Chapter 1
Phase Type and Matrix Exponential distributions in stochastic modelling

Andras Horvath, Marco Scarpa, Miklos Telek

Abstract Since their introduction, properties of Phase Type (PH) distributions have been analyzed and many interesting theoretical results found. Thanks to these results, PH distributions have been profitably used in many modeling contexts where non exponentially distributed behavior is present. Matrix Exponential (ME) distributions are distributions whose matrix representation is structurally similar to that of PH distributions but represent a larger class. For this reason, ME distributions can be usefully employed in modeling contexts in place of PH distributions by using the same computational techniques and similar algorithms, giving rise to new opportunities the fact, they are able to represent different dynamics, e.g. faster dynamics, or the same dynamics but at lower computational cost. In this work, we deal with the characteristics of PH and ME distributions, and their use in stochastic analysis of complex systems. Moreover, the techniques used in the analysis to take advantage of them are revised.

1.1 Introduction

Stochastic modeling has been used for performance analysis and optimization of computer systems for more than five decades [18]. The main analysis method behind this effort was the continuous time Markov chains (CTMC) description of the system behavior and the CTMC based analysis of the performance measures of interest. With the evolution of computing devices,

Andras Horvath
University of Torino, Torino, Italy e-mail: horvath@di.unito.it

Marco Scarpa
University of Messina, Messina, Italy e-mail: mscarpa@unime.it

Miklos Telek
Technical University of Budapest, Budapest, Hungary e-mail: telek@hit.bme.hu

model description languages (e.g., queueing systems, Petri nets, process algebras), and model analysis techniques (a wide range of software tools with efficient analysis algorithm using adequate data representation and memory management) the analysis of more and more complex systems has become possible. One of main modeling limitations of the CTMC based approach is the limitation on the distribution of the random time durations, which is restricted to be exponentially distributed. Unfortunately, in a wide range of practical applications, the empirical distribution of field data differs significantly from the exponential distribution. The effort to relax this restriction of the CTMC based modeling on exponentially distributed durations resulted in the development of many alternative stochastic modeling methodologies (semi-Markov and Markov regenerative processes [11], analysis with the use of continuous system parameters [8]), yet all of the alternative modeling methodologies suffer from infeasible computational complexity very quickly when the complexity of the systems considered increases beyond basic examples.

It remains a significant research challenge to relax the modeling restriction of the exponentially distributed duration time and still evaluate complex model behaviors. To this end, one of the most promising approaches is the extension of CTMC based analysis to non-exponentially distributed durations. Initial steps in this direction date back to the activity of A. K. Erlang in the first decades of the twenties century as reported in [10]. These initial trials were referred to as the method of phases, which influenced later terminology. M. F. Neuts characterized a set of distributions which can be incorporated into CTMC based analysis by introducing the set of phase type (PH) distributions [16].

The extension of CTMC based analysis (where the durations are exponentially distributed) with PH distributed durations requires the generation of a large CTMC, referred to as extended Markov chain (EMC), which combines the system behavior with the description of the PH distributions. In this chapter, we summarize the basics of EMC based stochastic analysis and provide some application examples. Finally, we note that in this work we restrict our attention to continuous time stochastic models, but that the same approach applies for discrete time stochastic models as well.

Structure of the chapter

The next two sections, Section 1.2 and 1.3, summarize the basic information on PH and ME distributions, respectively. The following two sections, Section 1.4 and 1.5, discuss the analysis procedure for complex stochastic systems with PH and ME distributed durations, respectively. The tools available to support EMC based analysis of stochastic systems is presented in Section 1.6. Numerical examples demonstrate the modeling and analysis capabilities of the approach are discussed in Section 1.7 and the main findings and conclusions are given in Section 1.8.

1.2 PH distributions and their basic properties

1.2.1 Assumed knowledge

Transient behavior of a finite state Markov chain with generator $\mathbf{Q}$ and initial distribution π, specifically, the transient probability vector $p(t)$, satisfies the ordinary differential equation

$$\frac{d}{dt}p(t) = p(t)\mathbf{Q}, \text{ with initial condition } p(0) = \pi,$$

whose solution is a matrix exponential function

$$p(t) = \pi e^{\mathbf{Q}t}, \tag{1.1}$$

where the matrix exponential term is defined as

$$e^{\mathbf{Q}t} = \sum_{i=0}^{\infty} \frac{t^i}{i!}\mathbf{Q}^i.$$

The properties of generator $\mathbf{Q}$ and initial distribution π are as follows. The elements of π are probabilities, i.e. non-negative numbers not greater than one. The off diagonal elements of $\mathbf{Q}$ are transition intensities, i.e. non-negative numbers. The diagonal elements of $\mathbf{Q}$ are such that each row sum is zero, i.e. the diagonal elements are non-positive. The elements of π sum to one, that is $\sum_i \pi_i = \pi\mathbf{1} = 1$. Each row of a generator matrix sums to zero, that is $\sum_j Q_{ij} = 0$, or equivalently, in vector form, we can write $\mathbf{Q1} = \mathbf{0}$, where $\mathbf{1}$ is a column vector of ones and $\mathbf{0}$ is a column vector of zeros. Hereafter, the sizes of vector $\mathbf{1}$ and $\mathbf{0}$ are defined by the context such that the dimensions in the vector expressions are compatible.

The stationary distribution of an irreducible finite state Markov chain with generator $\mathbf{Q}$, $p \triangleq \lim_{t\to\infty} p(t)$, can be computed as the unique solution of the linear system of equations

$$p\mathbf{Q} = \mathbf{0}, \quad p\mathbf{1} = 1. \tag{1.2}$$

In this chapter we focus on the computation of the initial distribution and the generator matrix of the EMC and do not discuss the efficient solution methods for solving (1.1) and (1.2).

1.2.2 Phase type distributions

PH distributions are defined by the behavior of a Markov chain, which is often referred to as the background Markov chain behind a PH.

Let $X(t)$ be a Markov chain with n transient and one absorbing states, meaning that the absorbing state is reachable (by a series of state transitions) from all transient states, but when the Markov chain moves to the absorbing state it remains there forever. Let π be the initial distribution of the Markov chain, that is $\pi_i = P\left(X(0) = i\right)$. Without loss of generality we number the states of the Markov chain such that state $1, \ldots, n$ are transient states and state $n + 1$ is the absorbing state. The generator matrix of such a Markov chain has the following structure

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \mathbf{0} & 0 \end{bmatrix},$$

where $\boldsymbol{A}$ is a square matrix of size n and $\boldsymbol{a}$ is a column vector of size n. Since the rows of the generator matrix sum to zero, the elements of $\boldsymbol{a}$ can be computed from $\boldsymbol{A}$, that is $\boldsymbol{a} = -\boldsymbol{A}\mathbf{1}$. Similarly, the first n elements of the initial vector π, denoted by $\boldsymbol{\alpha}$, completely defines the initial vector, since the $(n + 1)$st element of π is $1 - \boldsymbol{\alpha}\mathbf{1}$. We note that, $\boldsymbol{\alpha}$ defines the initial probabilities of the transient states. With the help of this Markov chain, we are ready to define PH distributions.

Definition 1. The time to reach the absorbing state of a Markov chain with a finite number of transient and an absorbing state

$$T = \min\{t : X(t) = n + 1, t \geq 0\},$$

is phase type distributed.

Throughout this document we assume that the Markov chain starts from one of the transient states and consequently $\boldsymbol{\alpha}\mathbf{1} = 1$, i.e., there is no probability mass at zero and T has a continuous distribution on $\mathbb{R}^+$. Since the time to reach the absorbing state is a transient measure of the Markov chain, we can evaluate the distribution of random variable T, based on the transient analysis of the Markov chain with initial distribution π and and generator matrix $\mathbf{Q}$

$$F_T(t) = P\left(T < t\right) = P\left(X(t) = n + 1\right) = \pi e^{\mathbf{Q}t} e_{n+1},$$

where e_{n+1} is the $(n+1)$st unit vector (the column vector with zero elements except in position $n + 1$ which is one).

This straight forward description of the distribution of T is not widely used due to the redundancy of matrix $\mathbf{Q}$ and vector π. Indeed, matrix $\boldsymbol{A}$ and the initial vector associated with the transient states, $\boldsymbol{\alpha}$, define all information about the distribution of T and the analytical description based on $\boldsymbol{\alpha}$ and

$\boldsymbol{A}$ is much simpler to use in more complex stochastic models. To obtain the distribution based on $\boldsymbol{\alpha}$ and $\boldsymbol{A}$, we carry on the block structure of matrix $\mathbf{Q}$ in the computation.

$$F_T(t) = P(T < t) = P\left(X(t) = n+1\right) = 1 - \sum_{i=1}^{n} P\left(X(t) = n+1\right) =$$
$$= 1 - [\boldsymbol{\alpha}, 0] e^{\mathbf{Q}t} \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} = 1 - [\boldsymbol{\alpha}, 0] \sum_{i=0}^{\infty} \frac{t^i}{i!} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \mathbf{0} & 0 \end{bmatrix}^i \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}$$
$$= 1 - [\boldsymbol{\alpha}, 0] \sum_{i=0}^{\infty} \frac{t^i}{i!} \begin{bmatrix} \boldsymbol{A}^i & \bullet \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} = 1 - \boldsymbol{\alpha} \sum_{i=0}^{\infty} \frac{t^i}{i!} \boldsymbol{A}^i \mathbf{1} = 1 - \boldsymbol{\alpha} e^{\boldsymbol{A}t} \mathbf{1},$$

where $\bullet$ indicates irrelevant matrix block whose elements are multiplied by zero. The PDF of T can be obtained from the derivative of its CDF.

$$f_T(t) = \frac{d}{dt} F_T(t) = \frac{d}{dt} \left(1 - \boldsymbol{\alpha} \sum_{i=0}^{\infty} \frac{t^i}{i!} \boldsymbol{A}^i \mathbf{1} \right) = -\boldsymbol{\alpha} \sum_{i=0}^{\infty} \frac{d}{dt} \frac{t^i}{i!} \boldsymbol{A}^i \mathbf{1}$$
$$= -\boldsymbol{\alpha} \sum_{i=1}^{\infty} \frac{t^{i-1}}{(i-1)!} \boldsymbol{A}^{i-1} \boldsymbol{A} \mathbf{1} = -\boldsymbol{\alpha} e^{\boldsymbol{A}t} \boldsymbol{A} \mathbf{1} = \boldsymbol{\alpha} e^{\boldsymbol{A}t} \boldsymbol{a},$$

where we used $\boldsymbol{a} = -\boldsymbol{A}\mathbf{1}$ in the last step.

Before computing the remaining properties of PH distributions we need to classify the eigenvalues of $\boldsymbol{A}$. The i, j element of matrix $e^{\boldsymbol{A}t}$ contains the probability that starting from transient state i the Markov chain is in transient state j at time t. If states $1, \ldots, n$ are transient states then as t tends to infinity $e^{\boldsymbol{A}t}$ tends to zero, which means that the eigenvalues of $\boldsymbol{A}$ have negative real part and, as a consequence, $\boldsymbol{A}$ is non singular.

The Laplace transform of T, $E\left(e^{-sT}\right)$, can be computed as

$$f_T^*(s) = E\left(e^{-sT}\right) = \int_{t=0}^{\infty} e^{-st} f_T(t) dt = \int_{t=0}^{\infty} e^{-st} \boldsymbol{\alpha} e^{\boldsymbol{A}t} \boldsymbol{a} dt$$
$$= \boldsymbol{\alpha} \int_{t=0}^{\infty} e^{(-s\mathbf{I}+\boldsymbol{A})t} dt \boldsymbol{a} = \boldsymbol{\alpha} (s\mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{a},$$

where we note that the integral surely converges for $\mathcal{R}(s) \geq 0$ because in this cases the eigenvalues of $-s\mathbf{I} + \boldsymbol{A}$ also possess a negative real part.

To compute the kth moment of T, $E\left(T^k\right)$, we need the following integral relation

$$\left[t^k e^{\boldsymbol{A}t}\right]_0^{\infty} = \int_{t=0}^{\infty} k t^{k-1} e^{\boldsymbol{A}t} dt + \int_{t=0}^{\infty} t^k e^{\boldsymbol{A}t} \boldsymbol{A} dt,$$

whose left hand side is zero because the eigenvalues of $\boldsymbol{A}$ possess a negative real part. Multiplying both side with $(-\boldsymbol{A})^{-1}$ we get

$$\int_{t=0}^{\infty} t^k e^{\boldsymbol{A}t} dt = k \int_{t=0}^{\infty} t^{k-1} e^{\boldsymbol{A}t} dt (-\boldsymbol{A})^{-1}.$$

Using this relation, the kth moment of T is

$$E\left(T^k\right) = \int_{t=0}^{\infty} t^k f_T(t) dt = \boldsymbol{\alpha} \int_{t=0}^{\infty} t^k e^{\boldsymbol{A}t} dt (-\boldsymbol{A}) \mathbf{1} = k\boldsymbol{\alpha} \int_{t=0}^{\infty} t^{k-1} e^{\boldsymbol{A}t} dt \mathbf{1} =$$

$$= k(k-1)\boldsymbol{\alpha} \int_{t=0}^{\infty} t^{k-2} e^{\boldsymbol{A}t} dt (-\boldsymbol{A})^{-1} \mathbf{1} = \ldots = k! \boldsymbol{\alpha} (-\boldsymbol{A})^{-k} \mathbf{1}.$$

These four properties of PH distributions (CDF, PDF, Laplace transform, moments) have several interesting consequences some of which we summarize below.

- Matrix $(-\boldsymbol{A})^{-1}$ has an important stochastic meaning. Let T_{ij} be the time spent in transient state j before moving to the absorbing state when the Markov chain starts from state i. For $E\left(T_{ij}\right)$, we have

$$E\left(T_{ij}\right) = \frac{\delta_{ij}}{-\boldsymbol{A}_{ii}} + \sum_{k,k \neq i} \frac{\boldsymbol{A}_{ik}}{-\boldsymbol{A}_{ii}} E\left(T_{kj}\right),$$

  where δ_{ij} is the Kronecker delta symbol. The first term of the left hand side is the time spent in state j while the Markov chain is in the initial state, and the second term is the time spent in state j during later visits to j. Multiplying both sides by $-\boldsymbol{A}_{ii}$ and adding $E\left(T_{ij}\right) \boldsymbol{A}_{ii}$ gives

$$0 = \delta_{ij} + \sum_{k} \boldsymbol{A}_{ik} E\left(T_{kj}\right),$$

  whose matrix form is

$$\mathbf{0} = \mathbf{I} + \boldsymbol{A}\overline{\mathbf{T}} \quad \longrightarrow \quad \overline{\mathbf{T}} = (-\boldsymbol{A})^{-1},$$

  where $\overline{\mathbf{T}}$ is the matrix composed of the elements $E\left(T_{ij}\right)$. Consequently, the (ij) element of $(-\boldsymbol{A})^{-1}$ is $E\left(T_{ij}\right)$, which is a non-negative number.
- $f_T^*(s)$ is a rational function of s whose numerator is at most order $n-1$ and denominator is at most order n. This is because

$$f_T^*(s) = \boldsymbol{\alpha}(s\mathbf{I} - \boldsymbol{A})^{-1}\boldsymbol{a} = \sum_i \sum_j \boldsymbol{\alpha}_i (s\mathbf{I} - \boldsymbol{A})^{-1}_{ij} \boldsymbol{a}_j$$

$$= \sum_i \sum_j \boldsymbol{\alpha}_i \left[\frac{\det_{ji}(s\mathbf{I} - \boldsymbol{A})}{\det(s\mathbf{I} - \boldsymbol{A})}\right] \boldsymbol{a}_j = \frac{\sum_i \sum_j \boldsymbol{\alpha}_i \boldsymbol{a}_j \det_{ji}(s\mathbf{I} - \boldsymbol{A})}{\det(s\mathbf{I} - \boldsymbol{A})}.$$

  $\det_{ji}(\mathbf{M})$ denotes the determinant of the matrix obtained by removing row j and column i of matrix $\mathbf{M}$. The denominator of the last expression is an order n polynomial of s, while the numerator is the sum of order $n-1$ polynomials, which is at most an order $n-1$ polynomial of s.

- This rational Laplace transform representation indicates that a PH distribution with n transient state can be represented by $2n-1$ independent parameters. A polynomial of order n is defined by $n+1$ coefficients, and a rational function of order $n-1$ numerator, and order n denominator is defined by $2n+1$ parameters. Normalizing the denominator such that the coefficient of s^n is 1 and considering that $\int_t f_T(t)dt = \lim_{s\to 0} f_T^*(s) = 1$ adds two constraints for the coefficients, from which the number of independent parameters is $2n-1$.
- The PDF of a PH distribution is the sum of exponential functions. Let $\boldsymbol{A} = \mathbf{B}^{-1}\mathbf{\Lambda}\mathbf{B}$ be the Jordan decomposition[1] of $\boldsymbol{A}$ and let $u = \boldsymbol{\alpha}\mathbf{B}^{-1}$ and $v = \mathbf{B}\boldsymbol{a}$. Then,

$$f_T(t) = \boldsymbol{\alpha} e^{\boldsymbol{A}t}\boldsymbol{a} = \boldsymbol{\alpha}\mathbf{B}^{-1}e^{\mathbf{\Lambda}t}\mathbf{B}\boldsymbol{a} = ue^{\mathbf{\Lambda}t}v \ .$$

  At this point, we distinguish two cases.

  - The eigenvalues of $\boldsymbol{A}$ are different and $\mathbf{\Lambda}$ is a diagonal matrix. In this case, $f_T(t)$ is a sum of exponential functions because

$$f_T(t) = ue^{\mathbf{\Lambda}t}v = \sum_i u_i v_i e^{\lambda_i t} = \sum_i c_i e^{\lambda_i t},$$

    where $c_i = u_i v_i$ is a constant coefficient of the exponential function. Here, the eigenvalues (λ_i) as well as the associated coefficients (c_i) can be real or complex conjugate pairs. For a complex conjugate pair of eigenvalues, we have

$$c_i e^{\lambda_i t} + \bar{c}_i e^{\bar{\lambda}_i t} = 2|c_i| e^{\mathcal{R}(\lambda_i)t}\cos(\mathcal{I}(\lambda_i)t - \varphi_i),$$

    where $c_i = |c_i|e^{\imath\varphi_i}$, $\mathcal{R}(\lambda_i)$ and $\mathcal{I}(\lambda_i)$ are the real and the imaginary part of λ_i and $\imath$ is the imaginary unit.
  - There are eigenvalues of $\boldsymbol{A}$ with higher multiplicity and $\mathbf{\Lambda}$ contains real Jordan blocks. The matrix exponent of a Jordan block is

$$\exp\left[\begin{pmatrix} \lambda & 1 & & \\ & \lambda & 1 & \\ & & \ddots & \ddots \\ & & & \lambda \end{pmatrix} t\right] = \begin{pmatrix} e^{\lambda t} & te^{\lambda t} & \frac{1}{2!}t^2e^{\lambda t} & \frac{1}{3!}t^3e^{\lambda t} \\ & e^{\lambda t} & te^{\lambda t} & \frac{1}{2!}t^2e^{\lambda t} \\ & & \ddots & \ddots \\ & & & e^{\lambda t} \end{pmatrix} .$$

    Consequently, the density function takes the form

[1] The case of different Jordan blocks with identical eigenvalue is not considered here, because it cannot occur in non-redundant PH representations.

$$f_T(t) = \sum_{i=1}^{\#\lambda} \sum_{j=1}^{\#\lambda_i} c_{ij} t^{j-1} e^{\lambda_i t},$$

where $\#\lambda$ is the number of different eigenvalues and $\#\lambda_i$ is the multiplicity of λ_i.
Similar to the previous case, the eigenvalues (λ_i) as well as the associated coefficients $(c_{i,j})$ can be real or complex conjugate pairs. For a complex conjugate pair of eigenvalues, we have

$$c_{i,j} t^{j-1} e^{\lambda_i t} + \bar{c}_{i,j} t^{j-1} e^{\bar{\lambda}_i t} = 2|c_{i,j}| t^{j-1} e^{\mathcal{R}(\lambda_i) t} \cos(\mathcal{I}(\lambda_i) t - \varphi_{i,j}),$$

where $c_{i,j} = |c_{i,j}| e^{\imath \varphi_{i,j}}$.

As a result of all of these cases, the density function of a PH distribution possesses the form

$$f_T(t) = \sum_{i=1}^{\#\lambda_R} \sum_{j=1}^{\#\lambda_i^R} c_{ij} t^{j-1} e^{\lambda_i^R t} + \sum_{i=1}^{\#\lambda_C} \sum_{j=1}^{\#\lambda_i^C} 2|c_{i,j}| t^{j-1} e^{\mathcal{R}(\lambda_i^C) t} \cos(\mathcal{I}(\lambda_i^C) t - \varphi_{i,j}) \tag{1.3}$$

where $\#\lambda_R$ is the number of different real eigenvalues and $\#\lambda_C$ is the number of different complex conjugate eigenvalue pairs.

- In general, infinitely many Markov chains can represent the same PH distribution.

  - The following *similarity transformation* generates representations with identical size.
    Let $\mathbf{T}$ be a non-singular matrix with unit row sums ($\mathbf{T}\mathbf{1} = \mathbf{1}$). The vector matrix pairs $(\boldsymbol{\alpha}, \boldsymbol{A})$ and $(\boldsymbol{\alpha}\mathbf{T}, \mathbf{T}^{-1}\boldsymbol{A}\mathbf{T})$ are two different vector matrix representations of the same PH distribution, since

    $$F_T(t) = 1 - \boldsymbol{\alpha}\mathbf{T} e^{\mathbf{T}^{-1}\boldsymbol{A}\mathbf{T} t} \mathbf{1} = 1 - \boldsymbol{\alpha}\mathbf{T}\mathbf{T}^{-1} e^{\boldsymbol{A}t} \mathbf{T}\mathbf{1} = 1 - \boldsymbol{\alpha} e^{\boldsymbol{A}t} \mathbf{1}.$$

  - Representations with different sizes can be obtained as follows.
    Let matrix $\mathbf{V}$ of size $m \times n$ be such that $\mathbf{V}\mathbf{1} = \mathbf{1}$.
    The vector matrix pairs $(\boldsymbol{\alpha}, \boldsymbol{A})$ of size n and $(\boldsymbol{\gamma}, \mathbf{G})$ of size m are two different vector matrix representations of the same PH distribution if $\boldsymbol{A}\mathbf{V} = \mathbf{V}\mathbf{G}$ and $\boldsymbol{\alpha}\mathbf{V} = \boldsymbol{\gamma}$ because

    $$F_T(t) = 1 - \boldsymbol{\gamma} e^{\mathbf{G}t} \mathbf{1} = 1 - \boldsymbol{\alpha}\mathbf{V} e^{\mathbf{G}t} \mathbf{1} = 1 - \boldsymbol{\alpha} e^{\boldsymbol{A}t} \mathbf{V}\mathbf{1} = 1 - \boldsymbol{\alpha} e^{\boldsymbol{A}t} \mathbf{1}$$

    in this case.

1.3 Matrix exponential distributions and their basic properties

In the definition of PH distributions, vector $\boldsymbol{\alpha}$ is a probability vector with non-negative elements and matrix $\boldsymbol{A}$ is a generator matrix with negative diagonal and non-negative off diagonal elements. Relaxing these sign constraints for the vector and matrix elements and maintaining the matrix exponential distribution (and density) function results in the set of matrix exponential (ME) distributions.

Definition 2. Random variable T with distribution function

$$F_T(t) = 1 - \boldsymbol{\alpha} e^{\boldsymbol{A}t}\mathbf{1},$$

where $\boldsymbol{\alpha}$ is a finite real vector and $\boldsymbol{A}$ is a finite real matrix, is matrix exponentially distributed.

The size of $\boldsymbol{\alpha}$ and $\boldsymbol{A}$ plays the same role as the number of transient states in case of PH distributions. By definition, the set of PH distributions with a given size is a subset of the set of PH distributions with the same size.

ME distributions share the following basic properties with PH distributions: matrix exponential distribution function, matrix exponential density function, moments, rational Laplace transform, the same set of functions as in (1.3), and non-unique representation. The main difference between the matrix exponential and the PH classes comes from the fact that the sign constraints on the elements of generator matrixes restrict the eigenvalue structure of such matrixes, while such restrictions do not apply in case of ME distributions. For example, the eigenvalues of an order three PH distribution with dominant eigenvalue θ satisfy $\mathcal{R}(\lambda_i) \leq \theta$ and $|\mathcal{I}(\lambda_i)| \leq (\theta - \mathcal{R}(\lambda_i)/\sqrt{3}$, while the eigenvalues of an order three ME distribution with dominant eigenvalue θ satisfy $\mathcal{R}(\lambda_i) \leq \theta$ only. This flexibility of the eigenvalues has significant consequence on the flexibility of the set of order three PH and ME distributions. For example, the minimal squared coefficient of variation among the order three PH and ME distributions are 1/3 and 0.200902 respectively.

The main difficulty encountered when working with ME distributions is that a general vector-matrix pair does not always define a non-negative density function, while a vector-matrix pair with the sign constraints of PH distributions does. Efficient numerical methods have been proposed recently to check the non-negativity of a matrix exponential function defined by a general vector-matrix pair, but general symbolic conditions are still missing.

1.4 Analysis of models with PH distributed durations

If all durations (service times, inter-arrival times, repair times, etc.) in a system are distributed according to PH distributions, then its overall behavior can be captured by a continuous time Markov chain, referred to as extended Markov chain (EMC). In this section we show how to derive the infinitesimal generator of this EMC by using Kronecker operations.

To this end we first introduce the notation used to describe the model. By $\mathcal{S}$ we denote the set of states and by $N = |\mathcal{S}|$ the number of states. The states itself are denoted by $s_1, s_2, ..., s_N$. The set of activities (services, arrivals, repairs, etc.) is denoted by $\mathcal{A}$ and the set of those that are active in state s_i is denoted by $\mathcal{A}_i$. The activities itself are denoted by $a_1, a_2, ..., a_M$ with $M = |\mathcal{A}|$. When activity a_i is completed in state s_j then the system moves from state s_j to state $n(j, i)$, i.e., n is the function that provides the next state. We assume that the next state is a deterministic function of the current state and the activity that completes. We further assume that there does not exist a triple, k, i, j, for which $s_k \in \mathcal{S}, a_i \in \mathcal{A}, a_j \in \mathcal{A}$ and $n(k, i) = n(k, j)$. These two assumptions, which make the formulas simpler, are easy to relax in practice. There can be activities that are put to an end when the system moves from state s_i to state s_j even if they do not complete and are active both in s_i and in s_j. These activities are collected in the set $e(i, j)$. The PH distribution that is associated with activity a_i is characterized by the initial vector $\boldsymbol{\alpha}_i$ and by the matrix $\boldsymbol{A}_i$. As before, we use the notation $\boldsymbol{a}_i = -\boldsymbol{A}_i\mathbf{1}$ to refer to the vector containing the intensities that lead to completion of activity a_i. The number of phases of the PH distribution associated with activity a_i is denoted by n_i.

Example 1. PH/PH/1/K queue with server break-downs. As an example, using the above introduced notation we describe a queue in which the server is subject to failure that can occur only if the queue is not empty. The set of states is $\mathcal{S} = \{s_1, s_2, ..., s_{2K+1}\}$ where s_1 represents the empty queue, s_{2i} with $1 \leq i \leq K$ represents the state with i clients in the queue and the server up and s_{2i+1} with $1 \leq i \leq K$ represents the state with i clients and the server down. There are four activities in the system: a_1 represents arrival, a_2 service, a_3 failure and a_4 repair. The vectors and matrices that describe the associated PH distributions are $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3, \boldsymbol{\alpha}_4$ and $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4$. In this example we assume that the arrival activity is active if the system is not full and it is inactive if the system is full. The service and the failure activities are active if the queue is not empty and the server is up. The repair activity is active if the queue is not empty and the server is down. Accordingly we have $\mathcal{A}_1 = \{a_1\}$, $\mathcal{A}_{2i} = \{a_1, a_2, a_3\}$ for $1 \leq i \leq K-1$, $\mathcal{A}_{2i+1} = \{a_1, a_4\}$ for $1 \leq i \leq K-1$, $\mathcal{A}_{2K} = \{a_2, a_3\}$ and $\mathcal{A}_{2K+1} = \{a_4\}$. The next state function is as follows: for arrivals we have $n(1, 1) = s_2$ and $n(i, 1) = s_{i+2}$ with $2 \leq i \leq 2K-1$; for services $n(2, 2) = s_1$ and $n(2i, 2) = s_{2i-2}$ with $2 \leq i \leq K$; for failures $n(2i, 3) = s_{2i+1}$ with $1 \leq i \leq K$; for repairs $n(2i+1, 4) = s_{2i}$ with

$1 \leq i \leq K$. We assume that the failure activity is put to an end every time when a service activity completes, i.e., failure is connected to single jobs and not to the aging of the server. Other activities are put to an end only when they complete or when such a state is reached in which they are not active. Accordingly, $e(2i, 2i-2) = \{a_3\}$ for $2 \leq i \leq K$.

Based on the description of the ingredients of the model, it is possible to derive blocks of the initial probability vector and the blocks of the infinitesimal generator of the corresponding CTMC. Let us start with the infinitesimal generator, which we denote by $\boldsymbol{Q}$, composed of $N \times N$ blocks. The block of $\boldsymbol{Q}$ that is situated in the ith row of blocks and in the jth column of blocks is denoted by $\boldsymbol{Q}_{ij}$. A block in the diagonal, $\boldsymbol{Q}_{ii}$ describes the parallel execution of the activities that are active in s_i. The parallel execution of CTMCs can be captured by the Kronecker-sum operator ($\oplus$) and thus we have

$$\boldsymbol{Q}_{ii} = \bigoplus_{j: s_j \in \mathcal{A}_i} \boldsymbol{A}_j \ .$$

An off-diagonal block, $\boldsymbol{Q}_{ij}$, is not a zero matrix only if there exists an activity whose completion moves the system from state s_i to state s_j. Let us assume that the completion of activity a_k moves the system from state s_i to state s_j, i.e., $n(i,k) = s_j$. The corresponding block, $\boldsymbol{Q}_{ij}$,

- has to reflect the fact that activity a_k completes and restarts if a_k is active in s_j,
- has to reflect the fact that activity a_k completes and does not restart if a_k is not active in s_j,
- has to put an end to the activities that are active in s_i but not in s_j,
- has to start those activities that are not active in s_i but are active in s_j,
- has to put an end to and restart those activities that are active both in s_i and in s_j but are in $e(i,j)$,
- and has to maintain the phase of those that are active both in s_i and in s_j and are not in $e(i,j)$.

The joint treatment of the above cases can be carried out by the Kronecker-product operator and thus we have:

$$\boldsymbol{Q}_{ij} = \bigotimes_{l: 1 \leq l \leq M} \boldsymbol{R}_l$$

with

$$\boldsymbol{R}_l = \begin{cases} \boldsymbol{a}_k & \text{if } l = k \text{ and } k \notin \mathcal{A}_j \\ \boldsymbol{a}_k \boldsymbol{\alpha}_k & \text{if } l = k \text{ and } k \in \mathcal{A}_j \\ \mathbf{1}_{n_l} & \text{if } l \neq k \text{ and } k \in \mathcal{A}_i \text{ and } k \notin \mathcal{A}_j \\ \boldsymbol{\alpha}_l & \text{if } l \neq k \text{ and } k \notin \mathcal{A}_i \text{ and } k \in \mathcal{A}_j \\ \mathbf{1}_{n_l} \boldsymbol{\alpha}_l & \text{if } l \neq k \text{ and } k \in \mathcal{A}_i \text{ and } k \in \mathcal{A}_j \text{ and } k \in e(i,j) \\ \mathbf{I}_{n_l} & \text{if } l \neq k \text{ and } k \in \mathcal{A}_i \text{ and } k \in \mathcal{A}_j \text{ and } k \notin e(i,j) \\ 1 & \text{otherwise} \end{cases}$$

where the subscripts to $\mathbf{1}$ and $\mathbf{I}$ indicate their size.

The initial probability vector of the CTMC, $\boldsymbol{\pi}$, is a row vector composed of N blocks. It has to reflect the initial probabilities of the states of the system and the initial probabilities of the PH distributions of the active activities. Denoting by π_i the initial probability of state s_i, the ith block of the initial probability vector, $\boldsymbol{\pi}_i$, is given as

$$\boldsymbol{\pi}_i = \bigotimes_{j: s_j \in \mathcal{A}_i} \boldsymbol{\alpha}_j \ .$$

Example 2. For the previous example, the diagonal blocks, which have to reflect the ongoing activities, are the following:

$$\begin{aligned} &\boldsymbol{Q}_{1,1} = \boldsymbol{A}_1,\ \boldsymbol{Q}_{2i,2i} = \boldsymbol{A}_1 \bigoplus \boldsymbol{A}_2 \bigoplus \boldsymbol{A}_3,\ \boldsymbol{Q}_{2i+1,2i+1} = \boldsymbol{A}_1 \bigoplus \boldsymbol{A}_4, \\ &\boldsymbol{Q}_{2K,2K} = \boldsymbol{A}_2 \bigoplus \boldsymbol{A}_3,\ \boldsymbol{Q}_{2K+1,2K+1} = \boldsymbol{A}_4 \quad \text{with } 1 \leq i \leq K-1 \end{aligned}$$

Arrival in state s_1 takes the system to state s_2. The corresponding block has to complete and restart the arrival activity and has to restart both the service and the failure activity:

$$\boldsymbol{Q}_{12} = \boldsymbol{a}_1 \boldsymbol{\alpha}_1 \bigotimes \boldsymbol{\alpha}_2 \bigotimes \boldsymbol{\alpha}_3 \tag{1.4}$$

Arrival in state s_{2i} (server up) takes the system to state s_{2i+2}. If the system does not become full then the corresponding block has to complete and restart the arrival activity and has to maintain the phase of both the service and the failure activity. If the system becomes full, the arrival activity is not restarted. Accordingly we have

$$\begin{aligned} &\boldsymbol{Q}_{2i,2i+2} = \boldsymbol{a}_1 \boldsymbol{\alpha}_1 \bigotimes \mathbf{I}_{n_2} \bigotimes \mathbf{I}_{n_3} \quad \text{with } 1 \leq i \leq K-2 \\ &\boldsymbol{Q}_{2K-2,2K} = \boldsymbol{a}_1 \bigotimes \mathbf{I}_{n_2} \bigotimes \mathbf{I}_{n_3} \end{aligned}$$

Arrival in state s_{2i+1} (server down) takes the system to state s_{2i+3}. If the system does not become full then the corresponding block has to complete and restart the arrival activity and has to maintain the phase of the repair activity. If the system becomes full, the arrival activity is not restarted. Accordingly we have

$$
\begin{aligned}
\boldsymbol{Q}_{2i+1,2i+3} &= \boldsymbol{a}_1\boldsymbol{\alpha}_1 \bigotimes \mathbf{I}_{n_4} \quad \text{with } 1 \leq i \leq K-2 \\
\boldsymbol{Q}_{2K-1,2K+1} &= \boldsymbol{a}_1 \bigotimes \mathbf{I}_{n_4}
\end{aligned}
$$

Service completion can take place in three different situations. If the system becomes empty then the phase of the arrival activity is maintained, the service activity is completed and the failure activity is put to an end. If the system neither becomes empty nor it was full then the phase of the arrival activity is maintained, the service activity is completed and restarted, and the failure activity is put to an end and it is restarted. Finally, if the queue was full, then the arrival activity is restarted, the service activity is completed and restarted, and the failure activity is put to an end and it is restarted. Accordingly, we have

$$
\begin{aligned}
\boldsymbol{Q}_{2,1} &= \mathbf{I}_{n_1} \bigotimes \boldsymbol{a}_2 \bigotimes \mathbf{1}_{n_3} \\
\boldsymbol{Q}_{2i,2i-1} &= \mathbf{I}_{n_1} \bigotimes \boldsymbol{a}_2\boldsymbol{\alpha}_2 \bigotimes \mathbf{1}_{n_3}\boldsymbol{\alpha}_3 \quad \text{with } 1 < i < K \\
\boldsymbol{Q}_{2K,2K-2} &= \boldsymbol{\alpha}_1 \bigotimes \boldsymbol{a}_2\boldsymbol{\alpha}_2 \bigotimes \mathbf{1}_{n_3}\boldsymbol{\alpha}_3
\end{aligned}
\tag{1.5}
$$

The failure activity can be completed in two different situations. If the system is not full, then the phase of the arrival activity is maintained. If the system is full then the arrival activity is not active. In both cases, the service activity is put to an end, the failure activity is completed and the repair activity is initialized.

$$
\begin{aligned}
\boldsymbol{Q}_{2i,2i+1} &= \mathbf{I}_{n_1} \bigotimes \mathbf{1}_{n_2} \bigotimes \boldsymbol{a}_3 \bigotimes \boldsymbol{\alpha}_4 \quad \text{with } 1 \leq i < K \\
\boldsymbol{Q}_{2K,2K+1} &= \mathbf{1}_{n_2} \bigotimes \boldsymbol{a}_3 \bigotimes \boldsymbol{\alpha}_4
\end{aligned}
$$

Similarly to the failure activity, also the repair activity can be completed in two different situations because the arrival activity can be active or inactive. In both cases, the service activity and the failure activity must be initialized and the repair activity completes.

$$
\begin{aligned}
\boldsymbol{Q}_{2i+1,2i} &= \mathbf{I}_{n_1} \bigotimes \boldsymbol{\alpha}_2 \bigotimes \boldsymbol{\alpha}_3 \bigotimes \boldsymbol{a}_4 \quad \text{with } 1 \leq i < K \\
\boldsymbol{Q}_{2K+1,2K} &= \boldsymbol{\alpha}_2 \bigotimes \boldsymbol{\alpha}_3 \bigotimes \boldsymbol{a}_4
\end{aligned}
$$

1.5 Analysis of stochastic systems with ME distributed durations

The most important message of this section is that all steps of the method of EMCs (as explained in the previous section) remains directly applicable also in case of ME distributed durations (where the $(\boldsymbol{\alpha}_i, \boldsymbol{A}_i)$ vector-matrix pairs describe ME distributions). In that case the only difference is that the signs of the vector and matrix elements are not restricted (to be non-negative in case of the vector elements and off-diagonal matrix elements and to be negative in case of the diagonal matrix elements) and consequently the obtained model description do not allow a probabilistic interpretation via Markov chains.

This general conclusion was obtained in a long research effort. Following the results in [12] it was likely that in a stochastic model ME distributions can be used in place of PH distributions and several results will carry over, but it was not easy to prove results in the general setting because probabilistic arguments associated with PH distributions do no longer hold. In [2] it has been shown that matrix geometric methods can be applied for quasi birth death processes (QBDs) with rational arrival processes (RAPs) [1], which can be viewed as an extension of ME distributions to arrival processes. To prove that the matrix geometric relations hold, the authors of [2] use an interpretation of RAPs that has been proposed in [1]. However, the considered models are limited to QBDs. For the model class of SPNs with ME distributed firing times the applicability of the EMC-like analysis was proved in [4] and refined for the special case when the ME distribution has no PH representation in [3].

1.6 Analysis tools

Based on the common representation of the EMC through the Kronecker algebra, recently smart algorithms have been developed to optimize the memory usage. This algorithms build the EMC in a completely symbolic way both at the process state space level and at the expanded state space level.

The algorithm is based on two high level steps:

1. to generate the reachability graph of the model (which collects the system states in a graph according to their reachability from an initial set of states) by using a symbolic technique to store it;
2. to enrich the symbolically stored reachability graph with all the necessary information to evaluate Kronecker expressions representing the expanded state space.

Step 1 is performed by using symbolic technique based on complex data structures like Multi-Valued Decision Diagram (MDD) [17] to encode the model state space; step 2 adds information related to each event memory policy to the encoded state space. In such a way it is possible to use on the fly expressions introduced in section 1.4 and 1.5 to compute various probability measures of the model.

1.6.1 Symbolic generation of reachability graph

Both traditional performance or dependability evaluation techniques and more recent model checking based approaches are grounded on the knowledge of the set of states that the considered system can reach starting from a particular initial state (or in general from a set of initial states). Symbolic techniques [5] focus on generating compact representation of huge state spaces by exploiting model's structure and regularity. A model has a structure when it is composed of K sub-models, for some $K \in \mathbb{N}$. In this case, a global system state can be represented as a K-tuple $(s^1, \ldots, s^K)$, where s^k is the local state of sub-model k (having some finite size n^k).

The use of (MDDs) for the encoding of model state spaces have been introduced by Miner and Ciardo in [15]. MDDs are rooted, directed, acyclic graphs associated with a finite ordered set of integer variables. When used to encode a state space, an MDD has the following structure:

- nodes are organized into $K+1$ levels, where K is the number of sub-models;
- level K contains only a single non-terminal node, the root, whereas levels $K-1$ through 1 contain one or more non-terminal nodes;
- a non-terminal node at level k has n^k arcs pointing to nodes at level $k-1$;

A state $s = (s^1, \ldots, s^K)$ belongs to S if and only if a path exists from the root node to the terminal node 1, such that at each node the arc corresponding to the local state s^k is followed.

In [6], and then in [7], Ciardo et al. proposed the *Saturation* algorithm for the generation of reachability graphs using MDDs. Such an iteration strategy improves both memory and execution-time efficiency.

An efficient encoding of the reachability graph is built in the form of a set of Kronecker matrices $\mathbf{W}_{e,k}$ with $e \in \mathcal{E}$ and $k = 1, \ldots, K$, where $\mathcal{E}$ is the set collecting all the system events. $\mathbf{W}_{e,k}[i_k, j_k] = 1$ if state j_k of sub-model k is reachable from state i_k due to event e. According to such a definition, the next-state function of the model can be encoded as the incidence matrix given by the boolean sum of Kronecker products $\sum_{e \in \mathcal{E}} \bigotimes_{K \geq k \geq 1} \mathbf{W}_{e,k}$. As a consequence, the matrix representation $\mathbf{R}$ of the reachability graph of the model can be obtained by filtering the rows and columns of such a matrix corresponding to the reachable global states (our macro-states) encoded in the MDD and replacing each non-null element with the labels of the events that cause the corresponding state transition.

Saturation Unbound is a very effective way to represent model state space and the related reachability graph of a model. Anyway, the methodology we are dealing with is not strictly depending on it and whichever algorithm able to efficiently store the reachability graph could be used as alternative. We refer to it just because its efficiency is well known.

1.6.2 Annotating the reachability graph

The use of Saturation together with the Kronecker representation presented in previous Sections allows to solve the derived stochastic process. However, the knowledge of the reachability graph of the untimed system as produced by Saturation is not enough for managing the infinitesimal generator matrix $\mathbf{Q}$ on the fly according to the symbolic representation. Considering that the information about the enabled events for all the system states is contained in the high level description of the model and it can be evaluated on the fly when needed with a negligible overhead, the only further needed information is the knowledge about the sets enabled but not active events in each state s ($T_a^{(s)}$). Using Saturation for the evaluation of the reachability graph brings in this case to the necessity of applying a further analysis step for the computation of such an information and using a different data structure to store it. Multi Terminal Multi-Valued Decision Diagram (MTMDD) [14] is used to this purpose.

The main differences with respect to MDDs are that: 1) more than two terminal nodes are present in an MTMDD, and 2) such nodes can be labeled with arbitrary integer values, rather than just 0 and 1. An MTMDD can efficiently store both the system state space S and the sets $T_a^{(s)}$ of active but not enabled events for all $s \in \mathcal{S}$ that are necessary in our approach for the evaluation of non-null blocks of matrix $\mathbf{Q}$. In fact, while an MDD is only able to encode a state space, an MTMDD is also able to associate an integer to each state. Thus, the encoding of sets $T_a^{(s)}$ can be done associating to each possible set of events an integer code that unequivocally represents it. Let us associate to each event an unique index n such that $1 \leq n \leq \|\mathcal{E}\|$. Then the integer value associated to one of the possible sets $T_a^{(s)}$ is computed starting from the indexes associated to the system events that belong to it in the following way:

$$b_{\|\mathcal{E}\|} \cdot 2^{\|\mathcal{E}\|} + \cdots + b_n \cdot 2^n + \ldots b_1 \cdot 2^1 + 1 = \sum_{i=1}^{\|\mathcal{E}\|} b_i 2^i + 1$$

where:

$$b_i = \begin{cases} 1, \text{ if event } e_i \in T_a^{(s)} \\ 0, \qquad\qquad \text{otherwise} \end{cases}$$

In this way all the necessary information to apply on the fly the Kronecker based expressions are provided; the only final need is a method to evaluate the set $T_a^{(s)}$ given a referring state s.

In [13] the following theorem has been proved.

Theorem 1. *Given a model $\mathcal{M}$, a state $s_0 \in S$ and an event $\overline{e} \in \mathcal{E}$ with an age memory policy associated, then $\overline{e} \in T_a^{(s_0)}$ iff $\overline{e} \notin T_e^{(s_0)}$ and one of the following statements holds:*

1. $\exists\ s_1 \in \mathcal{S},\ \exists\ e_1 \in \mathcal{E}, s_1 \neq s_0,\ e_1 \neq \overline{e} \mid s_0 \in \mathcal{N}_{e_1}(s_1)\ \wedge\ \overline{e} \in T_e^{(s_1)}$
2. $\exists\ s_1 \in \mathcal{S}, s_1 \neq s_0 \mid s_0 \in \mathcal{N}(s_1)\ \wedge\ \overline{e} \in T_a^{(s_1)}$

where $\mathcal{N}_{e_1}$ is the next-state function associated to event e_1.

Theorem 1 gives a way to evaluate if an event e belongs or not to the set $T_a^{(s_0)}$. In fact, according to the proved statements it is possible to characterize a state s_0 with respect to the system event memory policies by exploring its reachability graph. Exploration can be performed by using classical bread-first search and depth-first search algorithms, easily applicable to an explicitly stored reachability graph; it is more complicated applying classical search algorithms when the graph is stored in implicit way like done with the use of MTMDD data structures.

In this case a different approach can be used by resorting to Computational Tree Logic (CTL) formulas that are shown to be very efficiently evaluated over data structures like MDD and MTMDD. The use of CTL formulas to evaluate sets $T_a^{(s)}$ is justified by a theorem introduced in [13]. Before to recall it, we need to introduce a CTL operator.

Definition 3. Let $s_0 \in \mathcal{S}$ be a state of a discrete state process with state space $\mathcal{S}$, and let p and q be two logical conditions on the states. Let also $\mathcal{F}(s) \subseteq \mathcal{N}(s) \cup \mathcal{N}^{-1}(s)$ be a reachability relationship between two states in $\mathcal{S}$ that defines a desired condition over the paths. Then s_0 satisfies the formula $E_{\mathcal{F}}[pUq]$, and we will write $s_0 \vDash E_{\mathcal{F}}[pUq]$, iff $\exists\ n \geq 0, \exists\ s_1 \in \mathcal{F}(s_0), \ldots, \exists\ s_n \in \mathcal{F}(s_{n-1}) \mid (s_n \vDash q) \wedge (\forall m < n, s_m \vDash p)$.

In definition above, we used the path quantifier E with the meaning *there exists a path* and the tense operator U with the meaning *until*, as usually adopted in CTL formulas.

Upon definition 3, the following theorem holds:

Theorem 2. *An event $\overline{e} \in \mathcal{E}$, with an age memory policy associated, belongs to $T_a^{(s_0)}$, with $s_0 \in \mathcal{S}$, iff $s_0 \vDash E_{\mathcal{F}}[pUq]$ over a path at least long one, where p and q are the statements "$\overline{e}$ is not enabled" and "$\overline{e}$ is enabled" respectively, and $\mathcal{F}(s) = \mathcal{N}^{-1}(s) \setminus \mathcal{N}_{\overline{e}}^{-1}(s)$.*

Thanks to Theorem 2, a simple evaluation of the CTL formula $E_{\mathcal{F}}[pUq]$ makes possible to evaluate whether an event $\overline{e}$ is active but not enabled in state s_0 or not by setting condition p as $\overline{e}$ *is not enable* and q as $\overline{e}$ *is enabled.* This is the last brick to build an algorithm able to compute state probabilities of a model where the event are PH or ME distributed; in fact it is possible to characterize all the active and/or enabled events in all the different states and to apply with this information the Kronecker expressions to solve the derived EMC.

1.7 Examples

In this section we present two examples where non-exponentially distributed durations are present. In the first example these durations are approximated by PH distributions, while in the second example they are described by ME distributions.

1.7.1 Reliability model of computer system

We introduce a reliability model where we make use of PH distributions to model failure times. The model is specified through the Petri net depicted in Fig. 1.1, where the usual graphical notation for the places, transitions and arcs has been adopted.

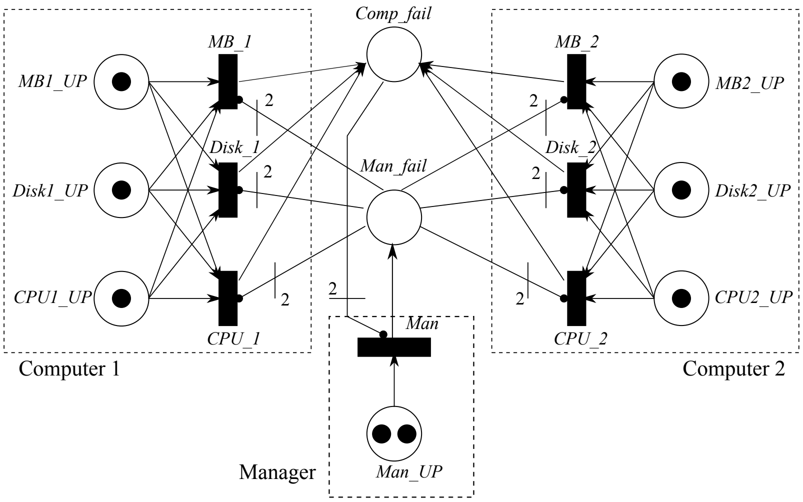

Fig. 1.1 Computer system reliability model.

The system under study is a distributed computing system based on a cluster of two computers. Each of them has three main weak points: the motherboard, the CPU and the disk. Interconnections inside the cluster are provided by a manager in such a way the overall system is seen as a single elaboration unit. In the distributed system, the two computers work independently driven by the manager that acts as load balancer splitting the work between them. Since the manager represents a single point of failure, a second instance of it is deployed as redundancy in the system; this latter operates in cold standby when the main computer manager works and it is powered on when it fails.

Due to this configuration, the distributed system works when at least one of the two computers works and the computer manager properly operates. The main components of each computational unit (CPU, motherboard and disk) may fail making the unit broken. In the Petri net model, the faulty events of CPU, motherboard and disk are modeled by the timed transitions MB_i, $Disk\_i$ and CPU_i whose firing represents the respective faulty event in the i-th Computer; the operating conditions of components are represented by a token in the places $CPUi\_UP$, MBi_UP and $Diski\_UP$. When one of the transitions above fires a token is flushed out of the place and a token is put in the place $Comp\_fail$. At the same time, all the other transitions related to the faulty events in the same unit become disabled because the unit is considered down thus no more faults can occur. Two tokens in the place $Comp\_fail$ means that the two computational units are both broken and the overall distributed system is not operational. Similarly, transition Man models the fault of a manager unit. Its firing flushes a token out of the place Man_UP and puts a token in the place Man_fail. Thanks to the redundancy, the first manager unit fault is tolerated whereas the system goes down when a second fault occurs. This state is represented in the Petri net by two tokens in the place Man_fail. In both faulty states all the transitions become disabled and an absorbing state is reached. In terms of Petri net objects, the not operational condition is expressed by the following statement:

$$(\#Comp\_fail = 2) \vee (\#Man\_fail = 2) \ , \tag{1.6}$$

where the symbol $\#P$ states the number of token in place P.

As usual in reliability modeling the time to failure of the components has been modeled by using Weibull distributions whose cumulative distribution function is

$$F(t) = 1 - e^{(1/\eta_f)^{\beta_f}} .$$

This choice has been also supported by measures done on real systems like that analyzed in [9]. The parameters of the Weibull distributions used for the Petri net transitions of Fig. 1.1 are reported in Table 1.7.1.

Table 1.1 Failure time distribution parameters.

	Weibull			
Transition	β_f	η_f	E	λ
MB_1, MB_2	0.5965	1.20	1.82	0.55
$Disk\_1$, $Disk\_2$	0.5415	1.00	1.71	0.59
CPU_1, CPU_2	0.399	1.46	3.42	0.29
Man	0.5965	1.20	1.82	0.55

Weibull distributions have been introduced in the model through the use of 10-phase PH distributions approximating them and it has been solved evaluating the formula (1.6). The obtained result is depicted in Figure 1.2 as Weibull line.

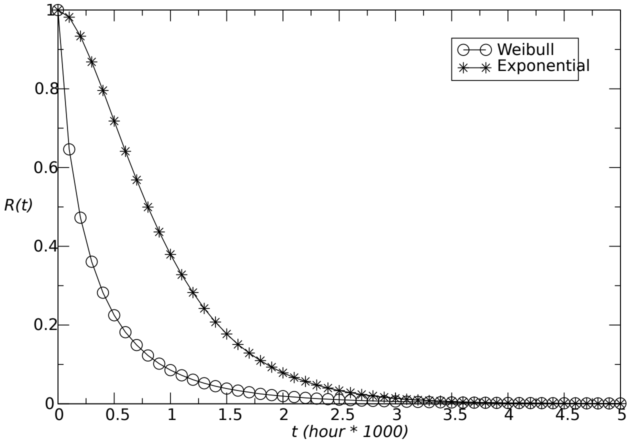

Fig. 1.2 Computer system reliability $R(t)$.

To better highlight the usefulness of the modeling approach here presented, the Petri net model was solved by imposing exponential distributions as transition firing times. In fact the use of exponential distributions is quite usual to have a more tractable model. The value of the parameters λ used in this second run was computed as the reciprocal of the expected value, E, of the corresponding Weibull distributions (listed in Table). The obtained result is depicted in Figure 1.2 as Exponential line. As can be easily note, the use of exponential distributions produces optimistic result with respect to the use of Weibull distributions, making the system look more reliable than it is in reality.

1.7.2 Numerical example with "oscillating" ME distribution

For second example, we consider the Activity Network depicted in Figure 1.3 which represents a "mission" composed of 5 activities and the constraints on the order in which the 5 activities can be carried out. Initially activities 1 and 2 are active. If activity 1 is finished then activities 3 and 4 start and thus there are three activities under execution, namely, activities 2, 3

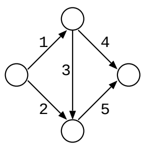

Fig. 1.3 An activity network.

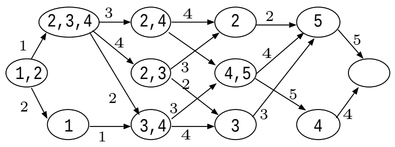

Fig. 1.4 CTMC of the activity network in Figure 1.3.

and 4. If activity 3 finishes first among these three activities then no new activity starts because in order to start activity 5, both activity 2 and 3 must be finished. The graph of all the possible states of the Activity Network is shown in Figure 1.4 where in every node we reported the activities that are under execution in the node. The label on the edges indicates the activity whose accomplishment triggers the edge. The duration of the activities are modeled with ME distributions and we denote the vector and the matrix that represent the duration of activity i by β_i and T_i, respectively. Further, we use the notation $t_i = (-T_i)\mathbf{1}$ and denote by I_i the identity matrix whose dimension is equal to that of T_i.

Following the approach described in Section 1.4, one can determine the infinitesimal generator of the model. Its first 7 block-columns are given as (i.e., the left side of the matrix is)

$$\left|\begin{array}{ccccccc}
T_1\bigoplus T_2 & t_1\bigotimes I_2\bigotimes\beta_3\bigotimes\beta_4 & 0 & 0 & I_1\bigotimes t_2 & 0 & 0 \\
0 & T_2\bigoplus T_3\bigoplus T_4 & I_2\bigotimes I_3\bigotimes t_4 & 0 & 0 & I_2\bigotimes t_3\bigotimes I_4 & t_2\bigotimes I_3\bigotimes I_4 \\
0 & 0 & T_2\bigoplus T_3 & I_2\bigotimes t_3 & 0 & 0 & 0 \\
0 & 0 & 0 & T_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & T_1 & 0 & t_1\bigotimes\beta_3\bigotimes\beta_4 \\
0 & 0 & 0 & I_2\bigotimes t_4 & 0 & T_2\bigoplus T_4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & T_3\bigoplus T_4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right.$$

and the remaining 5 block-columns are (i.e., the right side of the matrix is)

$$\left.\begin{array}{ccccc}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
t_2 \bigotimes I_3 & 0 & 0 & 0 & 0 \\
0 & t_2 \bigotimes \beta_5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & t_2 \bigotimes I_4 \bigotimes \beta_5 & 0 & 0 \\
I_3 \bigotimes t_4 & 0 & t_3 \bigotimes I_4 \bigotimes \beta_5 & 0 & 0 \\
T_3 & t_3 \bigotimes \beta_5 & 0 & 0 & 0 \\
0 & T_5 & 0 & 0 & t_5 \\
0 & t_4 \bigotimes I_5 & T_4 \bigoplus T_5 & I_4 \bigotimes t_5 & 0 \\
0 & 0 & 0 & T_4 & t_4 \\
0 & 0 & 0 & 0 & 0
\end{array}\right|$$

The vector that provides the initial configuration is $|\beta_1 \otimes \beta_2, 0, ..., 0|$.

In order to illustrate a feature of ME distributions that cannot be exhibited by PH distributions, we applied an ME distribution with "oscillating" PDF to describe the duration of activities 1, 2, 4 and 5. The vector-matrix pair of this ME distribution is

$$\beta_1 = \beta_2 = \beta_4 = \beta_5 = |1.04865, -0.0340166, -0.0146293| \ ,$$

$$T_1 = T_2 = T_4 = T_5 = \left|\begin{array}{lll} -1 & 0 & 0 \\ 0 & -1 & -20 \\ 0 & 20 & -1 \end{array}\right| \ ,$$

and its PDF is depicted in Figure 1.5. The duration of the remaining activity, activity 3, is distributed according to an Erlang distribution with 4 phases and with average execution time equal to 1, i.e.,

$$\beta_3 = |1, 0, 0, 0| \ , \quad T_3 = \frac{1}{4} \left|\begin{array}{llll} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{array}\right| \ .$$

The model was then used to characterize the PDF of the time that is needed to accomplish the whole mission. The resulting PDF is shown in Figure 1.6 and one can observe that the oscillating nature of the distribution of the activity durations carries over into the overall accomplishment time distribution.

1.8 Conclusions

While the evolution of computing devices and analysis methods resulted in a sharp increase in the complexity of computable CTMC models, CTMC based analysis had been restricted to the analysis of stochastic models with expo-

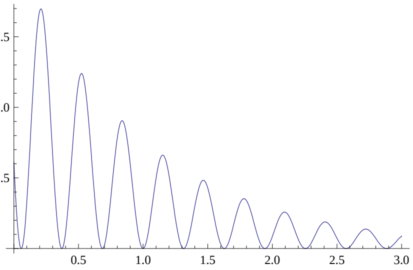

Fig. 1.5 Oscillating activity duration pdf.

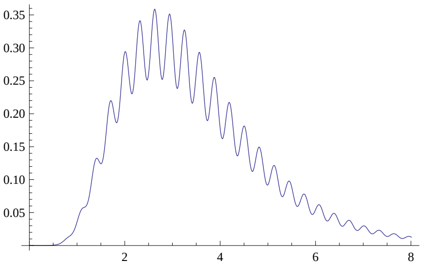

Fig. 1.6 Overall accomplishment time pdf.

nentially distributed duration times. A potential extension of CTMC based analysis is the inclusion of PH distributed duration times, which enlarges the state space, but still has a feasible computational complexity. We surveyed the basics of PH distributions and the analysis approach to generate the EMC.

A more recent development in this field is the extension of the EMC-based analysis with ME distributed duration times. With respect to the steps of the analysis method, the EMC based analysis and its extension with ME distributions are identical. However, because ME distributions are more flexible than the PH distributions (more precisely, the set of PH distributions of a given size is a subset of the set of ME distributions of the same size) this extension increases the modeling flexibility of the set of models which can be analyzed with a given computational complexity.

Apart of the steps of the EMC based analysis method we discussed the tool support available for the automatic execution of the analysis method. Finally, application examples demonstrate the abilities of the modeling and analysis methods.

References

1. S. Asmussen and M. Bladt. Point processes with finite-dimensional conditional probabilities. *Stochastic Processes and their Application*, 82:127–142, 1999.
2. N.G. Bean and B.F. Nielsen. Quasi-birth-and-death processes with rational arrival process components. *Stochastic Models*, 26(3):309–334, 2010.
3. P. Buchholz, A. Horvath, and M. Telek. Stochastic Petri nets with low variation matrix exponentially distributed firing time. *International Journal of Performability Engineering*, 7:441–454, 2011. Special issue on Performance and Dependability Modeling of Dynamic Systems.
4. P. Buchholz and M. Telek. Stochastic petri nets with matrix exponentially distributed firing times. *Performance Evaluation*, 67:1373 – 1385, 2010.
5. J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 1020 states and beyond. *Logic in Computer Science, 1990. LICS '90, Proceedings., Fifth Annual IEEE Symposium on e*, pages 428–439, Jun 1990.

6. G. Ciardo, G. Luttgen, and R. Siminiceanu. Saturation: an efficient iteration strategy for symbolic state space generation. In *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS), LNCS 2031*, pages 328–342. Springer-Verlag, 2001.
7. G. Ciardo, R. Marmorstein, and R. Siminiceanu. Saturation unbound. In *Proc. TACAS*, pages 379–393. Springer, 2003.
8. D. R. Cox. The analysis of non-markovian stochastic processes by the inclusion of supplementary variables. *Proceedings Cambridge Philosophical Society*, 51(3):433–441, 1955.
9. Salvatore Distefano, Francesco Longo, Marco Scarpa, and KishorS. Trivedi. Non-markovian modeling of a bladecenter chassis midplane. In *Computer Performance Engineering*, volume 8721 of *Lecture Notes in Computer Science*, pages 255–269. Springer International Publishing, 2014.
10. L. Kleinrock. *Queuing systems, Volume 1: Theory.* Wiley Interscience, New York, 1975.
11. V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems.* Chapman & Hall, 1995.
12. L. Lipsky. *Queueing Theory: A Linear Algebraic Approach.* Springer, 2008.
13. Francesco Longo and Marco Scarpa. Two-layer symbolic representation for stochastic models with phase-type distributed events. *International Journal of Systems Science*, 46(9):1540–1571, 2015.
14. A. Miner and D. Parker. Symbolic representations and analysis of large state spaces. In *Validation of Stochastic Systems*, LNCS 2925, pages 296–338, Dagstuhl (Germany), 2004. Springer.
15. A. S. Miner and G. Ciardo. Efficient reachability set generation and storage using decision diagrams. In *Application and Theory of Petri Nets 1999 (Proc. 20th Int. Conf. on Applications and Theory of Petri Nets*, pages 6–25. Springer-Verlag, 1999.
16. M. Neuts. Probability distributions of phase type. In *Liber Amicorum Prof. Emeritus H. Florin*, pages 173–206. University of Louvain, 1975.
17. A. Srinivasan, T. Ham, S. Malik, and R.K. Brayton. Algorithms for discrete function manipulation. *Computer-Aided Design, 1990. ICCAD-90. Digest of Technical Papers., 1990 IEEE International Conference on*, pages 92–95, Nov 1990.
18. K. Trivedi. *Probability & Statistics with Reliability, Queueing & Computer Science applications.* Prentice-Hall, 1982.