The generating function of matrix series $\boldsymbol{G}_i^{(L)}$ defined by $\boldsymbol{G}^{(L)}(z) = \sum_{k=0}^{\infty} z^k \boldsymbol{G}_k^{(L)}$ will be used several times in the sequel. From (290) and (289) we have that $\boldsymbol{G}^{(L)}(z)$ is the solution of the following matrix-quadratic equation:

$$\boldsymbol{0} = \boldsymbol{J}_1^{(H)} + (\boldsymbol{E}_0 + z\boldsymbol{E}_L + \boldsymbol{J}_0^{(H)})\boldsymbol{G}^{(L)}(z) + \boldsymbol{E}_H {\boldsymbol{G}^{(L)}(z)}^2. \tag{295}$$

Finally, based on (285) and the block structure of matrix $\boldsymbol{R}$ the steady state probability vector of the number of high and low priority customers in the system can be expressed as

$$\underline{y}_{i,j} = \sum_{k=0}^{j} \underline{y}_{i-1,k} \boldsymbol{R}_{j-k}^{(L)}, \quad i \geq 1, j \geq 0, \tag{296}$$

To completely characterize the joint distribution of the number of high and low priority customers, it remains to derive the stationary probability vector at level 0, $\underline{y}_0 = \{\underline{y}_{0,j},\ j \geq 0\}$.

7.4.2 *Analysis of level zero*

Relations for vector $\underline{y}_0$ can be derived from the boundary equations $\underline{y}\boldsymbol{Q} = \underline{0}$ as

$$\underline{y}_0 \boldsymbol{L_0} + \underline{y}_0 \boldsymbol{R} \boldsymbol{B} = \underline{0}, \quad \underline{y}_{0,0} \mathbb{1} = 1 - \lambda^{(H)}/\mu^{(H)} - \lambda^{(L)}/\mu^{(L)}. \tag{297}$$

Due to the structure of matrices $\boldsymbol{L_0}$, $\boldsymbol{B}$ and $\boldsymbol{R}$, (297) is equivalent to the solution of an M/G/1 type CTMC (see [90]). However, by using the relation $\boldsymbol{RB} = \boldsymbol{FG}$ (see e.g. [57], page 144) it is possible to re-formulate (297) and the corresponding M/G/1 type CTMC to a more appropriate form. The equations for $\underline{y}_0$ by using $\boldsymbol{G}$ instead of $\boldsymbol{R}$ are then

$$\underline{y}_0 \boldsymbol{L_0} + \underline{y}_0 \boldsymbol{F} \boldsymbol{G} = \underline{0}, \quad \underline{y}_{0,0} \mathbb{1} = 1 - \lambda^{(H)}/\mu^{(H)} - \lambda^{(L)}/\mu^{(L)}, \tag{298}$$

and the generator of the related M/G/1 type CTMC becomes

$$\boldsymbol{Q_0} = \boldsymbol{L_0} + \boldsymbol{FG} = \begin{bmatrix} \boldsymbol{E_0} + \boldsymbol{E_H} \boldsymbol{G_0^{(L)}} & \boldsymbol{E_L} + \boldsymbol{E_H} \boldsymbol{G_1^{(L)}} & \boldsymbol{E_H} \boldsymbol{G_2^{(L)}} & \boldsymbol{E_H} \boldsymbol{G_3^{(L)}} & \dots \\ \boldsymbol{J_1^{(L)}} & \boldsymbol{E_0} + \boldsymbol{J_0^{(L)}} + \boldsymbol{E_H} \boldsymbol{G_0^{(L)}} & \boldsymbol{E_L} + \boldsymbol{E_H} \boldsymbol{G_1^{(L)}} & \boldsymbol{E_H} \boldsymbol{G_2^{(L)}} & \dots \\ & \boldsymbol{J_1^{(L)}} & \boldsymbol{E_0} + \boldsymbol{J_0^{(L)}} + \boldsymbol{E_H} \boldsymbol{G_0^{(L)}} & \boldsymbol{E_L} + \boldsymbol{E_H} \boldsymbol{G_1^{(L)}} & \dots \\ & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{299}$$

The solution of M/G/1-type Markov chains is based on their invariant matrix $\boldsymbol{G_{H_0}}$ whose entry in position (i, j) is the probability that starting in phase i at level n the first phase visited at level $n-1$ is state j [57]. (Hence, this matrix plays the same role as the matrix $\boldsymbol{G}$ of QBDs). In this particular M/G/1-type system matrix $\boldsymbol{G_{H_0}}$ is the minimal non-negative solution to the matrix polynomial equation

$$\boldsymbol{0} = \boldsymbol{J_1^{(L)}} + (\boldsymbol{E_0} + \boldsymbol{J_0^{(L)}}) \boldsymbol{G_{H_0}} + \boldsymbol{E_L} \boldsymbol{G_{H_0}}^2 + \boldsymbol{E_H} \sum_{i=0}^{\infty} \boldsymbol{G_i^{(L)}} \boldsymbol{G_{H_0}}^i \cdot \boldsymbol{G_{H_0}}. \tag{300}$$

The stationary probability vectors $\underline{y}_{0,i}$ can be calculated recursively using the Ramaswami formula [70]. Tailoring it to this particular system gives

$$\underline{y}_{0,i} = \left(\sum_{k=0}^{i-1} \underline{y}_{0,k} \boldsymbol{T_{i-k}} \right) (-\boldsymbol{T_0})^{-1}, \quad i \geq 1, \tag{301}$$

where matrices $\boldsymbol{T_i}$ are defined by

$$\begin{aligned} \boldsymbol{T_i} &= \sum_{k=i}^{\infty} \boldsymbol{E_H} \boldsymbol{G_k^{(L)}} \boldsymbol{G_{H_0}}^{k-i}, \quad i \geq 2, \\ \boldsymbol{T_1} &= \boldsymbol{E_L} + \sum_{k=1}^{\infty} \boldsymbol{E_H} \boldsymbol{G_k^{(L)}} \boldsymbol{G_{H_0}}^{k-1}, \\ \boldsymbol{T_0} &= \boldsymbol{E_0} + \boldsymbol{J_0^{(L)}} + \boldsymbol{E_H} \boldsymbol{G_0^{(L)}} + \boldsymbol{T_1} \boldsymbol{G_{H_0}}, \end{aligned} \tag{302}$$

and, vector $\underline{y}_{0,0}$ is the solution of the linear system

$$\begin{aligned} &\underline{y}_{0,0}\left(\boldsymbol{E_0}+\boldsymbol{E_H}\boldsymbol{G_0^{(L)}}+\boldsymbol{T_1}(-\boldsymbol{T_0})^{-1}\boldsymbol{J_1^{(L)}}\right)=\underline{0},\\ &\underline{y}_{0,0}\mathbb{1}=1-\lambda^{(H)}/\mu^{(H)}-\lambda^{(L)}/\mu^{(L)}, \end{aligned} \tag{303}$$

which, utilizing that $\boldsymbol{T_0}\boldsymbol{G_{H_0}}=-\boldsymbol{J_1^{(L)}}$ can be simplified to

$$\begin{aligned} &\underline{y}_{0,0}\left(\boldsymbol{E_0}+\boldsymbol{E_L}\boldsymbol{G_{H_0}}+\boldsymbol{E_H}\sum_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}\boldsymbol{G_{H_0}}^{i}\right)=\underline{0},\\ &\underline{y}_{0,0}\mathbb{1}=1-\lambda^{(H)}/\mu^{(H)}-\lambda^{(L)}/\mu^{(L)}. \end{aligned} \tag{304}$$

(Observe that the matrix in the parenthesis is the generator of the background process restricted to level $(0,0)$).

7.4.3 *The joint moments of the departure process*

As shown in Section 5.3.3, the departure process of a MAP/MAP/1 queue is an infinite state MAP. Similarly, the departure process of the MMAP[K]/PH[K]/1 priority queue is an infinite MMAP as well. This MMAP generates a high priority (low priority) arrival when a high priority (low priority) departure occurs in the system. The computation of the joint moments of the departure process, $\eta_{i,j}^{(L)}, \eta_{i,j}^{(H)}$, is rather difficult based on this infinite MMAP representation. Instead of using this representation directly, we construct a finite MMAP (considering the 6 cases listed above in the beginning of Section 7.4), such that the joint distribution of the first two arrivals of this MMAP – starting from the appropriate initial distribution – is identical with the one of two consecutive stationary departures of the MMAP[K]/PH[K]/1 priority queue.

The blocks of the matrices of this finite MMAP are constructed from the block matrices of the QBD describing the queue length process ((280)-(283)), such that the transitions between the 6 listed cases are taken into consideration. The initial probability distribution is computed according to the stationary distribution of the queue length process just after a departure considering the 6 listed cases.

The matrices of the resulting MMAP representation are as follows:

$$\boldsymbol{H_0}=\begin{bmatrix} \boldsymbol{M_1} & \boldsymbol{E_H} & & & \boldsymbol{E_L} & \\ & \boldsymbol{M_2} & \boldsymbol{E_L} & \boldsymbol{E_H} & & \\ & & \boldsymbol{M_3} & \boldsymbol{E_H} & & \\ & & & \boldsymbol{M_4} & & \\ & & \boldsymbol{E_H} & & \boldsymbol{M_5} & \boldsymbol{E_L} \\ & & \boldsymbol{E_H} & & & \boldsymbol{M_6} \end{bmatrix}, \tag{305}$$

$$\boldsymbol{H_H}=\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{J_1^{(H)}} & & & & & \boldsymbol{0} \\ & & & & \boldsymbol{J_1^{(H)}} & \boldsymbol{0} \\ & & & \boldsymbol{J_1^{(H)}} & & \boldsymbol{0} \\ & & & & & \boldsymbol{0} \\ & & & & & \boldsymbol{0} \end{bmatrix}, \boldsymbol{H_L}=\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ & & & & & \boldsymbol{0} \\ & & & & & \boldsymbol{0} \\ & & & & & \boldsymbol{0} \\ \boldsymbol{J_1^{(L)}} & & & & & \boldsymbol{0} \\ & & & & & \boldsymbol{J_1^{(L)}} \end{bmatrix}, \tag{306}$$

where the diagonal blocks of $\boldsymbol{H_0}$ are:

$$\begin{aligned}
\boldsymbol{M_1} &= \boldsymbol{E_0}, \\
\boldsymbol{M_2} &= \boldsymbol{E_0} + \boldsymbol{J_0^{(H)}}, \\
\boldsymbol{M_3} &= \boldsymbol{E_0} + \boldsymbol{E_L} + \boldsymbol{J_0^{(H)}}, \\
\boldsymbol{M_4} &= \boldsymbol{E_0} + \boldsymbol{E_H} + \boldsymbol{E_L} + \boldsymbol{J_0^{(H)}}, \\
\boldsymbol{M_5} &= \boldsymbol{E_0} + \boldsymbol{J_0^{(L)}}, \\
\boldsymbol{M_6} &= \boldsymbol{E_0} + \boldsymbol{E_L} + \boldsymbol{J_0^{(L)}}.
\end{aligned}$$

Matrices $\boldsymbol{H_0}, \boldsymbol{H_L}$ and $\boldsymbol{H_H}$ can be interpreted as follows. The first set of states correspond to case $0,0$. From this state, a high priority arrival moves the Markov chain to state $1,0$, and a low priority arrival to state $0,1$. No service events can occur in $0,0$. From state $1,0$ (second set of states) there are transitions to $1,1+$ due to a low priority arrival ($\boldsymbol{E_L}$), to $2+,0$ due to a high priority arrival ($\boldsymbol{E_H}$), and to $0,0$ due to a high priority service ($\boldsymbol{J_1^{(H)}}$). The latter one is accompanied by a high priority departure event, hence the corresponding matrix block is located in $\boldsymbol{H_H}$. The rest of the blocks can be interpreted similarly.

The steady-state distribution of the MMAP[K]/PH[K]/1 queue just after a departure can be calculated from the stationary distribution as

$$\underline{x}_{i,j} = \begin{cases} \dfrac{\underline{y}_{i+1,j}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}}, & i > 0, j \geq 0 \\ \dfrac{\underline{y}_{1,j}\boldsymbol{J_1^{(H)}} + \underline{y}_{0,j+1}\boldsymbol{J_1^{(L)}}}{\lambda^{(L)}+\lambda^{(H)}}, & i = 0, j \geq 0. \end{cases} \tag{307}$$

The initial probability distribution of the 6 cases, $\underline{v} = [\underline{x}_{0,0}, \underline{x}_{1,0}, \underline{x}_{1,1+}, \underline{x}_{2+,0+}, \underline{x}_{0,1}, \underline{x}_{0,2+}]$, are computed based on (307) and (296) as

$$\underline{x}_{0,0} = \frac{\underline{y}_{1,0}\boldsymbol{J_1^{(H)}} + \underline{y}_{0,1}\boldsymbol{J_1^{(L)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\underline{y}_{0,0}\boldsymbol{R_0^{(L)}}\boldsymbol{J_1^{(H)}} + \underline{y}_{0,1}\boldsymbol{J_1^{(L)}}}{\lambda^{(L)}+\lambda^{(H)}}, \tag{308}$$

$$\underline{x}_{1,0} = \frac{\underline{y}_{2,0}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\underline{y}_{0,0}\boldsymbol{R_0^{(L)}}^2\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}}, \tag{309}$$

$$\begin{aligned}
\underline{x}_{1,1+} &= \frac{\sum_{j=1}^{\infty}\underline{y}_{2,j}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\sum_{j=0}^{\infty}\sum_{k=0}^{j}\sum_{l=0}^{k}\underline{y}_{0,l}\boldsymbol{R_{k-l}^{(L)}}\boldsymbol{R_{j-k}^{(L)}}\boldsymbol{J_1^{(H)}} - \underline{y}_{0,0}\boldsymbol{R_0^{(L)}}^2\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} \\
&= \frac{\sum_{l=0}^{\infty}\underline{y}_{0,l}\hat{\boldsymbol{R}}^{(L)^2}\boldsymbol{J_1^{(H)}} - \underline{y}_{0,0}\boldsymbol{R_0^{(L)}}^2\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\underline{y}_0^{(H)}\hat{\boldsymbol{R}}^{(L)^2}\boldsymbol{J_1^{(H)}} - \underline{y}_{0,0}\boldsymbol{R_0^{(L)}}^2\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}},
\end{aligned} \tag{310}$$

$$\begin{aligned}
\underline{x}_{2+,0+} &= \frac{\sum_{i=3}^{\infty}\sum_{j=0}^{\infty}\underline{y}_{i,j}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\sum_{i=3}^{\infty}\underline{y}_i^{(H)}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{\sum_{i=3}^{\infty}\underline{y}_0^{(H)}\hat{\boldsymbol{R}}^{(L)^i}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}} \\
&= \frac{\underline{y}_0^{(H)}\hat{\boldsymbol{R}}^{(L)^3}(\boldsymbol{I}-\hat{\boldsymbol{R}}^{(L)})^{-1}\boldsymbol{J_1^{(H)}}}{\lambda^{(L)}+\lambda^{(H)}},
\end{aligned} \tag{311}$$

$$\underline{x}_{0,1} = \frac{\underline{y}_{1,1}\boldsymbol{J}_1^{(H)} + \underline{y}_{0,2}\boldsymbol{J}_1^{(L)}}{\lambda^{(L)}+\lambda^{(H)}} = \frac{(\underline{y}_{0,0}\boldsymbol{R}_1^{(L)} + \underline{y}_{0,1}\boldsymbol{R}_0^{(L)})\boldsymbol{J}_1^{(H)} + \underline{y}_{0,2}\boldsymbol{J}_1^{(L)}}{\lambda^{(L)}+\lambda^{(H)}}, \tag{312}$$

$$\begin{aligned}
\underline{x}_{0,2+} &= \frac{\sum_{j=3}^{\infty}\underline{y}_{0,j}\boldsymbol{J}_1^{(L)} + \sum_{j=2}^{\infty}\underline{y}_{1,j}\boldsymbol{J}_1^{(H)}}{\lambda^{(L)}+\lambda^{(H)}} \\
&= \frac{(\underline{y}_0^{(H)} - \underline{y}_{0,0} - \underline{y}_{0,1} - \underline{y}_{0,2})\boldsymbol{J}_1^{(L)} + (\sum_{j=0}^{\infty}\underline{y}_{1,j} - \underline{y}_{0,0}\boldsymbol{R}_0^{(L)} - \underline{y}_{0,0}\boldsymbol{R}_1^{(L)} - \underline{y}_{0,1}\boldsymbol{R}_0^{(L)})\boldsymbol{J}_1^{(H)}}{\lambda^{(L)}+\lambda^{(H)}} \\
&= \frac{(\underline{y}_0^{(H)} - \underline{y}_{0,0} - \underline{y}_{0,1} - \underline{y}_{0,2})\boldsymbol{J}_1^{(L)}}{\lambda^{(L)}+\lambda^{(H)}} \\
&\quad + \frac{(\sum_{j=0}^{\infty}\sum_{k=0}^{j}\underline{y}_{0,k}\boldsymbol{R}_{j-k}^{(L)} - \underline{y}_{0,0}\boldsymbol{R}_0^{(L)} - \underline{y}_{0,0}\boldsymbol{R}_1^{(L)} - \underline{y}_{0,1}\boldsymbol{R}_0^{(L)})\boldsymbol{J}_1^{(H)}}{\lambda^{(L)}+\lambda^{(H)}} \\
&= \frac{(\underline{y}_0^{(H)} - \underline{y}_{0,0} - \underline{y}_{0,1} - \underline{y}_{0,2})\boldsymbol{J}_1^{(L)} + (\underline{y}_0^{(H)}\hat{\boldsymbol{R}}^{(L)} - \underline{y}_{0,0}\boldsymbol{R}_0^{(L)} - \underline{y}_{0,0}\boldsymbol{R}_1^{(L)} - \underline{y}_{0,1}\boldsymbol{R}_0^{(L)})\boldsymbol{J}_1^{(H)}}{\lambda^{(L)}+\lambda^{(H)}},
\end{aligned} \tag{313}$$

where vector $\underline{y}_0^{(H)}$ denotes the probability vector that there are no high priority customers in the system, thus $\underline{y}_0^{(H)} = \sum_{i=0}^{\infty}\underline{y}_{0,i}$.

Having computed the vector $\underline{v}$ and the matrices $\boldsymbol{H_0}$, $\boldsymbol{H_H}$ and $\boldsymbol{H_L}$, the joint moments of the departure process are obtained according to (64), hence

$$\eta_{i,j}^{(c)} = i!j!\,\underline{v}(-\boldsymbol{H_0})^{-i-1}\boldsymbol{H}_c(-\boldsymbol{H_0})^{-j}\mathbb{1}, \quad i,j \geq 0, c = \{H, L\}. \tag{314}$$

7.4.4 *An efficient, truncation-free procedure*

Section 7.4.2 describes how to calculate the steady state joint distribution of the number of customers, based on which Section 7.4.3 derives the joint moments of the inter-departure times. These results, however, are not straight forward to implement in an efficient way. The potential numerical pitfalls are:

- To obtain matrix $\boldsymbol{G}_{\boldsymbol{H_0}}$ the matrix polynomial equation (300) has to be solved, but this matrix equation has infinitely many terms, and relies on infinitely many elements of matrix series $\boldsymbol{G}_{\boldsymbol{i}}^{(L)}$.
- To obtain vectors $\underline{y}_{0,j}, j \geq 0$, matrices $\boldsymbol{T_i}$ are required (see (301)), that are defined by infinite summations (302).
- To obtain vectors $\underline{y}_{i,j}, i > 0, j \geq 0$, a convolution like formula needs to be evaluated that gets slower and slower with increasing i (see (296)).
- To obtain vector $\underline{v}$, vectors like $\underline{x}_{2+,0+}$ are necessary to compute, which involves infinite summations of vectors $\underline{y}_{0,j}$.

The naive numerical implementation of such a procedure evaluates the infinite summations by truncation. In order to avoid the loss of accuracy, the truncation threshold must be high, depending on the parameters (in particular, the load) of the system, which makes the procedure slow.

Fortunately, it is possible to develop a numerically efficient procedure that addresses all the above listed critical steps without using any truncation. The efficient solution of (300) is addressed in Section 7.4.4.1, while the accurate calculation of the vectors $\underline{y}_{0,i}$ and further related quantities is discussed in Section 7.4.4.2.

7.4.4.1 *Two fundamental matrices and their relations*

There are two matrices that play key roles in the efficient analysis of the system. One of them is $\boldsymbol{G}_{\boldsymbol{H_0}}$, that is the solution of (300), and the other one is matrix $\boldsymbol{Z}$, defined by

$$\boldsymbol{Z} = \sum_{i=0}^{\infty} \boldsymbol{G}_{\boldsymbol{i}}^{(\boldsymbol{L})} \boldsymbol{G}_{\boldsymbol{H_0}}{}^{i}. \tag{315}$$

Theorem 33. *If the algebraic multiplicities of the eigenvalues of matrices $\boldsymbol{G}_{\boldsymbol{H_0}}$ and $\boldsymbol{Z}$ are one, $\boldsymbol{G}_{\boldsymbol{H_0}}\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{G}_{\boldsymbol{H_0}}$ holds.*

To prove this theorem we first need the following Lemma:

Lemma 7. *If the algebraic multiplicities of the eigenvalues of matrix $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ and matrix $\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{E_H}\boldsymbol{G}^{(\boldsymbol{L})}(z)$ are one, the eigenvectors of $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ and $\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{E_H}\boldsymbol{G}^{(\boldsymbol{L})}(z)$ are the same.*

Proof of the Lemma. The proof uses the same techniques as in [47].

Let ν_i and $\underline{u}_i$ be the eigenvalue and the corresponding right eigenvector of $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ (for simplicity we assume that $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ has distinct eigenvalues). As $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ satisfies the matrix-quadratic equation of (295), ν_i satisfies

$$\det\left[\boldsymbol{J}_{\boldsymbol{1}}^{(\boldsymbol{H})} + (\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{J}_{\boldsymbol{0}}^{(\boldsymbol{H})})\nu_i + \boldsymbol{E_H}\nu_i^2\right] = 0, \tag{316}$$

and the associated right eigenvector $\underline{u}_i$ is the solution of

$$\left[\boldsymbol{J}_{\boldsymbol{1}}^{(\boldsymbol{H})} + (\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{J}_{\boldsymbol{0}}^{(\boldsymbol{H})})\nu_i + \boldsymbol{E_H}\nu_i^2\right] \cdot \underline{u}_i = \underline{0}. \tag{317}$$

Note that both ν_i and vectors $\underline{u}_i$ are functions of z.

By substituting (284) into (316) and by some basic manipulations we get

$$\det\left[\left(\underbrace{(\nu_i\boldsymbol{D_0} + \nu_i z\boldsymbol{D_L} + \nu_i^2\boldsymbol{D_H})}_{\boldsymbol{M_1}} \oplus \underbrace{(\underline{s}_H\underline{\sigma}_H + \nu_i\boldsymbol{S_H})}_{\boldsymbol{M_2}}\right) \otimes \boldsymbol{I}\right] = 0, \tag{318}$$

from which it follows that $\boldsymbol{M_1} \oplus \boldsymbol{M_2}$ has a zero eigenvalue. Let δ_j and σ_k denote the eigenvalues of $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$, respectively. Since the eigenvalues of $\boldsymbol{M_1} \oplus \boldsymbol{M_2}$ are $\delta_j + \sigma_k$, to have a zero eigenvalue there must exist j' and k' such that $\delta_{j'} = -\sigma_{k'}$. The eigenvector of $\boldsymbol{M_1}$ belonging to $\delta_{j'}$ is denoted by $\underline{\theta}_{j'}$, the one of $\boldsymbol{M_2}$ belonging to $\sigma_{k'}$ is denoted by $\underline{\psi}_{k'}$. Let us introduce $\underline{\phi}_i = \underline{\theta}_{j'} \otimes \underline{\psi}_{k'} \otimes \mathbb{1}$.

Now we show that $\underline{\phi}_i$ is an eigenvector of $\boldsymbol{G}^{(\boldsymbol{L})}(z)$ associated with ν_i, thus it satisfies (317):

$$\begin{aligned}
&\left[\boldsymbol{J}_{\boldsymbol{1}}^{(\boldsymbol{H})} + (\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{J}_{\boldsymbol{0}}^{(\boldsymbol{H})})\nu_i + \boldsymbol{E_H}\nu_i^2\right] \cdot \underline{\phi}_i \\
&\quad = [\boldsymbol{I} \otimes \boldsymbol{M_2} \otimes \boldsymbol{I} + \boldsymbol{M_1} \otimes \boldsymbol{I} \otimes \boldsymbol{I}] \cdot (\underline{\theta}_{j'} \otimes \underline{\psi}_{k'} \otimes \mathbb{1}) \\
&\quad = \underline{\theta}_{j'} \otimes (\sigma_{k'}\underline{\psi}_{k'}) \otimes \mathbb{1} + (\delta_{j'}\underline{\theta}_{j'}) \otimes \underline{\psi}_{k'} \otimes \mathbb{1} = \underline{0}.
\end{aligned} \tag{319}$$

Next, we show that $\underline{\phi}_i$ is an eigenvector of $\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{E_H}\boldsymbol{G}^{(L)}(z)$:

$$\begin{aligned}&\left[\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{E_H}\boldsymbol{G}^{(L)}(z)\right] \cdot \underline{\phi}_i = [\boldsymbol{E_0} + z\boldsymbol{E_L} + \boldsymbol{E_H}\nu_i] \cdot \underline{\phi}_i \\ &= [\boldsymbol{D_0} \otimes \boldsymbol{I} \otimes \boldsymbol{I} + z\boldsymbol{D_L} \otimes \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{D_H}\nu_i \otimes \boldsymbol{I} \otimes \boldsymbol{I}] \cdot (\underline{\theta}_{j'} \otimes \underline{\psi}_{k'} \otimes \mathbb{1}) \\ &= (\boldsymbol{M_1}/\nu_i \otimes \boldsymbol{I} \otimes \boldsymbol{I}) \cdot (\underline{\theta}_{j'} \otimes \underline{\psi}_{k'} \otimes \mathbb{1}) = \delta_{j'}/\nu_i \cdot \underline{\phi}_i.\end{aligned} \tag{320}$$

□

Proof of Theorem 33. First observe that matrices $\boldsymbol{G_i^{(L)}}$ can be written as $\boldsymbol{G_i^{(L)}} = \tilde{\boldsymbol{G}}_{\boldsymbol{i}}^{(L)} \otimes \boldsymbol{I}$ since the service process of the low priority class is stopped during the busy period of the high priority class.

The proof will be similar to the one of Lemma 7, using the same techniques as in [47] again.

Let λ_k and $\underline{v}_k$ be the eigenvalue and the corresponding right eigenvector of $\boldsymbol{G_{H_0}}$ (for simplicity we assume that $\boldsymbol{G_{H_0}}$ has distinct eigenvalues). As $\boldsymbol{G_{H_0}}$ satisfies the matrix equation of (300), λ_k satisfies

$$\det\left[\boldsymbol{J_1^{(L)}} + (\boldsymbol{E_0} + \boldsymbol{J_0^{(L)}})\lambda_k + \boldsymbol{E_L}\lambda_k^2 + \boldsymbol{E_H}\sum_{i=0}^{\infty}(\tilde{\boldsymbol{G}}_{\boldsymbol{i}}^{(L)} \otimes \boldsymbol{I})\lambda_k^i \cdot \lambda_k\right] = 0, \tag{321}$$

and the associated right eigenvector $\underline{v}_k$ is the solution of

$$\left[\boldsymbol{J_1^{(L)}} + (\boldsymbol{E_0} + \boldsymbol{J_0^{(L)}})\lambda_k + \boldsymbol{E_L}\lambda_k^2 + \boldsymbol{E_H}\sum_{i=0}^{\infty}(\tilde{\boldsymbol{G}}_{\boldsymbol{i}}^{(L)} \otimes \boldsymbol{I})\lambda_k^i \cdot \lambda_k\right] \cdot \underline{v}_k = \underline{0}. \tag{322}$$

By substituting (284) into (321) and by some basic manipulation we get

$$\det\Bigg[\underbrace{\left(\lambda_k \boldsymbol{D_0}\otimes\boldsymbol{I} + \lambda_k^2\boldsymbol{D_L}\otimes\boldsymbol{I} + \sum_{i=0}^{\infty}\lambda_k^{i+1}(\boldsymbol{D_H}\otimes\boldsymbol{I})\tilde{\boldsymbol{G}}_{\boldsymbol{i}}^{(L)}\right)}_{\boldsymbol{N_1}} \oplus \underbrace{(\underline{s}_L\underline{\sigma}_L + \lambda_k\boldsymbol{S_L})}_{\boldsymbol{N_2}}\Bigg] = 0,$$

from which it follows that $\boldsymbol{N_1} \oplus \boldsymbol{N_2}$ has a zero eigenvalue. Let α_j and β_h denote the eigenvalues of $\boldsymbol{N_1}$ and $\boldsymbol{N_2}$, respectively. Since the eigenvalues of $\boldsymbol{N_1} \oplus \boldsymbol{N_2}$ are $\alpha_j + \beta_h$, to have a zero eigenvalue there must exist j' and h' such that $\alpha_{j'} = -\beta_{h'}$. The eigenvector of $\boldsymbol{N_1}$ belonging to $\alpha_{j'}$ is denoted by $\underline{\zeta}_{j'}$, the one of $\boldsymbol{N_2}$ belonging to $\beta_{h'}$ is denoted by $\underline{\xi}_{h'}$. Let us introduce $\underline{\mu}_k = \underline{\zeta}_{j'} \otimes \underline{\xi}_{h'}$.

Now we show that $\underline{\mu}_k$ is an eigenvector of $\boldsymbol{G_{H_0}}$, thus it satisfies (322):

$$\begin{aligned}(\boldsymbol{N_1} \oplus \boldsymbol{N_2}) \cdot \underline{\mu}_k &= (\boldsymbol{N_1} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{N_2}) \cdot (\underline{\zeta}_{j'} \otimes \underline{\xi}_{h'}) = \\ &= \alpha_{j'}\underline{\zeta}_{j'} \otimes \underline{\xi}_{h'} + \underline{\zeta}_{j'} \otimes (\beta_{h'}\underline{\xi}_{h'}) = \underline{0}.\end{aligned} \tag{323}$$

Next, we show that $\underline{\mu}_k$ is an eigenvector of $\boldsymbol{Z}$. Observe that $\underline{\mu}_k$ is an eigenvector of $\boldsymbol{Z}$ if and only if it is an eigenvector of $\boldsymbol{G}^{(L)}(z)|_{z=\lambda_k}$ since

$$\boldsymbol{Z} \cdot \underline{\mu}_k = \sum_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}\boldsymbol{G_{H_0}}^i \cdot \underline{\mu}_k = \sum_{i=0}^{\infty}\boldsymbol{G_i^{(L)}}\lambda_k^i\underline{\mu}_k = \boldsymbol{G}^{(L)}(z)|_{z=\lambda_k} \cdot \underline{\mu}_k. \tag{324}$$

As Lemma 7 states that the eigenvectors of matrix $\boldsymbol{G}^{(L)}(z)$ and matrix $\boldsymbol{E_0}+z\boldsymbol{E_L}+\boldsymbol{E_H}\boldsymbol{G}^{(L)}(z)$ are the same, it is enough to prove that $\underline{\mu}_k$ is an eigenvector of $(\boldsymbol{E_0}+z\boldsymbol{E_L}+\boldsymbol{E_H}\boldsymbol{G}^{(L)}(z))|_{z=\lambda_k}$:

$$\begin{aligned}&(\boldsymbol{E_0}+\lambda_k\boldsymbol{E_L}+\boldsymbol{E_H}\boldsymbol{G}^{(L)}(\lambda_k))\cdot\underline{\mu}_k\\&=\left[(\boldsymbol{D_0}\otimes\boldsymbol{I}\otimes\boldsymbol{I}+\lambda_k\boldsymbol{D_L}\otimes\boldsymbol{I}\otimes\boldsymbol{I}+\sum_{i=0}^{\infty}\lambda_k^i(\boldsymbol{D_H}\otimes\boldsymbol{I}\otimes\boldsymbol{I})(\tilde{\boldsymbol{G}}_i^{(L)}\otimes\boldsymbol{I})\right]\\&\quad\cdot(\underline{\zeta}_{j'}\otimes\underline{\xi}_{h'})\\&=[\boldsymbol{N_1}/\lambda_k\otimes\boldsymbol{I}]\cdot(\underline{\zeta}_{j'}\otimes\underline{\xi}_{h'})=\alpha_{j'}/\lambda_k(\underline{\zeta}_{j'}\otimes\underline{\xi}_{h'}).\end{aligned}\tag{325}$$

As $\boldsymbol{G_{H_0}}$ and $\boldsymbol{Z}$ have the same eigenvectors, the same matrix diagonalizes them, consequently they commute. □

Note that the theorem can be generalized to the case when the eigenvalues are not distinct, but it requires the detailed discussion of the combination of the multiplicities of the eigenvalues of $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ ($\boldsymbol{N_1}$ and $\boldsymbol{N_2}$) that we neglect here.

Based on this commutative property, the next theorem enables the efficient computation of $\boldsymbol{G_{H_0}}$ and $\boldsymbol{Z}$.

Theorem 34. *Matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{Z}$ satisfy the following coupled matrix-quadratic equations:*

$$\begin{aligned}\boldsymbol{0}&=\boldsymbol{J_1}^{(H)}+(\boldsymbol{E_0}+\boldsymbol{J_0}^{(H)})\boldsymbol{Z}+\boldsymbol{E_L}\boldsymbol{G_{H_0}}\boldsymbol{Z}+\boldsymbol{E_H}\boldsymbol{Z}^2,\\\boldsymbol{0}&=\boldsymbol{J_1}^{(L)}+(\boldsymbol{E_0}+\boldsymbol{J_0}^{(L)})\boldsymbol{G_{H_0}}+\boldsymbol{E_H}\boldsymbol{Z}\boldsymbol{G_{H_0}}+\boldsymbol{E_L}\boldsymbol{G_{H_0}}^2.\end{aligned}\tag{326}$$

Proof. To obtain equations for $\boldsymbol{Z}$, we multiply (290) by $\boldsymbol{G_{H_0}}^i$ from the right, sum it from $i=1$ to ∞, and add (289) to it. By using (315) we get

$$\boldsymbol{0}=\boldsymbol{J_1}^{(H)}+\boldsymbol{E_L}\boldsymbol{Z}\boldsymbol{G_{H_0}}+(\boldsymbol{E_0}+\boldsymbol{J_0}^{(H)})\boldsymbol{Z}+\boldsymbol{E_H}\sum_{i=0}^{\infty}\sum_{k=0}^{i}\boldsymbol{G_k}^{(L)}\boldsymbol{G_{i-k}}^{(L)}\boldsymbol{G_{H_0}}^i.\tag{327}$$

The last term becomes $\boldsymbol{E_H}\boldsymbol{Z}^2$ by swapping the sums and exploiting that $\boldsymbol{G_{H_0}}$ and $\boldsymbol{Z}$ commute, providing the first matrix quadratic equation. The second matrix quadratic equation can be obtained from (300), when the definition of $\boldsymbol{Z}$ is applied. □

Interestingly, the two matrix equations show perfect symmetry. While the solution of coupled Sylvester equations has an extensive literature (e.g. [27, 60, 89] are recent methods), there are no methods available for coupled matrix quadratic equations (according to our best knowledge). A very simple method is given by Algorithm 6. (We are sure that more efficient solution methods can be developed as well, but even this simple method performs very well in our numerical examples.)

Note that Algorithm 6 needs only successive solution of matrix quadratic equations, that is much more efficient than the direct application of the results of Section 7.4.2 both in time and in space requirement, since the infinite series of $\boldsymbol{G_i}^{(L)}$ matrices and the solution of (300) are not needed.

Although the results above have been derived in an algebraic way, matrices $\boldsymbol{G_{H_0}}$ and $\boldsymbol{Z}$ have important probabilistic interpretations. Let us denote by (n_H, n_L) the set of states in which there are n_H high and n_L low priority customers in the queue. Then, entry (a,b) of matrix $\boldsymbol{G_{H_0}}$ is the conditional probability that starting from state a in $(0,1)$ the first visit to $(0,0)$ occurs in state b. Similarly, the entry (a,b) of matrix $\boldsymbol{Z}$ is the conditional probability that

Algorithm 6 Solving the coupled matrix quadratic equations of (326)

procedure $\boldsymbol{G}_{H_0}, \boldsymbol{Z}$ = SOLVECOUPLED($\boldsymbol{E}_0, \boldsymbol{E}_H, \boldsymbol{E}_L, \boldsymbol{J}_0^{(H)}, \boldsymbol{J}_1^{(H)}, \boldsymbol{J}_0^{(L)}, \boldsymbol{J}_1^{(L)}$)
  $k \leftarrow 0$
  ${\boldsymbol{G}_{H_0}}^{(0)} \leftarrow \boldsymbol{I}$
  repeat
    Solve $\boldsymbol{0} = \boldsymbol{J}_1^{(H)} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} + \boldsymbol{E}_L {\boldsymbol{G}_{H_0}}^{(k)})\boldsymbol{Z}^{(k+1)} + \boldsymbol{E}_H {\boldsymbol{Z}^{(k+1)}}^2$ for $\boldsymbol{Z}^{(k+1)}$
    Solve $\boldsymbol{0} = \boldsymbol{J}_1^{(L)} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(L)} + \boldsymbol{E}_H \boldsymbol{Z}^{(k+1)}){\boldsymbol{G}_{H_0}}^{(k+1)} + \boldsymbol{E}_L {\boldsymbol{G}_{H_0}}^{(k+1)^2}$ for ${\boldsymbol{G}_{H_0}}^{(k+1)}$
    $k \leftarrow k+1$
  until $\|{\boldsymbol{G}_{H_0}}^{(k)} - {\boldsymbol{G}_{H_0}}^{(k-1)}\| < \epsilon$ and $\|\boldsymbol{Z}^{(k)} - \boldsymbol{Z}^{(k-1)}\| < \epsilon$
  return ${\boldsymbol{G}_{H_0}}^{(k)}$, $\boldsymbol{Z}^{(k)}$
end procedure

starting from state a in $(1,0)$ the first visit to $(0,0)$ occurs in state b. Using these probabilistic interpretations it is easy to see that $\boldsymbol{Z}$ and $\boldsymbol{G}_{H_0}$ commute: let us investigate the busy period generated by a high and a low priority customer, thus the system is in $(1,1)$ initially. Since the probability that the first passage to $(0,0)$ occurs in state b is not affected by the order of service, we immediately have that $\boldsymbol{G}_{H_0}\boldsymbol{Z} = \boldsymbol{Z}\boldsymbol{G}_{H_0}$.

7.4.4.2 *The boundary distribution*

Matrices $\boldsymbol{T}_i, i \geq 0$, that are necessary to compute boundary probabilities $\underline{y}_{0,i}$, are defined by infinite summations (see (302)). To obtain matrices $\boldsymbol{T}_i$ efficiently we introduce closely related matrices $\boldsymbol{A}_n$ defined by

$$\boldsymbol{A}_n = \sum_{i=n}^{\infty} \boldsymbol{G}_i^{(L)} {\boldsymbol{G}_{H_0}}^{i-n}, \quad n \geq 0. \tag{328}$$

Theorem 35. *Matrices $\boldsymbol{A}_n$ can be obtained recursively as the solutions of discrete Sylvester-type matrix equations*

$$\begin{aligned} \boldsymbol{0} = (-\boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)} - \boldsymbol{E}_H \boldsymbol{G}_0^{(L)})^{-1} \left(\boldsymbol{E}_L \boldsymbol{A}_{n-1} + \boldsymbol{E}_H \sum_{k=1}^{n-1} \boldsymbol{G}_k^{(L)} \boldsymbol{A}_{n-k} \right) - \boldsymbol{A}_n \\ + (-\boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)} - \boldsymbol{E}_H \boldsymbol{G}_0^{(L)})^{-1} \boldsymbol{E}_H \boldsymbol{A}_n \boldsymbol{Z} \end{aligned} \tag{329}$$

for $n > 0$, and for $n = 0$ we have that $\boldsymbol{A}_0 = \boldsymbol{Z}$.

Proof. The case of $n = 0$ is trivial (see (315) for the definition of $\boldsymbol{Z}$).

To derive equations for $n > 0$, let us multiply equations (290) by ${\boldsymbol{G}_{H_0}}^{i-n}$ from the right and sum them up from $i = n$ to ∞. We get

$$\boldsymbol{0} = \boldsymbol{E}_L \underbrace{\sum_{i=n}^{\infty} \boldsymbol{G}_{i-1}^{(L)} {\boldsymbol{G}_{H_0}}^{i-n}}_{\boldsymbol{A}_{n-1}} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)}) \underbrace{\sum_{i=n}^{\infty} \boldsymbol{G}_i^{(L)} {\boldsymbol{G}_{H_0}}^{i-n}}_{\boldsymbol{A}_n} + \boldsymbol{E}_H \sum_{i=n}^{\infty} \sum_{k=0}^{i} \boldsymbol{G}_k^{(L)} \boldsymbol{G}_{i-k}^{(L)} {\boldsymbol{G}_{H_0}}^{i-n}.$$

The last term can be transformed further by swapping the order of the two summations, leading to

$$\boldsymbol{E}_H \sum_{i=n}^{\infty} \sum_{k=0}^{i} \boldsymbol{G}_k^{(L)} \boldsymbol{G}_{i-k}^{(L)} \boldsymbol{G}_{H_0}{}^{i-n} = \boldsymbol{E}_H \sum_{k=0}^{n-1} \boldsymbol{G}_k^{(L)} \underbrace{\sum_{i=n}^{\infty} \boldsymbol{G}_{i-k}^{(L)} \boldsymbol{G}_{H_0}{}^{i-n}}_{\boldsymbol{A}_{n-k}} + \underbrace{\boldsymbol{E}_H \sum_{k=n}^{\infty} \boldsymbol{G}_k^{(L)} \underbrace{\sum_{i=k}^{\infty} \boldsymbol{G}_{i-k}^{(L)} \boldsymbol{G}_{H_0}{}^{i-k}}_{\boldsymbol{Z}} \boldsymbol{G}_{H_0}{}^{k-n}}_{\boldsymbol{A}_n \boldsymbol{Z}}, \tag{330}$$

where the commuting property of matrices $\boldsymbol{Z}$ and $\boldsymbol{G}_{H_0}$ was utilized. Collecting all parts of the equation gives

$$\boldsymbol{0} = \boldsymbol{E}_L \boldsymbol{A}_{n-1} + \boldsymbol{E}_H \sum_{k=1}^{n-1} \boldsymbol{G}_k^{(L)} \boldsymbol{A}_{n-k} + (\boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} + \boldsymbol{E}_H \boldsymbol{G}_0^{(L)}) \boldsymbol{A}_n + \boldsymbol{E}_H \boldsymbol{A}_n \boldsymbol{Z}. \tag{331}$$

Multiplying by $(-\boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)} - \boldsymbol{E}_H \boldsymbol{G}_0^{(L)})^{-1}$ from the left yields the traditional form of discrete Sylvester equations, establishing the theorem. □

Observe that the Sylvester equation for $\boldsymbol{A}_n$ depends only on matrices $\boldsymbol{A}_i, i < n$, that are already available in step n.

A consequence of Theorem 35 is that matrices $\boldsymbol{T}_i$ can be expressed explicitly.

Corollary 18. *Matrices $\boldsymbol{T}_i, i \geq 0$, can be obtained as*

$$\boldsymbol{T}_i = \begin{cases} \boldsymbol{E}_H \boldsymbol{A}_i, & \text{for } i > 1, \\ \boldsymbol{E}_L + \boldsymbol{E}_H \boldsymbol{A}_1, & \text{for } i = 1, \\ \boldsymbol{E}_0 + \boldsymbol{J}_0^{(H)} + \boldsymbol{E}_L \boldsymbol{G}_{H_0} + \boldsymbol{E}_H \boldsymbol{Z}, & \text{for } i = 0. \end{cases} \tag{332}$$

Finally, it is possible to solve the sum of matrices $\boldsymbol{A}_n$ explicitly as well, that will be useful later.

Corollary 19. *The sum of matrices $\boldsymbol{A}_i$, denoted by $\hat{\boldsymbol{A}} = \sum_{i=0}^{\infty} \boldsymbol{A}_i$, is the solution of the discrete Sylvester equation*

$$\begin{aligned} \boldsymbol{0} = {} & (-\boldsymbol{E}_L - \boldsymbol{E}_H \hat{\boldsymbol{G}}^{(L)} - \boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)})^{-1} (-\boldsymbol{E}_H \boldsymbol{Z} - \boldsymbol{E}_H \hat{\boldsymbol{G}}^{(L)} - \boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)}) \boldsymbol{Z} - \hat{\boldsymbol{A}} \\ & + (-\boldsymbol{E}_L - \boldsymbol{E}_H \hat{\boldsymbol{G}}^{(L)} - \boldsymbol{E}_0 - \boldsymbol{J}_0^{(H)})^{-1} \boldsymbol{E}_H \hat{\boldsymbol{A}} \boldsymbol{Z}. \end{aligned} \tag{333}$$

Proof. Summing equations (329) from $n = 1$ to ∞, swapping the sums, and applying the definition of $\hat{\boldsymbol{A}}$ provides the corollary after some simple algebraic manipulations. □

Corollary 20. *The sum of matrices $\boldsymbol{T}_i$, denoted by $\hat{\boldsymbol{T}} = \sum_{i=0}^{\infty} \boldsymbol{T}_i$, is given by*

$$\hat{\boldsymbol{T}} = \boldsymbol{E}_0 + \boldsymbol{J}_1^{(H)} + \boldsymbol{E}_L \boldsymbol{G}_{H_0} + \boldsymbol{E}_L + \boldsymbol{E}_H \hat{\boldsymbol{A}}. \tag{334}$$

Proof. The corollary follows from Corollaries 18 and 19. □

7.4.4.3 *Calculating the joint moments of the departure process*

The steps to obtain the quantities necessary for the departure process analysis are as follows.

1. First solve matrices $\boldsymbol{G}_{\boldsymbol{H}_0}$ and $\boldsymbol{Z}$ with Algorithm 6.
2. Solve the matrix quadratic equations to obtain matrices $\hat{\boldsymbol{G}}^{(L)}$ and $\hat{\boldsymbol{R}}^{(L)}$.
3. Calculate matrices $\boldsymbol{G}_{\boldsymbol{i}}$ for $i = 0, 1, 2$ based on (289) and (290), matrices $\boldsymbol{R}_{\boldsymbol{i}}$ for $i = 0, 1$ based on (291) and (292), matrices $\boldsymbol{A}_{\boldsymbol{i}}$ for $i = 1, 2$ based on (329), and matrices $\boldsymbol{T}_{\boldsymbol{i}}$ for $i = 1, 2$ based on Corollary 18.
4. Compute matrix $\hat{\boldsymbol{A}}$ by solving (333) and matrix $\hat{\boldsymbol{T}}$ from (334).

After this preparation the required stationary probabilities are calculated.

- Vectors $\underline{y}_{0,0}, \underline{y}_{0,1}$ and $\underline{y}_{0,2}$ are computed directly from (304) and (301) using the $\boldsymbol{T}_{\boldsymbol{i}}$ matrices computed above.
- To derive vector $\underline{y}_0^{(H)} = \sum_{i=0}^{\infty} \underline{y}_{0,i}$, equations (301) are summed up from $i = 1$ to ∞, and (304) is added leading to

$$\underline{0} = \underbrace{\sum_{i=1}^{\infty} \sum_{k=0}^{i} \underline{y}_{0,k} \boldsymbol{T}_{\boldsymbol{i-k}}}_{\text{from (301)}} + \underbrace{\underline{y}_{0,0} \boldsymbol{T}_{\boldsymbol{0}} - \underline{y}_{0,0} \boldsymbol{J}_{\boldsymbol{0}}^{(L)}}_{\text{from (304)}} = \underbrace{\sum_{k=0}^{\infty} \underline{y}_{0,k}}_{\underline{y}_0^{(H)}} \underbrace{\sum_{i=k}^{\infty} \boldsymbol{T}_{\boldsymbol{i-k}}}_{\hat{\boldsymbol{T}}} - \underline{y}_{0,0} \boldsymbol{J}_{\boldsymbol{0}}^{(L)}, \tag{335}$$

  implying $\underline{y}_0^{(H)} = \underline{y}_{0,0} \boldsymbol{J}_{\boldsymbol{0}}^{(L)} \hat{\boldsymbol{T}}^{-1}$.

Using the quantities calculated so far, it is possible to obtain all the six components of vector $\underline{v}$, namely $\underline{x}_{0,0}, \underline{x}_{1,0}, \underline{x}_{1,1+}, \underline{x}_{2+,0+}, \underline{x}_{0,1}$ and $\underline{x}_{0,2+}$. The joint moments of the departure process are given by (314).

dc_1412_17

Part III

QUEUEING NETWORKS

dc_1412_17

8

QUEUEING NETWORK ANALYSIS BASED ON THE JOINT MOMENTS

8.1 TRAFFIC BASED DECOMPOSITION FOR THE ANALYSIS OF QUEUEING NETWORKS

At the beginning the queueing network models were restricted to have Poisson arrival processes with iid. service times in the nodes. More general and multi-class queueing network models are also available for a while [7], but the inter-dependency of the traffic classes is not captured in these classical models. Apart of the exact analytical results which are based on the product form of the stationary distribution of the number of customers at the queueing nodes, a set of approximate analysis methods were developed. One of the most commonly applied approximate analysis methods for queueing networks is the *traffic based decomposition*, where the nodes of the queueing network are evaluated sequentially, in isolation [42]. A node analysis is composed by four main steps (also depicted in Figure 35),

1. the aggregation of the input streams of the node,
2. the queueing analysis of node (computing the performance measures),
3. the characterization of the departure process,
4. and splitting the departure process according to the routing of the traffic.

The evolution of MAP and MMAP based traffic models allowed the extension of the traffic based queueing network analysis to cope with dependent inter-arrival and service times [42, 43]. An important benefit of MAPs and MMAPs is that from the four node analysis steps listed above, three can be performed in an exact and efficient way. These traffic models are closed for the aggregation and splitting operations (see Sections 3.1.1 and 3.1.2), and the related queueing models can be solved by matrix-analytic methods. The only step where an approximation is needed is the modeling of the departure process. The accuracy of the queueing network analysis depends on who well the departure process of the queues is characterized.

This chapter puts the different parts of the dissertation together. The MAP and MMAP characterization results described in Chapter 3 will be used for the internal traffic representation and basic traffic operations (the first and the fourth step in the list above), while Chapters 5,

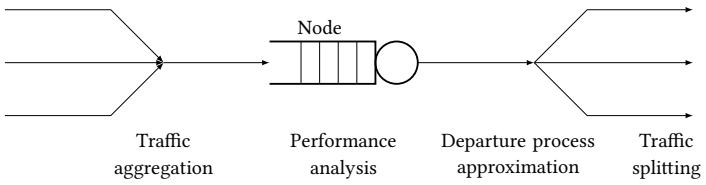


Figure 35.: Steps of the node analysis in traffic-based decomposition

6 and 7 provide the apparatus for the performance analysis of the nodes (step 2) and the characterization of the departure process (step 3).

In the rest of the chapter we restrict our attention to feed-forward queueing networks without loops, where the stability condition is satisfied for all nodes.

8.2 SINGLE-TYPE QUEUEING NETWORKS

Several approaches are available in the literature to model the departure process of MAP/MAP/1 queues by a MAP. This section provides a numerical study to compare the noteworthy algorithms published in the past with the lag-1 joint moment based approach presented by this dissertation.

8.2.1 *MAP based approximations for the departure process*

A MAP is a suitable approximation for the departure process of a node if it satisfies the following properties:

- Obviously, it must capture the behavior of the departure process reasonably well.
- The approximating MAP must have a Markovian representation, otherwise it is impossible to decide if it is a valid process or not. Remember that all performance measures obtained with an invalid arrival process are invalid.
- The MAP representation for the departure process should not be considerably larger than the one for the input process. Otherwise the MAP describing the traffic gets larger and larger every time it passes through a node, making the analysis of large queueing networks problematic.

The methods involved in the comparison study are introduced below.

POISSON MODEL

According to the simplest possible method the network traffic is approximated by a Poisson process, hence the representation of the MAP describing the departure process of node i is given by

$$\boldsymbol{H_0}^{(i)} = -\lambda^{(i)}, \quad \boldsymbol{H_1}^{(i)} = \lambda^{(i)}, \tag{336}$$

where $\lambda^{(i)}$ is the traffic rate of the node. The Poisson approximation is included in the comparison only to demonstrate the effect of ignoring the burstiness and the correlation of the traffic to the accuracy of the approximation.

This approximation is always Markovian ($\lambda^{(i)} > 0$), and the representation of the departure process is not larger than the one of the arrival process (both are of size 1).

SCALED SERVICE PROCESS MODELL

According to this model (introduced in [22]), the departure process is obtained from the service process with appropriate scaling, thus

$$\boldsymbol{H_0}^{(i)} = \rho^{(i)}\boldsymbol{S_0}^{(i)}, \quad \boldsymbol{H_1}^{(i)} = \rho^{(i)}\boldsymbol{S_1}^{(i)}, \tag{337}$$

where $\rho^{(i)}$ is the utilization of node i. This approximation is accurate when $\rho^{(i)} \to 1$, since in this case there are customers in the queue in almost all of the time, implying that the inter-departure times are given by the service times. At lower load, however, the accuracy of the approximation is expected to be worse.

This departure model is always Markovian, and its size always equals the size of the service process, hence the traffic model does not grow when passing through many nodes after each other.

BUSY PERIOD ANALYSIS BASED METHOD

The departure process is modeled in a unique way in [42]. It is based on the observation that the inter-departure times are equal to the service times as long as the queue is busy, and they are equal to a remaining arrival time plus a service time if the last customer left the system idle. According to the procedure in [42] a 2-phase discrete PH distribution is created to fit the first three moments of the number of departures in a busy period, and a MAP is constructed for the departure process along the aforementioned interpretation.

The busy period based method always gives a Markovian representation, but the model for the departure process is larger (has more phases) than the one of the arrival process, hence the MAP characterizing the traffic grows every time when it goes through a queue. Consequently, only queueing networks with a low number of nodes can be analyzed by this method.

TRUNCATION BASED METHODS

As mentioned in Section 5.3 the exact departure process of a MAP/MAP/1 queue is a MAP with infinitely many phases. There were several efforts to truncate this infinite process (including [10],[74]), with perhaps the most sophisticated one being the ETAQA truncation ([44]). The ETAQA truncation preserves the joint densities of the inter-departure time up to the truncation level exactly. The construction of the MAP matrices of the departure process are provided in Section 5.3.2.

The ETAQA truncation model has two important drawbacks. First, it gives a non-Markovian representation in almost all cases. The other issue is that the resulting representation is huge, making the analysis of even the simplest tandem networks numerically challenging.

LAG-1 JOINT MOMENT BASED APPROACH

As proven in Section 3.2, size N MAPs can be characterized by N^2 marginal moments and joint moments. According to this approach a number of such moments are calculated from the infinite, but exact MAP representation of the departure process and a finite, but approximate MAP representation is created based on these moments. The construction of the matrices characterizing this MAP is detailed in Section 5.3.3.

The representation created this way is, however, typically non-Markovian. The transformation method introduced in Section 3.2.4 solves this issue in many cases. In case this transformation fails, the procedures presented in Section 3.3 can be applied to find the closest possible MAP according to the joint moment distance (Section 3.3.3) and according to the distance of the joint densities (Section 3.3.4), respectively.

As a result, the representation is always Markovian, and the size is very compact, independent on the size of the input and service processes of the queue.

Figure 36.: A tandem queueing network with two nodes

8.2.2 *Numerical results with a tandem network*

For the first numerical study let us consider a simple 2-node tandem queueing network as shown by Figure 36. Traffic entering the network is directed to node 1, after getting served it is forwarded to node 2.

The parameters of the system are taken from real traffic measurements. For the first scenario the input MAP has been created from the inter-arrival times of the LBL trace based on 5 marginal and 2×2 joint moments using the moment matching procedure described in Section 3.2. The matrices of this MAP are

$$\boldsymbol{D_0}^{(LBL)} = \begin{bmatrix} -448.802 & 47.2474 & 4.0836 \\ 18.261 & -112.979 & 0.5703 \\ 5.644 & 5.7994 & -50.103 \end{bmatrix}, \quad \boldsymbol{D_1}^{(LBL)} = \begin{bmatrix} 395.093 & 1.701 & 0.677 \\ 9.577 & 82.983 & 1.5877 \\ 4.1416 & 21.419 & 13.099 \end{bmatrix}. \tag{338}$$

With these parameters the arrival rate of the packets is $\lambda = 188.29$, the squared coefficient of variation of the inter-arrival times is $c_A^2 = 2.183$, and the lag-1 auto-correlation is 0.159.

The same LBL trace file contains information on the packet sizes as well. Matching the first two moments of the packet sizes we got the following PH distribution:

$$\underline{\sigma} = \begin{bmatrix} 0.19937 & 0.80063 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} -0.0028715 & 0.0028715 \\ 0 & -0.014403 \end{bmatrix}, \tag{339}$$

by which the mean packet size is 138.86 and the squared coefficient of variation is 2.508. The packet size distribution and the speed of the transmission line C_i at node i together determine the service process of the packets as

$$\boldsymbol{S_0}^{(i)} = C_i \cdot \boldsymbol{S},$$
$$\boldsymbol{S_1}^{(i)} = C_i \cdot (-\boldsymbol{S})\mathbb{1} \cdot \underline{\sigma}.$$

The values of C_i are set such that utilization of both queues are equal to the desired value ρ.

All the methods for departure process approximation introduced in Section 8.2.1 are compared with simulation results in Figure 37, which depicts the mean queue length at node 2 as the function of the utilization. In the figure, the "Joint moment based" curve covers all methods based on the lag-1 joint moments, they gave exactly the same results in this particular example. The lag-1 joint moment based procedure has been tested both with matching 4 (marginal- and joint-) moments and with matching 9 moments, the difference between the results were marginal in this queueing network. As expected, the Poisson approximation performed worst, and the lag-1 based methods turned out to be the most accurate. The simple approximation based on the scaling of the service process was surprisingly accurate in this test case, but looking at the low utilization cases (right plot in Figure 37) reveals the superiority of the lag-1 based methods.

In Figure 38, comparing the squared coefficient of variation (SCV) of the number of customers, the lag-1 joint moment based method achieved the highest accuracy again. In this

case, however, increasing the number of joint moments to match had a positive impact on the accuracy, the one matching 9 moments managed to reproduce the simulation results almost perfectly.

Table 7 provides further interesting properties of the algorithms. The second column contains the size of the MAP modeling the departure process. It does not depend on the size of the arrival and service processes in the Poisson and in the lag-1 joint moment based methods. The ETAQA truncation method produces the largest departure process, at the minimal truncation level (at level 2) it needs 18 phases, however, to improve accuracy, the truncation level should be increased. The number of non-Markovian results from the 36 executions (varying the utilization between 0.2 and 0.9) is given in the second column. The ETAQA method did not manage to produce a Markovian representation in any of the cases. The lag-1 based method matching 9 moments occasionally returned a non-Markovian result as well, which were possible to fix with the algorithms described in Section 3.3. The last column indicates the mean relative absolute error compared to the simulation results for the mean number of customers and for the squared coefficient of variation (SCV) of the number of customers at node 2 (separated by a slash character). The most accurate results are given by the ETAQA method with a high truncation level, but it is a huge and non-Markovian representation which is not tractable analytically.

In this scenario the input was taken from real traffic measurements, the packet inter-arrival times were relatively busty (with squared coefficient of variation 2.183) and positively correlated. In the next scenario the input will be a synthetically generated MAP that has the "opposite" behavior: it is more deterministic (the squared coefficient of variation is 0.7278), and negatively correlated (the lag-1 auto-correlation is -0.2119). The corresponding matrix parameters are

$$\boldsymbol{D_0} = \begin{bmatrix} -2 & 2 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & -2.5 & 0 \\ 0 & 0 & 0 & -2.5 \end{bmatrix}, \quad \boldsymbol{D_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 1.7 \\ 2.3 & 0 & 0 & 0.2 \end{bmatrix}.$$

The packet size distribution is the same as before.

The mean and the squared coefficient of variation of the number of customers at node 2 is depicted in Figures 39 and 40. (The Poisson and the scaled service process based approximations are omitted to make it easier to distinguish between the curves). In this scenario the Lag-1 based method results in a non-valid stochastic process in many cases. The performance measures returned by the queueing analysis are invalid if the queue is fed by an invalid stochastic process, which is clearly visible in the figure. This is a clear example demonstrating how important it is to stick with Markovian representations, and how essential the role of Algorithms 4 and 5 is. These algorithms, by fitting an invalid process with a valid one, managed to produce a MAP with only very slightly different statistics, that approximate the queue length behavior of node 2 with a reasonable accuracy.

As visible in Table 8, this high accuracy is achieved by a compact representation, with 2 or 3 states only. Since the moment matching returned invalid processes, the difference between Algorithms 4 and 5 becomes visible, they follow different approaches to approximate the invalid process. In this particular example the accurate fitting of the joint moments turned out to be a slight better strategy, but both perform similarly well.

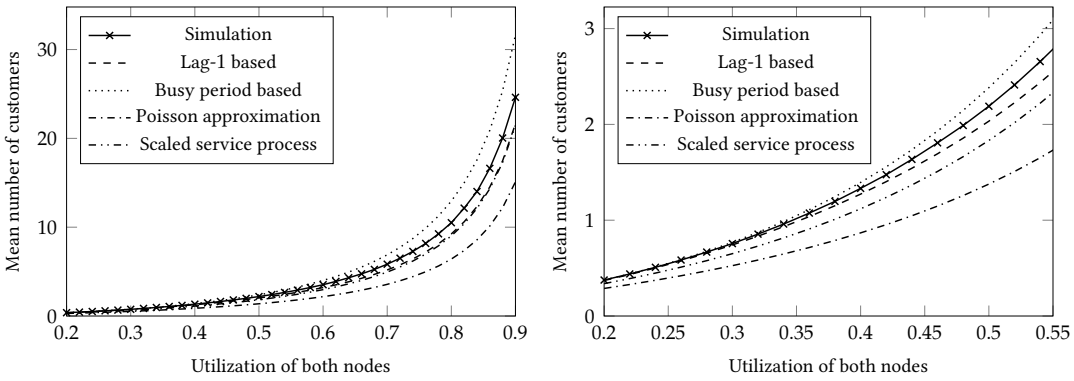


Figure 37.: Mean number of customers at node 2 of the tandem network with LBL input

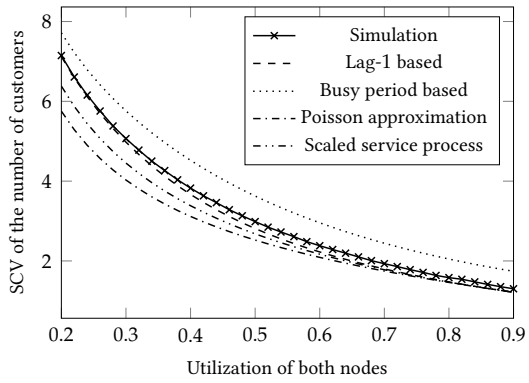


Figure 38.: Squared coefficient of variation of the number of customers at node 2 of the tandem network with LBL input

Approximation algorithm	# of phases	# of non-Mar.	Error
Poisson process	1	0/36	0.3569/0.1396
Scaled service process	2	0/36	0.1477/0.0961
Busy period based	12	0/36	0.119/0.219
ETAQA, truncation level=2	18	36/36	0.1364/0.068
ETAQA, truncation level=50	306	36/36	0.0034/0.007
Lag-1, 4 moments, Algorithm 3	2	0/36	0.1116/0.0695
Lag-1, 4 moments, Algorithm 5	2	0/36	0.1116/0.0695
Lag-1, 4 moments, Algorithm 4	2	0/36	0.1116/0.0695
Lag-1, 9 moments, Algorithm 3	3	12/36	0.076/0.0511
Lag-1, 9 moments, Algorithm 5	3	0/36	0.076/0.0511
Lag-1, 9 moments, Algorithm 4	3	0/36	0.076/0.0511

Table 7.: Properties of the departure process approximations for the tandem example with LBL input

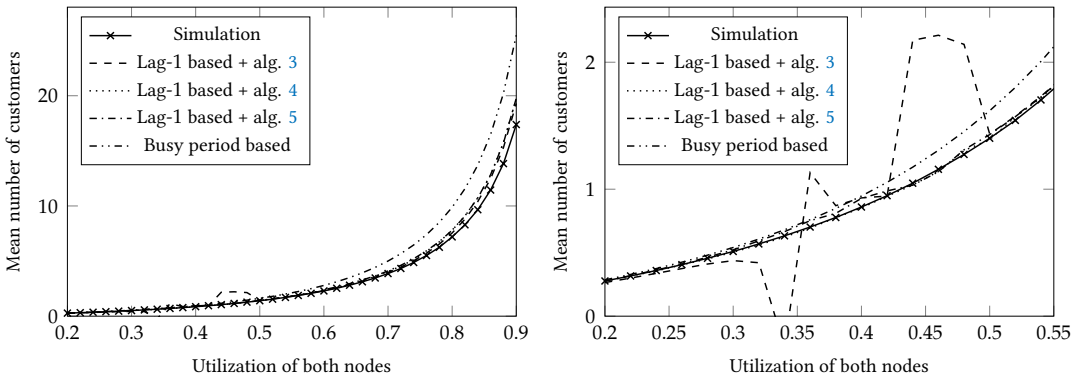


Figure 39.: Mean number of customers at node 2 of the tandem network with negatively correlated input

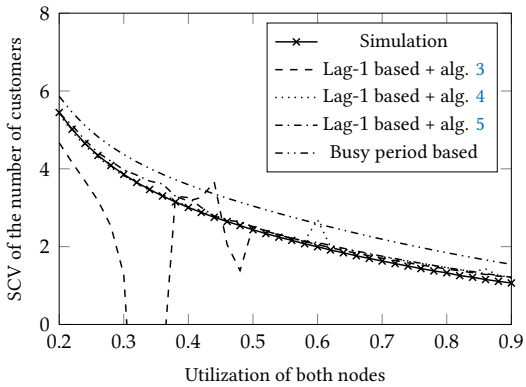


Figure 40.: Squared coefficient of variation of the number of customers at node 2 of the tandem network with negatively correlated input

Approximation algorithm	# of phases	# of non-Mar.	Error
Busy period based	14	0/36	0.2654/0.2686
ETAQA, truncation level=2	24	36/36	0.0567/0.0303
ETAQA, truncation level=50	408	36/36	0.0011/0.002
Lag-1, 4 moments, Algorithm 3	2	18/36	–/–
Lag-1, 4 moments, Algorithm 5	2	0/36	0.0413/0.0566
Lag-1, 4 moments, Algorithm 4	2	0/36	0.0411/0.052
Lag-1, 9 moments, Algorithm 3	3	36/36	–/–
Lag-1, 9 moments, Algorithm 5	3	0/36	0.0513/0.0615
Lag-1, 9 moments, Algorithm 4	3	0/36	0.037/0.0596

Table 8.: Properties of the departure process approximations for the tandem example with negatively correlated input

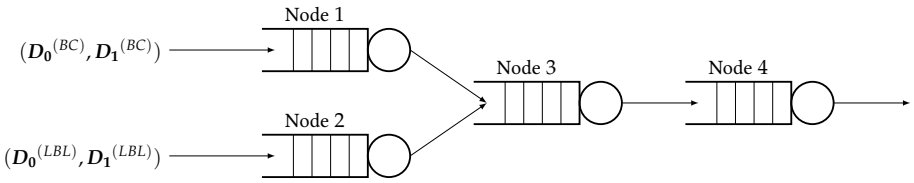


Figure 41.: A more complex queueing network with 4 nodes

8.2.3 *A more complex numerical example*

The next example considers a more complex queueing network with four nodes and superposition (see Figure 41).

The traffic parameters are taken from real measurements. The input of node 1 is the MAP defined by (338) created from the LBL trace by Algorithm 3, while the traffic entering node 2 is a three state MAP obtained by Algorithm 5 from the BC trace. The matrices characterizing this MAP are

$$\boldsymbol{D_0}^{(BC)} = \begin{bmatrix} -16.5173 & 0 & 0 \\ 0 & -48.1036 & 0 \\ 0 & 0 & -394.677 \end{bmatrix}, \boldsymbol{D_1}^{(BC)} = \begin{bmatrix} 0.1359 & 1.4837 & 14.8977 \\ 0.0006 & 35.837 & 12.266 \\ 0.0408 & 3.4362 & 391.2 \end{bmatrix}. \tag{340}$$

The packet sizes are represented by the same PH distributions at all nodes, given by parameters (339). The capacities of the links are set to ensure the same utilization at all nodes.

The mean number of customers at node 3 and 4 are depicted in Figure 42. There is only a single curve for all lag-1 based methods (involving 9 moments) because their results are almost identical. At both nodes the Poisson approximation performs worst (as expected), the busy-period based and lag-1 based methods perform best. Similar tendency can be seen in Figure 43 for the squared coefficient of variations. The scaled service process method delivers surprisingly accurate results in this example.

The absolute relative errors in the mean and the squared coefficient of variations of the number of customers are given by Table 9 (for node 3) and Table 10 (for node 4). The size of the traffic representation is also provided in the table, where the limitations of the busy period based method are clearly demonstrated: an additional node in the network would lead to a huge model size making the analysis infeasible numerically.

8.2.4 *Summary of the single-type results*

The presented numerical examples have proven that the lag-1 joint moment based queueing network analysis approach provides reasonable accuracy with a very compact MAP representation of the internal traffic.

We believe that increasing the number of marginal and joint moments involved in the departure process approximation increases the accuracy of the results. Nevertheless, there is a critical step in the algorithm that currently does not allow involving more that 5 (or sometimes 7) marginal moments. This step is the moment-matching method providing matrix $\boldsymbol{D_0}$. At one hand, the solution of the corresponding polynomial system becomes intolerably slow when matching more than 7 moments. At the other hand, the more moments are matched, the more difficult is to find a PH structure that is able to realize those moments.

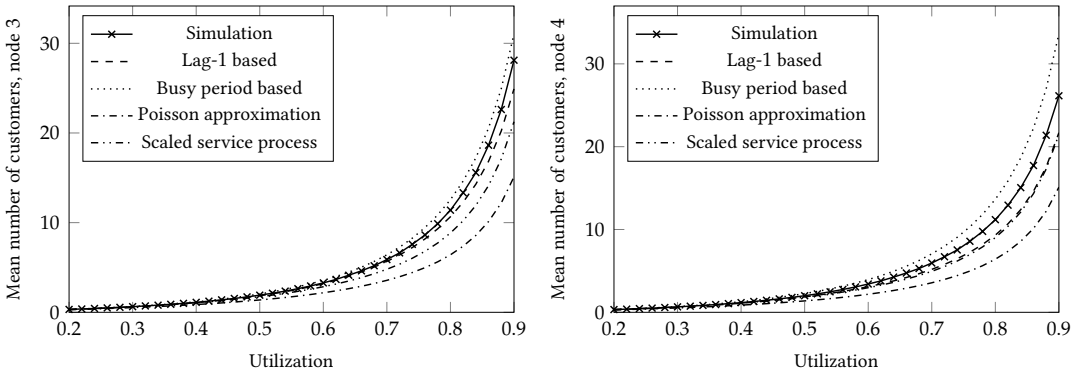


Figure 42.: Mean number of customers at node 3 and node 4 in the more complex single-type example

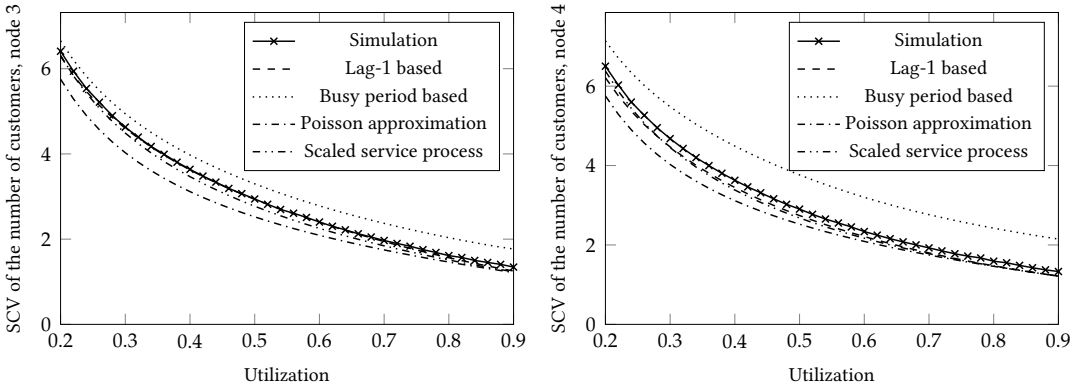


Figure 43.: Squared coefficient of variation of the number of customers at node 3 and 4 of the more complex single-type example

Approximation algorithm	# of phases	# of non-Mar.	Error
Poisson process	1	0/36	0.3029/0.1226
Scaled service process	4	0/36	0.1194/0.1528
Busy period based	144	0/36	0.0699/0.2686
Lag-1, 9 moments, Algorithm 3	9	36/36	0.0461/0.0171
Lag-1, 9 moments, Algorithm 5	9	0/36	0.0495/0.0186
Lag-1, 9 moments, Algorithm 4	9	0/36	0.0301/0.0145

Table 9.: Properties of the departure process approximations forming the input of node 3

Approximation algorithm	# of phases	# of non-Mar.	Error
Poisson process	1	0/36	0.3194/0.1135
Scaled service process	2	0/36	0.1043/0.0675
Busy period based	294	0/36	0.1414/0.3408
Lag-1, 9 moments, Algorithm 3	3	33/36	0.1016/0.0602
Lag-1, 9 moments, Algorithm 5	3	0/36	0.1029/0.0604
Lag-1, 9 moments, Algorithm 4	3	0/36	0.0934/0.0554

Table 10.: Properties of the departure process approximations forming the input of node 4

To overcome these limitations it is necessary to develop new, efficient PH moment fitting algorithms, that, instead of seeking after the exact solution, are able to give up some accuracy when the target moments can not be matched with the given number of phases.

8.3 MULTI-TYPE QUEUEING NETWORKS

Introducing multiple traffic types makes the queueing network analysis much more involved. To the best of our knowledge, the lag-1 joint moment based approach is the only reasonable procedure for the traffic decomposition based analysis.

The truncation methods (including the ETAQA truncation) and the busy period based method introduced in Section 8.2.1 can not be generalized to the multi-type case.

The Poisson and the scaled service process approximations can be used in a multi-type setting, however, these methods do not take the service policy into consideration, i.e., these methods treat the departure process of the multi-type FCFS and priority queue the same.

8.3.1 *Studying the effect of the service discipline*

In this example we consider a two-station tandem network, where the service discipline at the second station is FCFS. Two cases are compared: the case when the first station has a preemptive priority and the case when it has an FCFS server.

The input of the first station is created from the BC trace. Two arrival types are distinguished, arrivals of packets shorter than 256 bytes and arrivals with packet size ≥ 256. From the multi-type lag-1 joint moments extracted from the trace Algorithm 4 produced the following matrices:

$$\boldsymbol{D_0}^{(BC)} = \begin{bmatrix} -16.5173 & 0 & 0 \\ 0 & -48.1036 & 0 \\ 0 & 0 & -394.677 \end{bmatrix},$$
$$\boldsymbol{D_1}^{(BC)} = \begin{bmatrix} 1.3043 & 4.21 & 4.427 \\ 0 & 10.658 & 37.427 \\ 0.0104 & 10.485 & 265.7307 \end{bmatrix}, \tag{341}$$
$$\boldsymbol{D_2}^{(BC)} = \begin{bmatrix} 0 & 0 & 6.576 \\ 0.0028 & 0.016 & 0 \\ 0.0269 & 0 & 118.424 \end{bmatrix}.$$

The packet size distribution has been determined from the trace by moment matching as well. The parameters are

$$\underline{\sigma}_1 = \begin{bmatrix} 0.588 & 0.412 \end{bmatrix}, \qquad \underline{\sigma}_2 = \begin{bmatrix} 0.879 & 0 & 0.121 \end{bmatrix}, \tag{342}$$

$$\boldsymbol{S}_1 = \begin{bmatrix} -0.0086 & 0.0086 \\ 0 & -0.0146 \end{bmatrix}, \qquad \boldsymbol{S}_2 = \begin{bmatrix} -0.00227 & 0.00227 & 0 \\ 0 & -0.00227 & 0.00227 \\ 0 & 0 & -0.00258 \end{bmatrix}, \tag{343}$$

where $(\underline{\sigma}_1, \boldsymbol{S}_1)$ represents the distribution of the small and $(\underline{\sigma}_2, \boldsymbol{S}_2)$ the one of the large packets. The utilization of both queues is set to 0.8.

Approximation algorithm	FCFS		Priority	
	Class 1.	Class 2.	Class 1.	Class 2.
Simulation	5.115/1.612	2.215/1.322	4.023/1.949	2.242/1.36
Poisson process	3.688/1.634	2.056/1.257	3.688/1.634	2.056/1.257
Scaled service process	2.492/1.417	1.58/1.041	2.492/1.417	1.58/1.041
Lag-1, Algorithm 4 (single step)	5.161/1.457	2.47/1.271	4.425/1.624	2.392/1.295
Lag-1, Algorithm 4 (step-by-step)	4.495/1.506	2.088/1.202	4.006/1.761	2.184/1.258

Table 11.: Tandem multi-type network with the service at the first node set to FCFS and Priority

The results are summarized in Table 11. The numbers in the columns are the class 1 and class 2 performance measures of node 2 given that the service policy at node 1 is FCFS and preemptive priority, respectively. The two performance measures separated by a slash (/) character are the mean and the squared coefficient of variation of the queue length.

Algorithm 3 is excluded from the comparison, since it returned an invalid process, making the analysis of node 2 impossible. Algorithm 5 is also excluded, since its attempt to approximate a substantially invalid (negative) joint pdf resulting in a bad approximation for the departure process.

Algorithm 4, aiming to match the joint moments, is, however, always possible to apply. The insensitivity of the Poisson and scaled service algorithms to the service discipline is clearly visible in the table. The lag-1 joint moment based methods have a significant error as well, although they are still the best in this comparison. These methods managed to reflect the effect of the service discipline: class-1 packets have significantly higher, class-2 packet have slightly lower queue lengths in the FCFS case. The tendencies in the squared coefficient of variations are captured correctly as well.

The reason for the relatively inferior performance of the lag-1 moment based methods is that the joint moments turned out to be difficult to approximate. When the first node has a priority scheduler, the exact joint moments of the departure process are

$$\boldsymbol{N}_1 = \begin{bmatrix} 0.70978 & 0.60515 & 1.4979 \\ 0.70454 & 0.73155 & 3.4185 \\ 2.4417 & 4.3264 & 36.899 \end{bmatrix}, \quad \boldsymbol{N}_2 = \begin{bmatrix} 0.29022 & 0.39485 & 1.5011 \\ 0.29546 & 0.45836 & 2.2186 \\ 0.55721 & 0.9696 & 5.5256 \end{bmatrix}. \tag{344}$$

From these joint moments, the step-by-step variant of Algorithm 4 managed to create a MMAP whose joint moments are

$$\hat{\boldsymbol{N}}_1 = \begin{bmatrix} 0.70978 & 0.66435 & 1.5013 \\ 0.66435 & 0.73155 & 2.4865 \\ 1.7544 & 3.5115 & 22.134 \end{bmatrix}, \quad \hat{\boldsymbol{N}}_2 = \begin{bmatrix} 0.29022 & 0.33565 & 1.4977 \\ 0.33565 & 0.45836 & 3.1559 \\ 1.2446 & 2.131 & 20.454 \end{bmatrix}, \tag{345}$$

while the single-step variant of the algorithm created a MMAP with joint moments

$$\check{\boldsymbol{N}}_1 = \begin{bmatrix} 0.70978 & 0.70978 & 2.1286 \\ 0.70978 & 0.70978 & 2.1286 \\ 2.1286 & 2.1286 & 6.3837 \end{bmatrix}, \quad \check{\boldsymbol{N}}_2 = \begin{bmatrix} 0.29022 & 0.29022 & 0.87038 \\ 0.29022 & 0.29022 & 0.87038 \\ 0.87038 & 0.87038 & 2.6103 \end{bmatrix}. \tag{346}$$

Both of them are relatively poor approximations of joint moments (344).

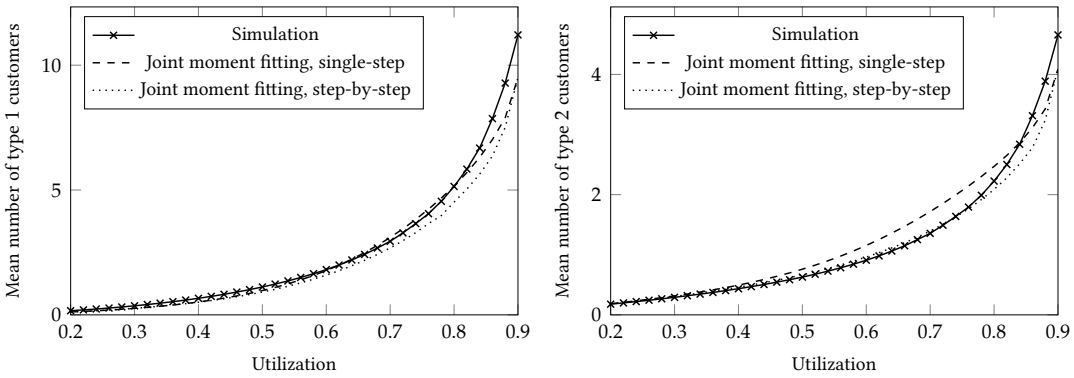


Figure 44.: Mean number of customers at node 2 in the tandem network with two classes and FCFS service

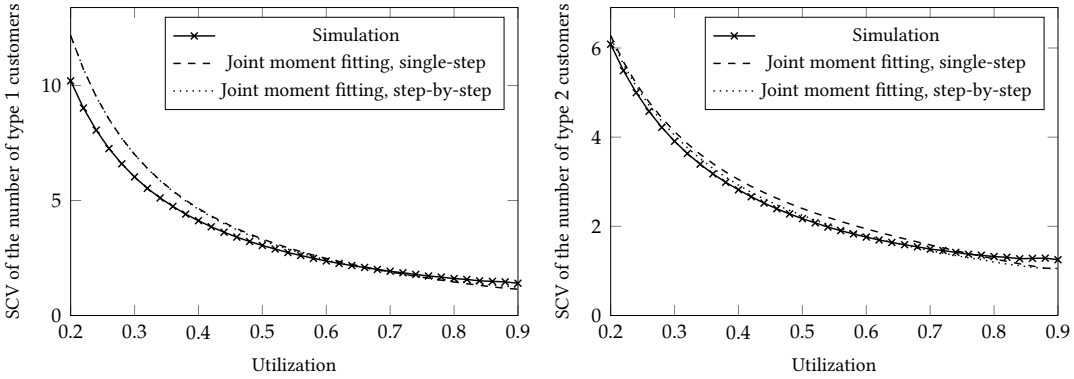


Figure 45.: Squared coefficient of variation of the number of customers at node 2 in the tandem network with two classes and FCFS service

Approximation algorithm	**Error, type** 1	**Error, type** 2
Lag-1, Algorithm 4, single-step	0.1392/0.1063	0.1595/0.078
Lag-1, Algorithm 4, step-by-step	0.1842/0.09	0.0584/0.0494

Table 12.: Errors of the departure process approximations, FCFS case

8.3.2 *A two-node tandem network*

In this section the two-class variant of the tandem network example of Section 8.2.2 is studied. The input traffic of the first node is the two-class MMAP generated from the BC trace, characterized by matrices (341). Two scenarios are considered: in the first one both nodes have an FCFS scheduler, in the second one they both have a priority scheduler.

The Poisson and the scaled service process based approximations are omitted due to their insensitivity to the service discipline. The moment matching algorithm (Algorithm 3) and the joint density minimization based approximation (Algorithm 5) are also omitted; the former one returned invalid process in every case, and the latter one performed bad when attempting to approximate the invalid joint density function.

The plots comparing the mean and the SCV as the function of the utilization corresponding to the FCFS and the priority service are presented by Figures 44, 45, 46 and 47, respectively. Interestingly, the approximation methods performed better in the second scenario, with the

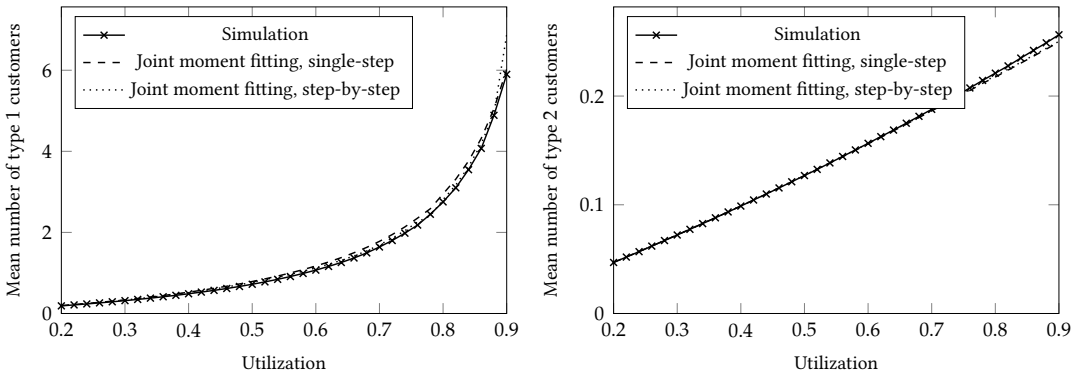


Figure 46.: Mean number of customers at node 2 in the tandem network with two classes and priority service

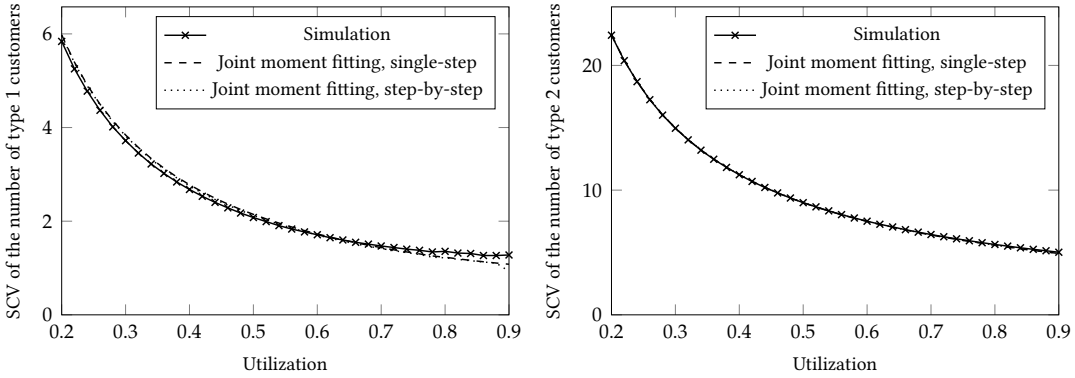


Figure 47.: Squared coefficient of variation of the number of customers at node 2 in the tandem network with two classes and priority service

Approximation algorithm	Error, type 2	Error, type 2
Lag-1, Algorithm 4, single-step	0.0739/0.0438	0.006/0.0049
Lag-1, Algorithm 4, step-by-step	0.0474/0.0733	0.0051/0.0042

Table 13.: Errors of the departure process approximations, priority case

priority server. In the FCFS case the relative absolute error was below 20%, and it was well bellow 10% in the priority case. (The exact values of the average relative errors of the mean and the SCV, separated by a slash, are shown in Tables 12 and 13).

8.3.3 *The more complex example with two customer types*

In the final numerical example the four node network presented in Section 8.2.3 is investigated. The service policy is set to FCFS at nodes 1,2 and 4, and it is set to preemptive priority at node 3. The input of nodes 1 and 2 are the same as in the previous section, defined by (341).

As shown in Figures 48 and 49, the presented lag-1 based departure process approximation methods are able to reproduce the simulation results relatively well. The relative errors, however, are higher in this complex example, they can be as high as 25% according to Table 14.

Observe that the shape of the curves in the plots are not always smooth. The reason is that the moment matching algorithm producing matrix $\boldsymbol{D_0}$ needed sometimes less, sometimes

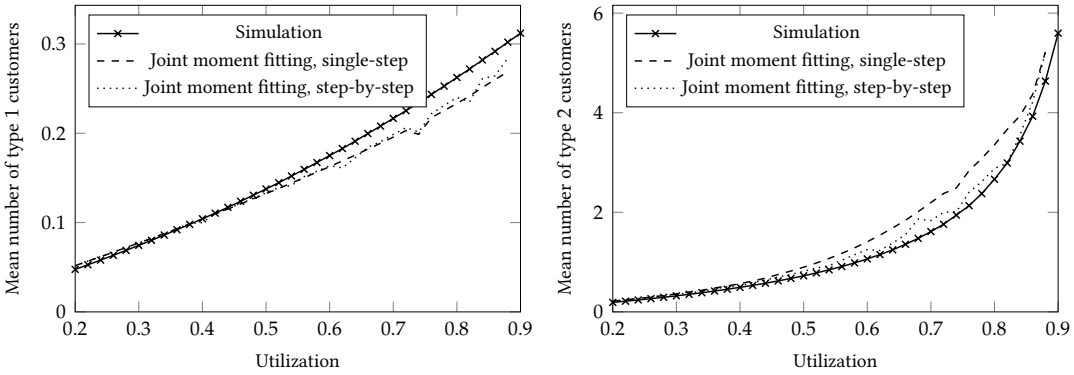


Figure 48.: Mean number of customers at node 3 in the four node network with two classes

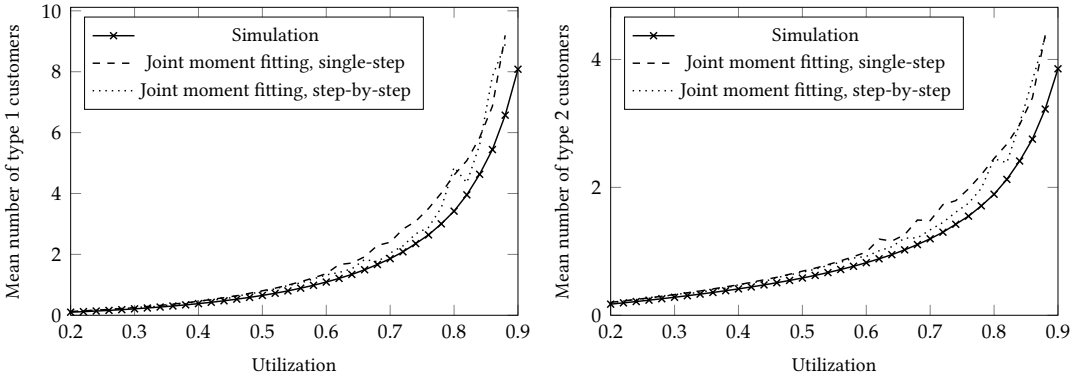


Figure 49.: Mean number of customers at node 4 in the four node network with two classes

more states to realize the target moments. The varying number of states leaded to sharp changes for the constraints while fitting the joint moments, leading to jagged curves in the figure.

Approximation algorithm	Node 3		Node 4	
	type 1	type 2	type 1	type 2
Lag-1, Algorithm 4, single-step	0.0655	0.2227	0.2582	0.2123
Lag-1, Algorithm 4, step-by-step	0.0608	0.1086	0.1811	0.1559

Table 14.: Errors of the departure process approximations in the four node network with two classes

8.3.4 *Summary of the multi-type results*

The performance of the lag-1 based method turned out to be slightly less convincing in case of multi-type queueing networks.

In the single class case, with an N-state approximation of the departure process N^2 moments were matched or approximated to create matrices $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$ consisting of $2N^2$ entries. Hence, with the redundancy factor of 2, there is a significant degree of freedom left to find a valid Markovian representation. In case of two customer types, $2N^2$ moments determine an N-state MMAP, which is characterized by $3N^2$ parameters, hence the redundancy factor is just 1.5, the representation transformation algorithms have much less degree of freedom to

obtain a Markovian solution. The decreasing redundancy factor is one possible reason for the sub-optimal performance observed with more customer types.

It is worth noting, however, that the lag-1 joint moment based approach is still the only possibility to analyze multi-class queuing networks with nodes having MMAP input and PH distributed service times. The alternative procedures developed for single-type queueing networks are either impossible to generalize to the multi-type case or are not able to take the service policy into account.

dc_1412_17

9

CONCLUDING REMARKS AND FUTURE WORK

The dissertation provides an overview on many elements of matrix-analytic methods, and several new results are provided as well.

In the field of traffic characterization, the most valuable contribution might be the canonical form of order-3 PH distributions and the lag-1 joint moments based representation of MMAPs The most interesting direction to continue this line of research in the future is the development of adaptive joint moment matching algorithms that can adjust the size of the representation automatically depending on the target moment set.

The main novelties in the second part are the departure process analysis of three queues. For the three queues the solution did not follow the same methodology, though. The MAP/MAP/1 queue was the first system for which the lag-1 joint moments of the departure process were derived. In case of the multi-class MMAP[K]/PH[K]/1 queue a completely different approach, based on the age process, turned out to be the key to the solution. The main challenge in the departure process of the MMAP[K]/PH[K]/1 priority queue was to make the solution numerically tractable.

Priority queues have been investigated many times in the past. Results exist for the MMAP[K]/G/1 system, which is similar to the one discussed in the dissertation as well, but the procedure presented here is the first one that is robust enough to be used in practical applications with a large number of phases.

The joint moment based queueing network analysis method, which combines all the results of the first two parts of the dissertation has proven to be a viable solution according to our numerical experiments. A possible direction for improvements can be the application of the adaptive moment matching algorithms mentioned above, and to take higher lag joint moments into consideration when characterizing the internal traffic.

In the future we plan to adapt these results to continuous systems as well, where the jobs are not discrete but infinitesimally small, considered as fluid drops. We already have solved and published many elements of the analysis of such fluid queueing systems, but there is still more work to be done, especially in the field of fluid traffic characterization and fitting.

dc_1412_17

10

SUMMARY

ORGANIZATION OF THE THESES

The dissertation consists of three parts building upon each other, hence the theses are grouped into three thesis groups.

In the first thesis group the main tools of the Markovian workload characterization, the phase-type distributions and the Markovian arrival processes are considered. Thesis 1.1 states that all size 3 PH distributions can be transformed to a given canonical form, which can be exploited to make PH fitting methods more efficient. A new moment matching procedure is presented by Thesis 1.2, that can adapt the size of the PH representation to match the target moment set automatically. Important MAP and MMAP characterization results are provided by Thesis 1.3, together with a joint moment matching method for both single and multi-type arrival processes. This thesis is supplemented by three numerical methods to transform the result of the moment matching method to a Markovian representation. These results make it possible to create Markovian models for the network traffic, that can be used both in simulation based and in analytical performance analysis.

The second thesis group is related to the solution of single-class and multi-class queues with correlated arrival processes and Markovian service times. In the multi-class case both the first-come-first-served (FCFS) and the priority service policies are considered. Thesis 2.4 provides the performance analysis of the priority queue with MMAP arrival process and PH distributed service times, based on the workload process. The distribution and the moments of the sojourn times and of the number of customers in the system are derived both for the preemptive resume and the non-preemptive service policy. Theses 2.1, 2.2 and 2.3 provide the characterization of the departure processes of the single class MAP/MAP/1 and the multi-class MMAP[K]/PH[K]/1 FCFS and priority queues, respectively. (For the three queues the solution did not follow the same methodology, though).

Queueing networks are considered in the third thesis group, that consists of a single thesis. In Thesis 3.1 a novel queueing network solution approach is introduced, that integrates the results of the first two thesis groups. In this approach the traffic of the queueing network is characterized by Markovian arrival processes discussed in the first part of the dissertation, and the nodes are the queues discussed in the second part of the dissertation. The Markovian arrival processes representing the internal traffic are obtained by moment matching.

THESIS GROUP 1

Thesis 1.1

I have proven that every order-3 PH distribution can be transformed to one of the following three canonical forms with an appropriate similarity transformation:

$$\boldsymbol{\alpha}^{(1)} = \begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}, \qquad \boldsymbol{\alpha}^{(2)} = \begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}, \qquad \boldsymbol{\alpha}^{(3)} = \begin{bmatrix}\gamma_1 & 0 & \gamma_3\end{bmatrix},$$

$$\boldsymbol{G}^{(1)} = \begin{bmatrix}-x_1 & 0 & 0\\ x_2 & -x_2 & 0\\ 0 & x_3 & -x_3\end{bmatrix}, \boldsymbol{G}^{(2)} = \begin{bmatrix}-x_1 & 0 & x_{13}\\ x_1 & -x_1 & 0\\ 0 & x_3 & -x_3\end{bmatrix}, \boldsymbol{G}^{(3)} = \begin{bmatrix}-x_1 & 0 & x_{13}\\ x_2 & -x_2 & 0\\ 0 & x_3 & -x_3\end{bmatrix}.$$

The results of this thesis have been published in [94] and [95].

Thesis 1.2

I have introduced a special PH structure, called generalized hyper-Erlang distribution, and proposed a flexible moment matching algorithm that adapts the size of the representation automatically according to the moments to match.

The corresponding results have been published in [93].

Thesis 1.3

I have pointed out that an order N non-redundant MMAP is uniquely determined by N^2 independent parameters. I have introduced a moment matching method that creates a MAP based on $2N-1$ marginal moments and $(N-1)^2$ lag-1 joint moments. The results have been generalized to marked MAPs as well: I have shown that an order N non-redundant MMAP with C arrival types is uniquely determined by $C \cdot N^2$ independent parameters. I have developed a moment matching method for MMAPs as well.

The corresponding results have been published in [81] for the single-type case and in [45] for the multi-type case. Further closely related publications are [50], [91] and [96].

THESIS GROUP 2

Thesis 2.1

I have derived the lag-1 joint moments of the departure process of the MAP/MAP/1 queue.

The corresponding results have been published in [92].

Thesis 2.2

I have derived the multi-class lag-1 joint moments of the departure process of the two-class MAP/MAP/1 priority queue.

The corresponding results have been published in [45] and in [48].

Thesis 2.3

I have provided the detailed departure process analysis of the multi-class MMAP[K]/PH[K]/1-FCFS queue. The analysis follows an entirely new approach: it is based on the age process instead of the queue length process.

The corresponding results have been published in [97].

Thesis 2.4

I have developed an analysis method for the MMAP[K]/PH[K]/1 priority queue, both for the preemptive resume and the non-preemptive scheduling policy. Efficient numerical procedures are provided to obtain the distribution function, its Laplace-Stieltjes transform and the moments for the both the sojourn times and the number of customers in the system.

The corresponding results have been published in [49].

THESIS GROUP 3

Thesis 3.1

I have introduced a lag-1 joint moment based method for the analysis of multi-class open queueing networks.

The corresponding results have been published in [92] and in [45].

dc_1412_17

Part IV

APPENDIX

dc_1412_17

A

FUNDAMENTAL RELATIONS

A.1 KRONECKER OPERATIONS

The *Kronecker product* of matrix $\boldsymbol{A}$ of size $N_A \times M_A$ and matrix $\boldsymbol{B}$ of size $N_B \times M_B$ is defined by

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \dots & a_{1,M_A}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \dots & a_{2,M_A}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_A,1}\boldsymbol{B} & a_{N_A,2}\boldsymbol{B} & \dots & a_{N_A,M_A}\boldsymbol{B} \end{bmatrix}, \tag{347}$$

where $a_{i,j}$ is the $i,j$the entry of matrix $\boldsymbol{A}$. The size of the Kronecker product is $N_A N_B \times M_A M_B$. For square matrices the definition of the *Kronecker sum* of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is

$$\boldsymbol{A} \oplus \boldsymbol{B} = \boldsymbol{A} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{B}. \tag{348}$$

The Kronecker operations are useful to express the joint generator matrix of independent Markov chains in a compact way.

If there are two DTMCs with generators $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ of size N_1 and N_2, then the generator of their joint behavior is given by $\boldsymbol{P} = \boldsymbol{P_1} \otimes \boldsymbol{P_2}$. If (i,j) identifies the state where the first and the second DTMCs are in state i and j, respectively, then the states in the Kronecker multiplied generator are ordered as $(1,1),\dots,(1,N_2),(2,1),\dots,(2,N_2),\dots(N_1,N_2)$. Figure 50 presents an example where a two-state and a three-state DTMC is superposed.

Similarly, in case of two CTMC generators $\boldsymbol{Q_1}$ and $\boldsymbol{Q_2}$ the generator of the joint process is obtained by the Kronecker summation $\boldsymbol{Q} = \boldsymbol{Q_1} \oplus \boldsymbol{Q_2}$ (see Figure 51 for an example).

Some identities related to Kronecker operations which are used many times in the dissertation are ([77]):

$$\boldsymbol{AC} \otimes \boldsymbol{BD} = (\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}), \tag{349}$$

$$(c\boldsymbol{A}) \otimes \boldsymbol{B} = \boldsymbol{A} \otimes (c\boldsymbol{B}) = c(\boldsymbol{A} \otimes \boldsymbol{B}), \tag{350}$$

$$(\boldsymbol{A} + \boldsymbol{B}) \otimes \boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{C} + \boldsymbol{B} \otimes \boldsymbol{C}, \tag{351}$$

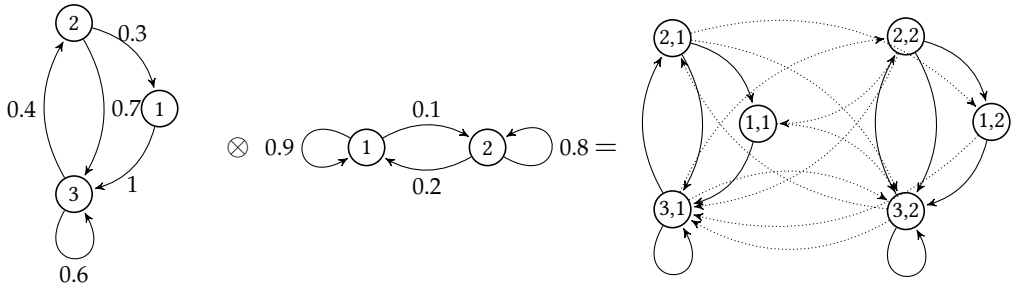
$$e^{(\boldsymbol{A}\oplus\boldsymbol{B})x} = e^{(\boldsymbol{A}\otimes\boldsymbol{I})x} e^{(\boldsymbol{I}\otimes\boldsymbol{B})x} = e^{\boldsymbol{A}x} \otimes e^{\boldsymbol{B}x}, \tag{352}$$

An other useful operator on matrices that is closely related to Kronecker operations is the column stacking $\mathrm{vec}\langle\rangle$ operator, which copies the columns of a martrix below each other. Assuming compatible matrices, some identities of the $\mathrm{vec}\langle\rangle$ operator are:

$$\mathrm{vec}\langle \boldsymbol{AXB} \rangle = (\boldsymbol{B}^T \otimes \boldsymbol{A})\mathrm{vec}\langle \boldsymbol{X} \rangle, \tag{353}$$

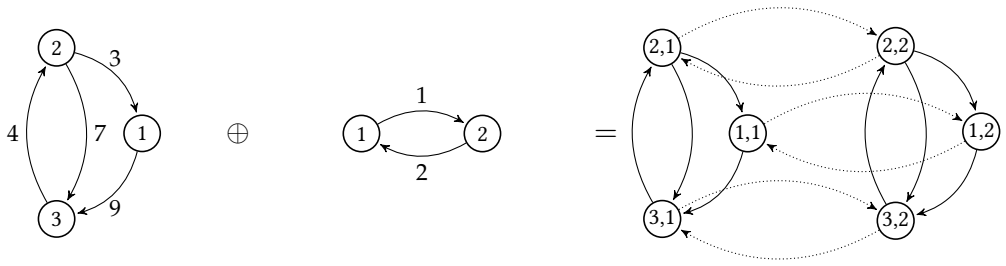
$$\mathrm{vec}\langle \underline{u}^T \underline{v} \rangle = (\underline{v}^T \otimes \underline{u}^T), \tag{354}$$

where $\underline{u}$ and $\underline{v}$ are row vectors (see [77]).



$$\begin{bmatrix} 0 & 0 & 1 \\ 0.3 & 0 & 0.7 \\ 0 & 0.4 & 0.6 \end{bmatrix} \otimes \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0 & 0.2 & 0.8 \\ 0.27 & 0.03 & 0 & 0 & 0.63 & 0.07 \\ 0.06 & 0.24 & 0 & 0 & 0.14 & 0.56 \\ 0 & 0 & 0.36 & 0.04 & 0.54 & 0.06 \\ 0 & 0 & 0.08 & 0.32 & 0.12 & 0.48 \end{bmatrix}$$

Figure 50.: Example for the Kronecker product of two DTMCs



$$\begin{bmatrix} -9 & 0 & 9 \\ 3 & -10 & 7 \\ 0 & 4 & -4 \end{bmatrix} \oplus \begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} -10 & 1 & 0 & 0 & 9 & 0 \\ 2 & -11 & 0 & 0 & 0 & 9 \\ 3 & 0 & -11 & 1 & 7 & 0 \\ 0 & 3 & 2 & -12 & 0 & 7 \\ 0 & 0 & 4 & 0 & -5 & 1 \\ 0 & 0 & 0 & 4 & 2 & -6 \end{bmatrix}$$

Figure 51.: Example for the Kronecker summation of two CTMCs

A.2 PROPERTIES OF THE MATRIX-EXPONENTIAL FUNCTION

Working with matrix-exponential functions has the benefit that several related integral quantities have an efficient or even an explicit solution.

Theorem 36. *([58, Theorem 13.19]) For compatible matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ the integral*

$$\boldsymbol{X} = \int_0^\infty e^{\boldsymbol{A}x} \boldsymbol{C} e^{\boldsymbol{B}x}\, dx \tag{355}$$

satisfies the Sylvester equation

$$\boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{B} + \boldsymbol{C} = \boldsymbol{0}. \tag{356}$$

This Sylvester equation has a unique solution if the eigenvalues of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have real parts in the open left half-plane.

There are two existing approaches to solve Sylvester equations of form (356).

The first approach relies on Kronecker operations. Applying the $\mathrm{vec}\langle\rangle$ operation on both sides of (356) and making use of the identity (353) gives

$$(\boldsymbol{I} \otimes \boldsymbol{A})\mathrm{vec}\langle \boldsymbol{X}\rangle + (\boldsymbol{B}^T \otimes \boldsymbol{I})\mathrm{vec}\langle \boldsymbol{X}\rangle + \mathrm{vec}\langle \boldsymbol{C}\rangle = \underline{0}, \tag{357}$$

from which the elements of matrix $\boldsymbol{X}$ can be explicitly obtained by

$$\mathrm{vec}\langle \boldsymbol{X}\rangle = \left(-\boldsymbol{B}^T \oplus \boldsymbol{A}\right)^{-1} \mathrm{vec}\langle \boldsymbol{C}\rangle. \tag{358}$$

The second approach to solve (356) is the direct numerical solution, which is much more efficient both in computational and memory complexity, since these direct methods operate on smaller matrices and avoid Kronecker operations. One of the fastest and most widely used direct solution method for Sylvester equations is the Hessenberg-Schur method [33].

Convolution integrals involving matrix-exponentials have explicit solution as well. The next theorem provides the basis of the solution.

Theorem 37 (Theorem 1 in [84])**.** *If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices with*

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}, \tag{359}$$

then

$$e^{\boldsymbol{X}t} = \begin{bmatrix} e^{\boldsymbol{A}t} & \int_{a=0}^{t} e^{\boldsymbol{A}a} \boldsymbol{C} e^{\boldsymbol{B}(t-a)} da \\ 0 & e^{\boldsymbol{B}t} \end{bmatrix}. \tag{360}$$

Hence, the convolution of two matrix-exponentials can be expressed explicitly as a single matrix exponential with larger size, as given by the following corollary.

Corollary 21. *The solution of the convolution integral $\int_{a=0}^{t} e^{\boldsymbol{A}a} \boldsymbol{C} e^{\boldsymbol{B}(t-a)} da$ is*

$$\int_{a=0}^{t} e^{\boldsymbol{A}a} \boldsymbol{C} e^{\boldsymbol{B}(t-a)} da = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} e^{\boldsymbol{X}t} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}, \tag{361}$$

where matrix $\boldsymbol{X}$ is defined by (359).

dc_1412_17

B

PROOFS OF THEOREMS

B.1 PROOF OF LEMMA 1

According to Theorem 1 the transformation matrix satisfies the linear equations $\boldsymbol{BG} = \boldsymbol{SB}$, $\boldsymbol{B}\mathbb{1} = \mathbb{1}$. Furthermore, a property of the similarity transformation is that the eigenvalues hence the characteristic polynomials of $\boldsymbol{S}$ and $\boldsymbol{G}$ are the same.

First we show that the columns of $\boldsymbol{B}$, defined by (23), satisfy the necessary linear equations. The product $\boldsymbol{BG}$ can be expressed by

$$\begin{aligned}\boldsymbol{BG} = &\frac{1}{x_{13}-x_1}\boldsymbol{S}\mathbb{1}\begin{bmatrix}-x_1 & 0 & x_{13}\end{bmatrix} + \frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\begin{bmatrix}x_2 & -x_2 & 0\end{bmatrix}\\ &+\frac{1}{(x_{13}-x_1)x_2x_3}(x_2\boldsymbol{I}+\boldsymbol{S})(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\begin{bmatrix}0 & x_3 & -x_3\end{bmatrix},\end{aligned}$$

whose first column is

$$\frac{1}{x_{13}-x_1}\boldsymbol{S}\mathbb{1}\cdot(-x_1) + \frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\cdot x_2 = \frac{1}{x_{13}-x_1}\boldsymbol{S}^2\mathbb{1} = \boldsymbol{S}\underline{b}_1,$$

the second column is

$$\begin{aligned}&\frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\cdot(-x_2) + \frac{1}{(x_{13}-x_1)x_2x_3}(x_2\boldsymbol{I}+\boldsymbol{S})(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\cdot x_3\\ &= \frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}^2\mathbb{1} = \boldsymbol{S}\underline{b}_2,\end{aligned}$$

and the third column is

$$\begin{aligned}&\frac{1}{x_{13}-x_1}\boldsymbol{S}\mathbb{1}\cdot x_{13} + \frac{1}{(x_{13}-x_1)x_2x_3}(x_2\boldsymbol{I}+\boldsymbol{S})(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}\mathbb{1}\cdot(-x_3)\\ &= \boldsymbol{S}\mathbb{1} - \frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}^2\mathbb{1} - \frac{1}{(x_{13}-x_1)x_2}(x_1\boldsymbol{I}+\boldsymbol{S})\boldsymbol{S}^2\mathbb{1} = \boldsymbol{S}(\mathbb{1}-\underline{b}_1-\underline{b}_2).\end{aligned}$$

Finally, we prove that $\underline{b}_1+\underline{b}_2+\underline{b}_3 = \mathbb{1}$. The sum of these vectors is

$$\begin{aligned}&\frac{1}{(x_{13}-x_1)x_2x_3}\boldsymbol{S}(x_2x_3 + x_3(x_1\boldsymbol{I}+\boldsymbol{S}) + (x_2\boldsymbol{I}+\boldsymbol{S})(x_1\boldsymbol{I}+\boldsymbol{S}))\mathbb{1}\\ &= \frac{1}{(x_{13}-x_1)x_2x_3}\boldsymbol{S}(x_1x_2+x_1x_3+x_2x_3+(x_1+x_2+x_3)\boldsymbol{S}+\boldsymbol{S}^2)\mathbb{1}.\end{aligned} \tag{362}$$

However, exploiting the fact that the resulting generator $\boldsymbol{G}$ has the same characteristic polynomial as the original $\boldsymbol{S}$, the parameters (24) are obtained from the solution of the equations

$$a_0 = (x_1-x_{13})x_2x_3,\ a_1 = x_1x_2+x_2x_3+x_3x_1,\ a_2 = x_1+x_2+x_3. \tag{363}$$

With these parameters (362) can be rewritten as $\frac{1}{-a_0}\boldsymbol{S}(a_1+a_2\boldsymbol{S}^2)\mathbb{1}$ that equals $\mathbb{1}$ since $a_2\boldsymbol{S}^2 + a_1\boldsymbol{S} + a_0 = \boldsymbol{0}$ holds due to the Cayley–Hamilton theorem.

B.2 PROOF OF THEOREM 7

Due to Theorem 6 and $\boldsymbol{B}\mathbb{1} = \mathbb{1}$, if $(\underline{\sigma}, \boldsymbol{S})$ has a Markovian representation, then $\boldsymbol{B}^{-1}\boldsymbol{S}\boldsymbol{B}$ is Markovian, and $x_1 - x_{13}$, x_2, x_3 are positive, when x_1 is in the $[\vartheta_\ell, \vartheta_u]$ interval. Thus, it is enough to prove that vector $\begin{bmatrix}\gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}$, defined in (28)-(30), is non-negative when x_1 takes is value according to (31).

$\gamma_1 = \frac{d_1}{x_1 - x_{13}} \geq 0$ follows immediately from $d_1 = f(0) = -\underline{\sigma}\boldsymbol{S}\mathbb{1} >= 0$, since if $(\underline{\sigma}, \boldsymbol{S})$ has a Markovian representation, then its density is non-negative at zero.

When $\underline{\sigma}\boldsymbol{S}\mathbb{1} = 0$, $\gamma_2 = \frac{x_1 d_1 + d_2}{(x_1 - x_{13})x_2}$ must be non-negative according to property P3.

When $\underline{\sigma}\boldsymbol{S}\mathbb{1} < 0$, we can re-write (29) as:

$$\gamma_2 = \frac{-\underline{\sigma}\boldsymbol{S}\mathbb{1}}{(x_1 - x_{13})x_2}(x_1 - \vartheta_2). \tag{364}$$

The first term of (364) is positive and the second term is non-negative when $x_1 = \max\{\vartheta_2, \vartheta_\ell\}$ according to (31).

For the analysis of γ_3 we re-write (30) as

$$\gamma_3 = \frac{1}{(x_1 - x_{13})x_2 x_3} \underbrace{(x_1 x_2 d_1 + (x_1 + x_2)d_2 + d_3)}_{g(x_1)} \tag{365}$$

The first term is positive again, thus it remains to prove that $g(x_1) \geq 0$ if x_1 is according to (31). The first derivative of $g(x_1)$ has at most two roots:

$$\frac{d}{dx_1} g(x_1) = 0 \quad \Leftrightarrow \quad x_1 = \frac{a_2 \pm \sqrt{a_2^2 - 3a_1}}{3}. \tag{366}$$

If $\sqrt{a_2^2 - 3a_1} = 0$ then $\vartheta_u = \vartheta_\ell = \vartheta_0$ and $x_1 = \vartheta_\ell$ is the only valid value according to Theorem 6.

If $\sqrt{a_2^2 - 3a_1} > 0$ then the larger root of (366) equals ϑ_0, hence $g(x_1)$ is a monotone function when $x_1 > \vartheta_0$. In the $x_1 > \vartheta_0$ region the increasing/decreasing behaviour of $g(x_1)$ is determined by the sign of the second derivative at $x_1 = \vartheta_0$:

$$\frac{d^2}{dx_1^2} g(x_1)|_{x_1 = \vartheta_0} = \frac{-2(a_2 d_1 + 4d_1\sqrt{a_2^2 - 3a_1} + 3d_2)}{3\sqrt{a_2^2 - 3a_1}} \tag{367}$$

When $d_1 = -\underline{\sigma}\boldsymbol{S}\mathbb{1} = 0$, then (367) is non-positive because the numerator is non-positive due to property P3 and the denominator is positive. In this case we have 2 subcases. If $d_2 = 0$, then $g(x_1)$ is constant and x_1 does not effect the sign of γ_3, when $\vartheta_\ell \leq x_1 \leq \vartheta_u$. If $d_2 > 0$, then $g(x_1)$ is monotone decreasing and the minimal x_1 value of the valid range ($\vartheta_\ell \leq x_1 \leq \vartheta_u$ and $\vartheta_2 \leq x_1$) ensures the non-negativity of γ_3 (assuming that a Markovian representation exists).

When $d_1 = -\underline{\sigma}\boldsymbol{S}\mathbb{1} > 0$ we have

$$\begin{aligned}\frac{d^2}{dx_1^2}\, g(x_1)|_{x_1=\vartheta_0} &= \frac{-2\,d_1\,(a_2 + 4\sqrt{a_2^2 - 3a_1} - 3\vartheta_2)}{3\sqrt{a_2^2 - 3a_1}} \\ &= -\underbrace{\frac{2\,d_1}{3\sqrt{a_2^2 - 3a_1}}}_{>0}\left[3\underbrace{(\vartheta_u - \vartheta_2)}_{\geq 0} + \underbrace{(3\vartheta_u - a_2)}_{>0}\right] \leq 0,\end{aligned} \tag{368}$$

where the positivity of the under-braced terms follows from $\sqrt{a_2^2 - 3a_1} > 0$, and the non-negativity of the second term must hold since $(\underline{\sigma}, \boldsymbol{S})$ has a Markovian representation (according to the condition of the theorem) and according to Theorem 6 it must have a unicyclic representation ($x_1 \leq \vartheta_u$) with a non-negative γ_2 ($x_1 \geq \vartheta_2$).

If the second derivative in (368) is negative then $g(x_1)$ is monotone decreasing at $x_1 > \vartheta_0$ and the minimal x_1 value of the valid range ($\vartheta_\ell \leq x_1 \leq \vartheta_u$ and $\vartheta_2 \leq x_1$) ensures the non-negativity of γ_3 (assuming that a Markovian representation exists).

If the second derivative in (368) equals zero (i.e., $\vartheta_u = \vartheta_2$) it means that there is only a single x_1 value, $x_1 = \vartheta_u = \vartheta_2$, which results in a Markovian representation, because for $x_1 > \vartheta_u$ matrix $\boldsymbol{G}$ is non-Markovian and for $x_1 < \vartheta_2$ vector γ is not a probability vector.

When $\sqrt{a_2^2 - 3a_1} > 0$, the possible behaviors of $g(x_1)$ and the associated choices of x_1 are summarized in the following table.

Cases	$g(x_1)$ at $x_1 > \vartheta_0$	constraint of x_1	choice of x_1
$d_1 = 0, d_2 > 0$	mon. decreasing		minimal value
$d_1 = 0, d_2 = 0$	constant		minimal value
$d_1 > 0, \vartheta_u > \vartheta_2$	mon. decreasing		minimal value
$d_1 > 0, \vartheta_u = \vartheta_2$		$x_1 = \vartheta_u = \vartheta_2$	constraint

That is, (31) sets x_1 such that the obtained representation is Markovian when a Markovian representation exists.

dc_1412_17

BIBLIOGRAPHY

[1] Soohan Ahn and Vaidyanathan Ramaswami. Efficient algorithms for transient analysis of stochastic fluid flow models. *Journal of Applied Probability*, pages 531–549, 2005.

[2] Nail Akar and Khosrow Sohraby. An invariant subspace approach in M/G/l and G/M/l type Markov chains. *Stochastic Models*, 13(3):381–416, 1997.

[3] Attahiru Sule Alfa, Bin Liu, and Qi-Ming He. Discrete-time analysis of MAP/PH/1 multiclass general preemptive priority queue. *Naval Research Logistics (NRL)*, 50(6):662–682, 2003.

[4] Martin Anokye, AR Abdul-Aziz, Kwame Annin, and Francis T Oduro. Application of queuing theory to vehicular traffic at signalized intersection in Kumasi-Ashanti region, Ghana. *American International Journal of Contemporary Research*, 3(7):23–29, 2013.

[5] S. Asmussen and M. Bladt. Point processes with finite-dimensional conditional probabilities. *Stochastic Processes and their Application*, 82:127–142, 1999.

[6] Søren Asmussen, Olle Nerman, and Marita Olsson. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, pages 419–441, 1996.

[7] Yonathan Bard. Some extensions to multiclass queueing network analysis. In *Proc. of the Third Int. Symposium on Modelling and Performance Evaluation of Computer Systems*, pages 51–62, Amsterdam, The Netherlands, 1979. North-Holland.

[8] Falko Bause. Doubly stochastic and circulant structured Markovian arrival processes. Technical Report Technical Reports in Computer Science, No. 824, TU Dortmund, Department of Computer Science, 2009.

[9] N. G. Bean and B. F. Nielsen. Quasi-birth-and-death processes with rational arrival process components. *Stochastic Models*, 26(3):309–334, 2010.

[10] N.G. Bean, D.A. Green, and P.G. Taylor. Approximations to the output process of MAP/PH/1 queues. In *2nd International Conference on Matrix Analytic Methods*, pages 151–169. Notable Publications Inc., NJ, 1998.

[11] D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, 2011. 2nd edition.

[12] Dario Bini and Beatrice Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In *Computations with Markov chains*, pages 21–38. Springer, 1995.

[13] A. Bobbio, A. Horváth, and M. Telek. Matching three moments with minimal acyclic phase type distributions. *Stochastic Models*, 21(2-3):303–326, 2005.

[14] A. Bobbio and M. Telek. A benchmark for PH estimation algorithms: results for Acyclic-PH. *Stochastic Models*, 10(3):661–677, 1994.

[15] G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi. *Queueing Networks and Markov Chains*. Wiley, 2006.

[16] Peter Buchholz. An EM-algorithm for MAP fitting from real traffic data. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 218–236. Springer, 2003.

[17] Peter Buchholz, Iryna Felko, and Jan Kriege. Transformation of acyclic phase type distributions for correlation fitting. In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pages 96–111. Springer, 2013.

[18] Peter Buchholz, Peter Kemper, and Jan Kriege. Multi-class Markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67(11):1092–1106, 2010.

[19] Peter Buchholz and Jan Kriege. A heuristic approach for fitting MAPs to moments and joint moments. In *Quantitative Evaluation of Systems, 2009. QEST'09. Sixth International Conference on the*, pages 53–62. IEEE, 2009.

[20] Peter Buchholz and Miklós Telek. Rational processes related to communicating Markov processes. *Journal of Applied Probability*, 49(1):40–59, 2012.

[21] Peter Buchholz and Miklós Telek. On minimal representation of rational arrival processes. *Annals of Operations Research*, 202(1):35–58, 2013.

[22] Giuliano Casale and Peter Harrison. A class of tractable models for run-time performance evaluation. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, pages 63–74. ACM, 2012.

[23] Giuliano Casale, Eddy Z Zhang, and Evgenia Smirni. Trace data characterization and fitting for Markov modeling. *Performance Evaluation*, 67(2):61–79, 2010.

[24] Srinivas R Chakravarthy and Alexander N Dudin. A queueing model for crowdsourcing. *Journal of the Operational Research Society*, pages 1–16, 2015.

[25] D. R. Cox. A use of complex probabilities in the theory of stochastic processes. *Proc. Cambridge Phil. Soc.*, 51:313–319, 1955.

[26] A. Cumani. On the Canonical Representation of Homogeneous Markov Processes Modelling Failure-time Distributions. *Microelectronics and Reliability*, 22:583–602, 1982.

[27] Feng Ding and Tongwen Chen. On iterative solutions of general coupled matrix equations. *SIAM J. Control Optim.*, 44:2269–2284, January 2006.

[28] Tessa Dzial, Lothar Breuer, Ana da Silva Soares, Guy Latouche, and Marie-Ange Remiche. Fluid queues to solve jump processes. *Performance Evaluation*, 62(1):132–146, 2005.

[29] Adesoji Oladapo Farayibi. Investigating the application of queue theory in the Nigerian banking system. 2016.

[30] Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance evaluation*, 31(3-4):245–279, 1998.

[31] Samuel Fomundam and Jeffrey W Herrmann. A survey of queuing theory applications in healthcare. Technical report, 2007.

[32] E. Gelenbe and G. Pujolle. *Introduction to queueing networks*. Wiley, 1987.

[33] Gene Golub, Stephen Nash, and Charles Van Loan. A Hessenberg-Schur method for the problem AX+XB=C. *Automatic Control, IEEE Transactions on*, 24(6):909–913, 1979.

[34] Manish K Govil and Michael C Fu. Queueing theory in manufacturing: A survey. *Journal of manufacturing systems*, 18(3):214, 1999.

[35] Stephen C Graves. The application of queueing theory to continuous perishable inventory systems. *Management Science*, 28(4):400–406, 1982.

[36] C.-H. Guo, B. Iannazzo, and B. Meini. On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.*, 29:1083–1100, 2007.

[37] RJ Hanson. On the constrained linear least-squares problem: a personal view. *Applied numerical mathematics*, 3(5):443–452, 1987.

[38] Mor Harchol-Balter. Real-world workloads: High variability and heavy tails. In *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*, pages 347–348. Cambridge University Press, 2012.

[39] Qi-Ming He and Marcel F Neuts. Markov chains with marked transitions. *Stochastic Processes and their Applications*, 74(1):37–52, 1998.

[40] Qi-Ming He and Hanqin Zhang. A note on unicyclic representation of PH-distributions. *Stochastic Models*, 21:465–483, 2005.

[41] Qiming He. Analysis of a continuous time SM[K]/PH[K]/1/FCFS queue: Age process, sojourn times, and queue lengths. *Journal of Systems Science and Complexity*, 25(1):133–155, 2012.

[42] A. Heindl. *Traffic based decomposition of general queueing networks with correlated input processes*. Shaker Verlag, 2001.

[43] A. Heindl and M. Telek. Output models of MAP/PH/1(/K) queues for an efficient network decomposition. *Performance Evaluation*, 49(1-4):321–339, 2002.

[44] A. Heindl, Q. Zhang, and E. Smirni. ETAQA truncation models for the MAP/MAP/1 departure process. In *QEST '04: Proceedings of the The Quantitative Evaluation of Systems, First International Conference on (QEST'04)*, pages 100–109, Washington, DC, USA, 2004. IEEE Computer Society.

[45] András Horváth, Gábor Horváth, and Miklós Telek. A traffic based decomposition of two-class queueing networks with priority service. *Computer Networks*, 53(8):1235–1248, 2009.

[46] András Horváth and Miklós Telek. Phfit: A general phase-type fitting tool. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 82–91. Springer, 2002.

[47] G. Horváth, B. Van Houdt, and M. Telek. Commuting matrices in MAP/MAP/1 queues. Technical report, 2011.

[48] Gábor Horváth. Efficient analysis of the queue length moments of the MMAP/MAP/1 preemptive priority queue. *Performance Evaluation*, 69(12):684–700, 2012.

[49] Gábor Horváth. Efficient analysis of the MMAP[K]/PH[K]/1 priority queue. *European Journal of Operational Research*, 246(1):128–139, 2015.

[50] Gábor Horváth, Peter Buchholz, and Miklós Telek. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag correlation. In *Second International Conference on the Quantitative Evaluation of Systems (QEST'05)*, pages 124–133. IEEE, 2005.

[51] M.A. Johnson and M.R. Taaffe. Matching moments to phase distributions: Mixtures of Erlang distributions of common order. *Stochastic Models*, 5(4):711–743, 1989.

[52] Edward H Kaplan. A public housing queue with reneging. *Decision Sciences*, 19(2):383–391, 1988.

[53] Leonard Kleinrock. *Queueing systems, volume I: theory*. Wiley Interscience, 1975.

[54] Vidyadhar G Kulkarni. Fluid models for single buffer systems. *Frontiers in queueing: Models and applications in science and engineering*, 321:338, 1997.

[55] C Lakshmi and Sivakumar Appa Iyer. Application of queueing theory in health care: A literature review. *Operations research for health care*, 2(1):25–39, 2013.

[56] Andreas Lang and Jeffrey L. Arthur. Parameter approximations for phase-type distributions. In *Proc. 1st Int. Conf. on Matrix-Analytic Methods in Stochastic Models*, pages 151–206, 1996.

[57] Guy Latouche and Vaidyanathan Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*, volume 5. SIAM, 1999.

[58] Alan J Laub. *Matrix analysis for scientists and engineers*. Siam, 2005.

[59] Charles L Lawson and Richard J Hanson. *Solving least squares problems*, volume 15. SIAM, 1995.

[60] Sheng-Kun Li and Ting-Zhu Huang. Two matrix iterative methods for solving general coupled matrix equations. In *Proceedings of the 2010 International Conference on Computational and Information Sciences*, ICCIS '10, pages 384–387, Washington, DC, USA, 2010. IEEE Computer Society.

[61] Rupert G Miller Jr. Priority queues. *The Annals of Mathematical Statistics*, pages 86–103, 1960.

[62] K. Mitchell. Constructing a correlated sequence of matrix exponentials with invariant first order properties. *Operations Research Letters*, 28:27–34, 2001.

[63] Isi Mitrani and Ram Chakka. Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method. *Performance Evaluation*, 23(3):241–260, 1995.

[64] Ş. Mocanu and C. Commault. Sparse representations of phase-type distributions. *Stochastic Models*, 15(4):759–778, 1999.

[65] Marcel F Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation, 1981.

[66] Colm Art O'Cinneide. On non-uniqueness of representations of phase-type distributions. *Communications in Statistics. Stochastic Models*, 5(2):247–259, 1989.

[67] Colm Art O'cinneide. Phase-type distributions: open problems and a few properties. *Stochastic Models*, 15(4):731–757, 1999.

[68] Hiroyuki Okamura and Tadashi Dohi. Faster maximum likelihood estimation algorithms for Markovian arrival processes. In *Quantitative Evaluation of Systems, 2009. QEST'09. Sixth International Conference on the*, pages 73–82. IEEE, 2009.

[69] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking (ToN)*, 3(3):226–244, 1995.

[70] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Stochastic Models*, 4(1):183–188, 1988.

[71] V. Ramaswami. Matrix analytic methods for stochastic fluid flows. In *Teletraffic Engineering in a Competitive World - Proc. of the 16th International Teletraffic Congress (ITC 16)*, pages 1019–1030. Elsevier Science B.V., 1999.

[72] V Ramaswami, Douglas G Woolford, and David A Stanford. The Erlangization method for Markovian fluid flows. *Annals of Operations Research*, 160(1):215–225, 2008.

[73] J. Roberts, U. Mocci, and J. Virtamo (eds.). *Broadband Network Teletraffic*. Springer, 1996.

[74] R. Sadre and B.R. Haverkort. Characterizing traffic streams in networks of MAP/MAP/1 queues. In *Proceedings 11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB 2001)*, pages 195–208. VDE Verlag, 2001.

[75] Bhaskar Sengupta. Markov processes whose steady state distribution is matrix-exponential with an application to the GI/PH/1 queue. *Advances in Applied Probability*, pages 159–180, 1989.

[76] S Soares and G Latouche. Further results on the similarity between fluid queues and QBDs. In *Proceedings of the 4th international conference on matrix-analytic methods*, pages 89–106, 2002.

[77] Willi-Hans Steeb. *Matrix calculus and Kronecker product with applications and C++ programs*. World Scientific, 1997.

[78] Tetsuya Takine. The workload in the MAP/G/1 queue with state-dependent services: Its application to a queue with preemptive resume priority. *Stochastic Models*, 10(1):183–204, 1994.

[79] Tetsuya Takine. A nonpreemptive priority MAP/G/1 queue with two classes of customers. *Journal of Operations Research Society of Japan*, 39(2):266–290, 1996.

[80] Tetsuya Takine. The nonpreemptive priority MAP/G/1 queue. *Operations Research*, 47(6):917–927, 1999.

[81] M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*, 64(9-12):1153–1168, Aug. 2007.

[82] A. van de Liefvoort. The moment problem for continuous distributions. Technical report, University of Missouri, WP-CM-1990-02, Kansas City, 1990.

[83] Benny Van Houdt. Analysis of the adaptive MMAP[K]/PH[K]/1 queue: a multi-type queue with adaptive arrivals and general impatience. *European Journal of Operational Research*, 220(3):695–704, 2012.

[84] C. Van Loan. Computing integrals involving the matrix exponential. *Automatic Control, IEEE Transactions on*, 23(3):395–404, Jun 1978.

[85] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999.

[86] W. G. Wang, W. C. Wang, and R. C. Li. Alternating-directional doubling algorithm for M-matrix algebraic Riccati equations. *SIAM Journal on Matrix Analysis and Applications*, 33(1):170–194, 2012.

[87] W. Whitt. Approximating a point process by a renewal process, I : Two basic methods. *Operations Research*, pages 125–147, 1982.

[88] W. Whitt. Approximations for departure processes and queues in series. *Naval Research Logistics Quarterly*, pages 499–521, 1984.

[89] Li Xie, Huizhong Yang, Yanjun Liu, and Feng Ding. Iterative solutions for general coupled matrix equations with real coefficients. In *American Control Conference (ACC), 2011*, pages 669 –674, 29 2011-july 1 2011.

[90] Ji-An Zhao, Bo Li, Xi-Ren Cao, and Ishfaq Ahmad. A matrix-analytic solution for the DBMAP/PH/1 priority queue. *Queueing Systems*, 53:127–145, 2006.

[91] Falko Bause and Gábor Horváth. Fitting Markovian arrival processes by incorporating correlation into phase type renewal processes. In *Quantitative Evaluation of Systems (QEST), 2010 Seventh International Conference on the*, pages 97–106. IEEE, 2010.

[92] András Horváth, Gábor Horváth, and Miklós Telek. A joint moments based analysis of networks of MAP/MAP/1 queues. *Performance Evaluation*, 67(9):759–778, 2010.

[93] Gábor Horváth. Moment matching-based distribution fitting with generalized hyper-Erlang distributions. In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pages 232–246. Springer, 2013.

[94] Gábor Horváth and Miklós Telek. A canonical representation of order 3 phase type distributions. In *European Performance Engineering Workshop*, pages 48–62. Springer, 2007.

[95] Gábor Horváth and Miklós Telek. On the canonical representation of phase type distributions. *Performance Evaluation*, 66(8):396–409, 2009.

[96] Gábor Horváth and Miklós Telek. Fitting methods based on distance measures of marked Markov arrival processes. In *Seminal Contributions to Modelling and Simulation*, pages 159–183. Springer, 2016.

[97] Gábor Horváth and Benny Van Houdt. Departure process analysis of the multi-type MMAP[K]/PH[K]/1 FCFS queue. *Performance Evaluation*, 70(6):423–439, 2013.