Phase-type Distributions

Philipp Reinecke, Levente Bodrog, and Alexandra Danilkina

Abstract Both analytical (Chapter ??) and simulation- and experimentation-based (Chapter ??) approaches to resilience assessment rely on models for the various phenomena that may affect the system under study. These models must be both accurate, in that they reflect the phenomenon well, and suitable for the chosen approach. Analytical methods require models that are analytically tractable, while methods for experimentation, such as fault-injection (see Chapter ??), require the efficient generation of random-variates from the models. Phase-type (PH) distributions are a versatile tool for modelling a wide range of real-world phenomena. These distributions can capture many important aspects of measurement data, while retaining analytical tractability and efficient random-variate generation. This chapter provides an introduction to the use of PH distributions in resilience assessment. The chapter starts with a discussion of the mathematical basics. We then describe tools for fitting PH distributions to measurement data, before illustrating application of PH distributions in analysis and in random-variate generation.

1 Introduction

Phase-type (PH) distributions are an often-used type of model for many phenomena in system evaluation, e.g. service-times, delays, and failure times. This chapter provides a gentle introduction to the theory of PH distributions and their application in common evaluation tasks. The typical workflow is shown in Figure 1: First, data has to be obtained for the phenomenon, e.g. the delays encountered in a system. Second, the data needs to be fitted by a phase-type distribution. This step is discussed in Section 3. The fitted distribution provides a model for the data which can be used in resilience evaluation. Phase-type distributions are equally well-suited for analytical approaches and for methods using simulation. We discuss their application both in matrix-analytic methods (Section 4) and in simulation (Section 5).

2 Mathematical Background

Continuous phase-type (PH) distributions represent the time to absorption in a Continuous-Time Markov Chain (CTMC) with one absorbing state [1]. PH distributions are commonly specified by a vector-matrix tuple $(\alpha, \mathbf{A})$, where

Philipp Reinecke Freie Universität Berlin, Institut für Informatik, Takustraße 9, 14195 Berlin, Germany, e-mail: philipp.reinecke@fu-berlin.de · Levente Bodrog Budapest University of Technology and Economics, Department of Telecommunications, 1521 Budapest, Hungary, e-mail: bodrog@webspn.hit.bme.hu · Alexandra Danilkina Freie Universität Berlin, Institut für Informatik, Takustraße 9, 14195 Berlin, Germany, e-mail: alexandra.danilkina@fu-berlin.de

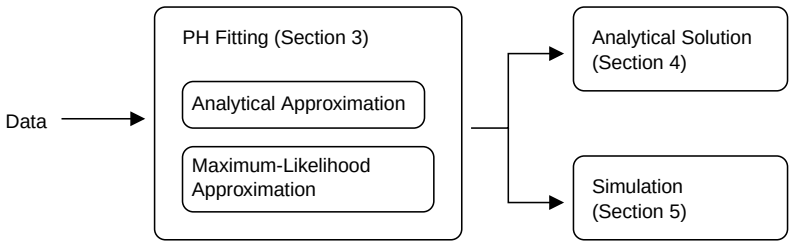


Fig. 1 Typical workflow when applying phase-type distributions in system evaluation.

$$\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{R}^n \text{ and } \mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \in \mathbf{R}^{n \times n}.$$

Definition 0.1. The *size* of the $(\alpha, \mathbf{A})$ representation is the size of the vector α, which is equal to the size of the square matrix $\mathbf{A}$.

Definition 0.2. The *probability density function (PDF)*, *cumulative distribution function (CDF)*, *Laplace-Stieltjes Transform (LST)* of the CDF and kth *moment*, respectively, are defined as follows [1, 2, 3]:

$$f(x) = \alpha e^{\mathbf{A}x}\mathbf{a}, \tag{1}$$
$$F(x) = 1 - \alpha e^{\mathbf{A}x}\mathbb{1}, \tag{2}$$
$$\tilde{F}(s) = \alpha_{n+1} + \alpha(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{a}, \tag{3}$$
$$E\left[X^k\right] = k!\alpha(-\mathbf{A})^{-k}\mathbb{1}. \tag{4}$$

where $\mathbf{a} = -\mathbf{A}\mathbb{1}$, and $\mathbb{1}$ is the column vector of ones of appropriate size. Note that phase-type distributions have rational LST and that the eigenvalues of the transient generator matrix are the poles of the LST of the distribution [4].

The vector-matrix representation of a PH distribution is not unique. In general, there exists another representation $(\beta, \mathbf{B})$ of size m that represents the same phase-type distribution. Representations may differ both in size ($n \neq m$) and in the contents of the tuples. In particular, every PH distribution has a *Markovian* representation $(\alpha, \mathbf{A})$, where $\alpha \geq 0$, $a_{ij} \geq 0, 1 \leq i \neq j \leq n$ and $\mathbf{a} = -\mathbf{A}\mathbb{1} \geq 0$. In this case, $\mathbf{A}$ is the transient part of the generator matrix of the associated CTMC

$$\overline{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{a} \\ \mathbf{0} & 0 \end{pmatrix}.$$

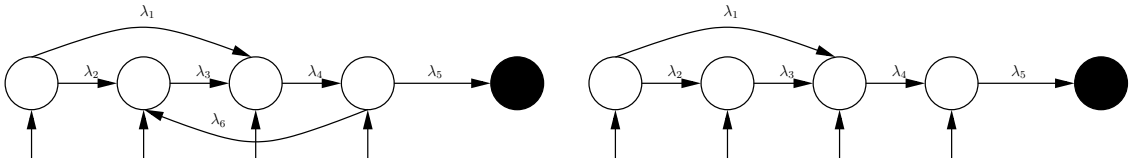
Another representation of the same size can be computed by a similarity transformation, as follows: When $\mathbf{B}$ is invertible and $\mathbf{B}\mathbb{1} = \mathbb{1}$, then $(\alpha\mathbf{B}, \mathbf{B}^{-1}\mathbf{A}\mathbf{B})$ is another representation of the same distribution, since its CDF is

$$1 - \alpha\mathbf{B}e^{\mathbf{B}^{-1}\mathbf{A}\mathbf{B}x}\mathbb{1} = 1 - \alpha\mathbf{B}\mathbf{B}^{-1}e^{\mathbf{A}x}\mathbf{B}\mathbb{1} = 1 - \alpha e^{\mathbf{A}x}\mathbb{1}.$$

The sizes of the $(\alpha, \mathbf{A})$ and the $(\alpha\mathbf{B}, \mathbf{B}^{-1}\mathbf{A}\mathbf{B})$ representations are the same in this case, but it is also possible to generate representations of the same distribution with another size, using a non-square matrix W.

2.1 PH Classes

Based on the structure of the underlying Markov chain, several classes of phase-type distributions can be distinguished. These classes differ in the statistical properties they can represent. Furthermore, the structure of a PH representation often has an impact on its application, as some structures allow more efficient solutions.

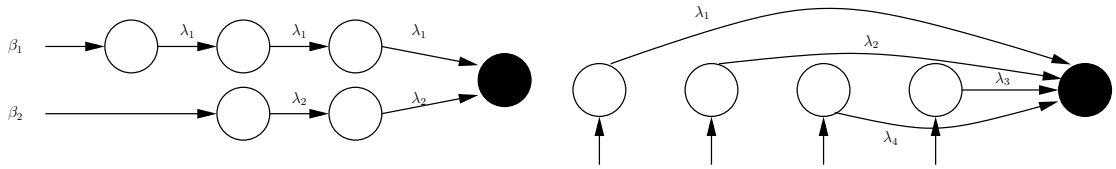


(a) General phase-type distribution containing cycles.

(b) Phase-type distribution without cycles (acyclic phase-type distribution).

Fig. 2 CTMC representations for general and acyclic phase-type distributions.

The most important distinction is the one into Acyclic and General Phase-type distributions: Every acyclic phase-type (APH) distribution has at least one Markovian representation without cycles in the sub-generator, while for general phase-type distributions cycles are allowed. This is illustrated in Figure 2: The distribution on the left contains a cycle, that is, a backward transition from state 4 to state 2. The distribution on the right does not contain this transition and therefore there are no cycles.



(a) Hyper-Erlang distribution with two branches.

(b) Hyper-exponential distribution with four branches.

Fig. 3 CTMC representations for Hyper-Erlang and hyper-exponential distributions.

Most approaches in fitting and application of PH distributions focus on the APH class, as this class offers better tractability than the general PH class. Within APH, we distinguish two important sub-classes: The first one is the class of Hyper-Erlang distributions (HErD). Hyper-Erlang distributions are mixtures of Erlang-distributions with different lengths and rates. They can be specified by a tuple (β, m, b, λ), where β is the vector of initial probabilities of each Erlang branch, m is the number of Erlang branches, b is the vector of the lengths of the Erlang branches, and λ is a vector containing the rates. The size of a Hyper-Erlang distribution is given by the sum of the lengths of the branches, i.e. $n = b\mathbb{1}$. The general structure is illustrated in Figure 3(a), where we show a hyper-Erlang distribution with $m = 2$ branches of length $b_1 = 3$ and $b_2 = 2$, respectively. The initial probabilities and the transition rates are given by $\beta = (\beta_1, \beta_2)$ and $\lambda = (\lambda_1, \lambda_2)$. The size of this representation is $n = b_1 + b_2 = 5$. One important example is the Erlang distribution, i.e. a Hyper-Erlang distribution with only one branch and initial probability $\beta_1 = 1$.

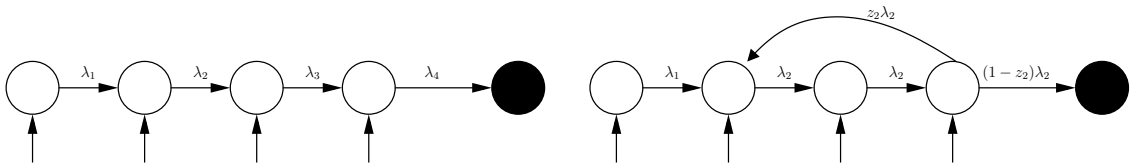
The second sub-class of APH we consider is the class of Hyper-Exponential distributions (HEx) of order n, specified by initial probability vector α and rate vector λ. Figure 3(b) shows an example for a hyper-exponential distribution of size $n = 4$. From this example, it is obvious that the hyper-exponential distributions are a subclass of the hyper-Erlang distributions, as every hyper-exponential distribution is a hyper-Erlang distribution with branch length vector $\mathbf{b} = \mathbb{1}$. Furthermore, setting $n = 1$ and $\alpha_1 = 1$ yields the exponential distribution with rate λ_1.

2.2 Canonical Representations

While in general representations for phase-type distributions are not unique, several canonical forms have been defined. For each PH distribution, the canonical form of a given size n is unique in the sense that there exists no representation of the same size n with the structure of the canonical form, but different parameters. Therefore, by comparing canonical forms, we can determine whether PH distributions given by different

representations are identical. More important, however, is the use of canonical forms in fitting, analysis, and simulation, where their typically low number of parameters and simple structure enable efficient methods.

In the following we discuss Cumani's Canonical Form 1 (CF-1) [5] and the Monocyclic form introduced in [6], as these are the most common ones.



(a) APH distribution in CF-1 form. (b) General phase-type distribution in monocyclic form.

Fig. 4 Canonical representations for phase-type distributions.

2.2.1 The Canonical Form for APH Distributions

The Canonical Form 1 (CF-1) was defined in [5]. The structure of its underlying CTMC is shown in Figure 4(a): The Markov chain can be entered at any state $i = 1 \ldots (n+1)$ (with probability α_i), but the absorbing state can only be reached by traversing all remaining states. Furthermore, the rates λ_i are ordered such that $\lambda_i \leq \lambda_{i+1}$ for all i. The formal definition is as follows:

Definition 0.3. [5] The *Canonical Form 1 (CF-1 form)* is a bi-diagonal representation (α, Λ) where α is Markovian and the rates a_i in Λ are in increasing order: $a_1 \leq a_2 \leq \cdots \leq a_n$.

[5, 7] showed that every acyclic phase-type distribution with a Markovian representation of size n has a unique CF-1 representation of the same size.[1] The CF-1 form for an APH given as $(\alpha, \mathbf{A})$ can be obtained by a similarity transformation. A procedure for constructing the similarity transformation matrix is given in [8].

Note that transforming an APH representation of size n to the CF-1 form considerably reduces the number of parameters: A general APH representation has n initial probabilities $\alpha_1, \ldots, \alpha_n$ and n^2 entries in the subgenerator matrix $\mathbf{A}$, i.e. the number of parameters is $n + n^2$. In the CF-1 form $\mathbf{A}$ is an upper bi-diagonal matrix with $a_{i,i} = -a_{i+1,i}$. The CF-1 form therefore has $2n$ parameters.

2.2.2 The Monocyclic Form for General PH Distributions

General PH distributions may have complex poles, and the poles of a PH distribution are given by the eigenvalues of the subgenerator matrix $\mathbf{A}$. As the eigenvalues of a bi-diagonal representation (α, Λ) are equal to the entries of the diagonal and $\mathbf{A} \in \mathbf{R}^{n \times n}$ it is easy to see that a bi-diagonal structure like the CF-1 form cannot represent phase-type distributions with complex poles.

For this reason, [6] proposed the Monocyclic form as a chain of Feedback-Erlang (FE) blocks, defined as follows:

Definition 0.4. A *Feedback-Erlang (FE) block* is given by a tuple (b, λ, z) of the length b, transition rate λ, and feedback probability $z \in [0:1)$. The Feedback-Erlang block consists of an Erlang-distribution with length b and rate λ and an additional (feedback) transition from the last state of the block to the first state.

Figure 5 illustrates this concept. Note that the cases $z = 0$ and $b = 1$ are allowed. For $z = 0$, the Feedback-Erlang is simply an Erlang of order b, while for $b = 1$ it is an exponential distribution. The importance of

[1] Smaller CF-1 representations may exist if there is redundancy in the original representation [1, 8, 9].

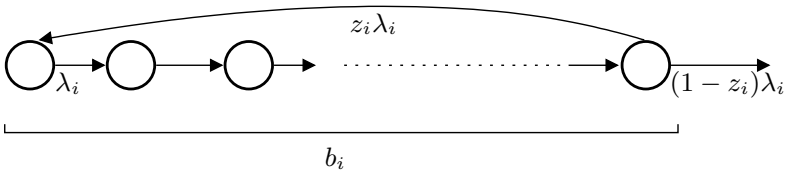


Fig. 5 Structure of a Feedback-Erlang block.

this structure lies in the fact that for $z > 0$ and $b > 1$ the block has a conjugate-complex pair of eigenvalues [6]. Therefore, a chain of FE blocks can be used to represent the complex eigenvalue pairs of a general phase-type distribution.

Based on this observation, [6] define the Monocyclic representation as a chain of Feedback-Erlang blocks:

Definition 0.5. A *Monocyclic representation* is given by the tuple $(\alpha, m, b, \lambda, z)$, where the vector $\alpha \in \mathbf{R}^{b\mathbb{1}}$ specifies the initial state probabilities, and b, λ and z define the length, rate, and feedback probability of the m Feedback-Erlang blocks.

The FE blocks are positioned such that the absolute values of the dominant eigenvalues r_i are in ascending order: $r_i \leq r_{i+1}$.

Any PH distribution has a monocyclic representation [6]. If the representation of the PH distribution is PH-simple [7] and of size n, then the size of the monocyclic representation is $n' \geq n$. This potential size expansion makes the monocyclic representation less efficient in analytical studies, but its simple and still Markovian structure makes it promising for simulation studies.

The structure of a Monocyclic representation is shown in Figure 4(b). Note that if $z_i = 0$ for all FE blocks $i = 1, \ldots, b$ the Monocyclic form is equivalent to the CF-1 form. That is, the CF-1 form is actually a special case of the Monocyclic form.

2.3 Properties

One nice property of the PH distributions class is that it is closed for minimum/maximum, summation, etc. From the point of view of applying PH distributions to fitting data, the main problem of the class is the incomplete exploration of the moment bounds of the general PH distribution. However, there are results on them for several particular cases.

The feasible first moment range of the PH class is the set of non-negative numbers as PH gives a non-negative random variable.

It is proven that the feasible range of the squared coefficient of variation for the PH of size n (PH(n)) is

$$cv^2 \geq \frac{1}{n}, \tag{5}$$

where the equality holds for the n dimensional Erlang distribution (Erl(n)).[10]

For higher moments there is no general knowledge, however there are several special cases for which some insights on the moment bounds exist, like e.g., the moment bounds of the APH$(2) \equiv$ PH(2) class [11], the moment bounds of the PH(3) class implied by the canonical form given in [12], the bound of the general APH class within the PH is known according to the APH canonical form and there exists also a numerical method to determine the general PH bound in [13].

From the fitting perspective the *reduced moment problem* (when a distribution function is determined based on its moments) can also be crucial which is only solved for the wider class of distributions the matrix exponentials.[14]

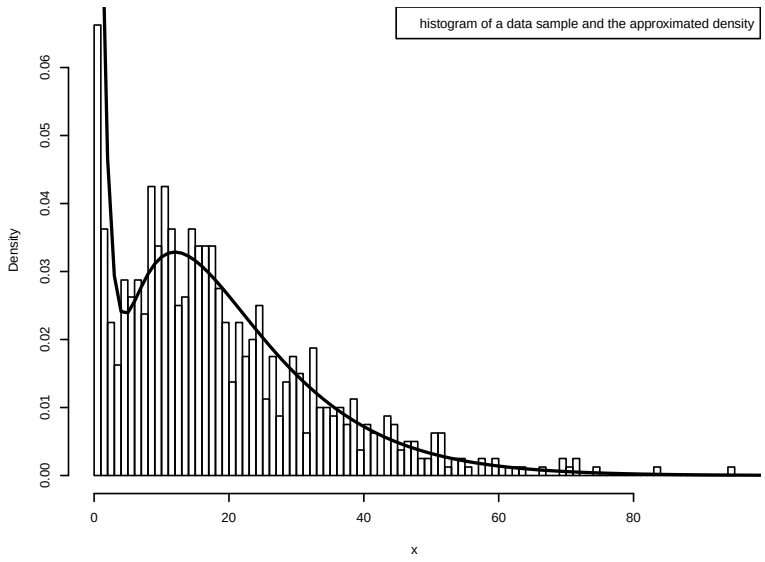


Fig. 6 Example data and its approximation with a phase-type distribution.

Table 1 Performance measures defined in [15]

Performance Measure	Definition
Area difference between distribution functions ΔF	$\Delta F = \int_0^\infty \lvert\hat{F}(x) - F(x)\rvert dt$
Area difference between densities Δf	$\Delta f = \int_0^\infty \lvert\hat{f}(x) - f(x)\rvert dt$
Relative error in the first moment (mean c_1)	$e_1 = \frac{\lvert\hat{c_1} - c_1\rvert}{c_1}$
Relative error in the second central moment (variance c_2)	$e_2 = \frac{\lvert\hat{c_2} - c_2\rvert}{c_1}$
Relative error in the third central moment (skewness c_3)	$e_3 = \frac{\lvert\hat{c_3} - c_3\rvert}{c_3}$

3 Fitting Measurement Traces With PH Distributions

As illustrated in Figure 1, the first step in applying phase-type distributions in resilience evaluation is to fit a PH distribution to a data set. Consider Figure 6, where we show both a histogram of some data and the density of a phase-type distribution approximating the data. Our aim is to approximate the data as closely as possible, in order to obtain correct results when using the approximating distribution later on. In this section we provide the basics for fitting data sets with phase-type distributions. We discuss costs, quality metrics, and introduce three established fitting tools.

3.1 Costs of Fitting PH Distributions to Data

Since a PH distribution is defined by the tuple $(\alpha, \mathbf{A})$, the problem of fitting translates to finding an initial probability vector α and a sub-generator matrix $\mathbf{A}$ of appropriate size n. While, in general, higher-order PH distributions can provide a better approximation [16], they are more expensive in both analysis and simulation. Furthermore, the time required for fitting a distribution increases with n, as more parameters have to be fitted. Consequently, careful choice of n is important.

As will be shown in Sections 4 and 5, the cost of using a PH distribution depends not only on the size n, but also on the structure of the representation. The same holds for the fitting problem. Here, the number of free parameters to be fitted can be reduced significantly by choosing an appropriate representation: If we assume the size n of the representation to be constant, then general phase-type distributions in an arbitrary Markovian representation have $n + n^2$ free parameters, as α is a row vector of length n, and $\mathbf{A}$ is a matrix

of size $n \times n$. If we assume that the representation is Monocyclic, we have a chain of m Feedback-Erlang blocks, each with a length parameter b_j, rate parameter λ_j and feedback probability z_j, and an initial probability vector of size n. As $m \leq n$, the upper limit for the number of free parameters is $3n+n$. Limiting ourselves to the APH class, we can utilise the CF-1 canonical form, which has only $2n$ free parameters: n transition rates and n initial probabilities. Finally, if we consider only HErD distributions in representations as shown in Figure 3(a), the number of free parameters reduces to $3m$: m initial probabilities for the m Erlang branches, m lengths for the Erlang branches, and m transition rates.

3.2 Quality Measures

Fitting a phase-type distribution to data requires careful choice of the right fitting tool, as well as of fitting parameters such as sub-class and size. As just discussed, the approximation problem becomes less complex if data is fitted with subclasses of phase-type distributions, however, fitting quality may decrease as well, as subclasses cannot represent all properties of the general PH class. For example, hyperexponential distributions cannot approximate distributions with oscillating densities [17].

In order to assess the quality of data approximation, quality measures are required. An intuitive method consists in simply comparing the shape of the empirical PDF or CDF to that of the approximating PH distribution. This gives not only absolute differences, but also gives a visual impression how well the shape of the empirical PDF/CDF was approximated (e.g. in Figure 6 the approximated density fits the data quite well).

While a visual impression often yield a good initial assessment, a more formal approach requires exact definitions of quality measures. Table 1 shows the standard quality measures for PH fitting, as defined in [15]. The first two performance measures formalise the visual comparison of empirical and approximated data, by computing the distance between both curves. The last three measures capture how well the fitted distribution approximates the empirical moments of the data. Based on these performance measures we can decide which tool to use, and which fitting is most appropriate for the requirements and future application of approximation results. For instance, for use in a stochastic model whose behaviour primarily depends on the first three moments, one would aim to get small relative moment errors, while in other applications fitting the shape of the density may be more important.

3.3 Introduction to PH-Fitting Tools

Here we outline three tools for data approximation with phase-type distributions: Moment Matching, G-FIT and PhFit. They mainly differ with respect to the algorithms they employ and the subclass of PH distributions they support. There are two general and relevant classes of algorithms: Analytical and statistical methods, where the former relies on direct computation of the parameters and the latter is based on the maximum likelihood method for parameter estimation, typically implemented as an iterative procedure.

3.3.1 Analytic Approximation: Moment Matching

Analytic moment-matching methods have the advantage of being fast, easy to implement, and giving low errors in the moments. On the other hand, accuracy of the fitting may be limited by the representation. We illustrate this using the method proposed in [3], which can fit an APH(2) distribution to the first three moments of a data set. The approach proceeds by computing the approximation parameters directly from the moments, as follows: An APH(2) in CF-1 form with $\alpha = (\alpha_1, 1-\alpha_1)$ and

$$\mathbf{A} = \begin{pmatrix} -a_1 & a_1 \\ 0 & -a_2 \end{pmatrix},$$

is defined by three parameters, $a_1, a_2,$ and α_1. Recall from Definition 0.2 the general moments-generating function for a PH distribution. Writing the first three moments explicitly:

$$E[X] = m_1 = \frac{a_1 + \alpha_1 a_2}{a_1 a_2},$$
$$E[X^2] = m_2 = \frac{2(a_1^2 + \alpha_1 a_1 a_2 + \alpha_1 a_2^2)}{a_1^2 a_2^2},$$
$$E[X^3] = m_3 = \frac{6(a_1^3 + \alpha_1 a_1^2 a_2 + \alpha_1 a_1 a_2^2 + \alpha_1 a_2^3)}{a_1^3 a_2^3},$$

[3] obtain a system of 3 linear equations. Solving this system for a_1, a_2, α_1 yields an APH(2) that matches the first three moments. However, possible solutions are limited by the moment bounds for the APH(2) class (cf. Section 2.3). For combinations of moments outside the moment bounds, the system has no solution, i.e. data sets with these moments cannot be fitted exactly by an APH(2). For instance, as follows from (5), the smallest SCV cv^2 that can be represented by an APH(2) is

$$cv^2 = \frac{1}{2},$$

which puts constraints on the relation of the mean and variance. Data sets with $cv^2 < \frac{1}{2}$ require PH distributions of higher order. Similar constraints exist for the third moment, although in some cases the third moment can be approximated even when no exact fitting is possible.

3.4 Maximum Likelihood Method

The second class of fitting algorithms is based on the maximum-likelihood method. The maximum-likelihood method is a common tool for parameter estimation [18]. Using Bayes' rule, the approach identifies the parameter set Θ that is the most probable estimation for the observations, using the likelihood function

$$L(\Theta|x) = P(x|\Theta).$$

When fitting PH distributions to the data set x, the parameter set is $\Theta = (\alpha, \mathbf{A})$. The general approach to maximum-likelihood parameter estimation is the Expectation-Maximisation (EM) algorithm [18]:

1. Choose initial values for the parameter set Θ.
2. E-Step (Estimation): Using the data x and likelihood-function, estimate the set of parameters Θ.
3. M-Step: (Maximisation) Pick the set Θ from the estimated values calculated in step 2 that maximises the likelihood-function.
4. If the abort criterion is not fulfilled, go to step 2, otherwise stop.

The EM algorithm is an iterative procedure, alternating between the E-Step (Expectation) and the M-Step (Maximization). The abort criterion for step 4 can be defined as a fixed maximum value for the likelihood function or a mimimum difference in likelihood values for consecutive iterations. Note that for multiple optima of the likelihood function the EM algorithm may stop at a local optimum, depending on the initial values.

The maximum likelihood method for PH fitting has the advantage of providing more flexibility than analytical moment-matching methods. On the other hand, the iterative fitting process is usually slower than the analytical approach. In the following we discuss two tools whose fundamental method is the maximum likelihood approach.

3.4.1 G-FIT for fitting Hyper-Erlang distributions

The G-FIT tool [17] approximates data using Hyper-Erlang distributions. Recall that the number of transition rates and the size of the initial vector of a Hyper-Erlang distribution only depend on the number of Erlang branches. This enables an efficient fitting method: Once the number m and length b of the Erlang branches has been set, the parameters are

$$\Theta = (\beta, \lambda).$$

In each iteration the EM algorithm estimates β and λ which maximise the likelihood function. G-FIT provides convergence checks based on the maximal change in Θ and on the relative differences of the log-likelihood between successive iterations.

The user may specify the number and length of Erlang branches prior to fitting or let G-FIT determine an optimal size. In the first case the user has to set a number of Erlang branches and their length. The second option is more general and is useful for the unexperienced user. It requires as input only a number of phases for the resulting distribution. G-FIT will then estimate optimal number of Erlang branches and their parameters, by trying all possible combinations.

G-FIT expects an input as a text file containing the data set. The first line should be a number of data points in the data set followed by data points themselves, which are given one per line. The output is also a text file, containing the number of Erlang branches, number of phases, initial probabilities and transition rates for each Erlang branch.

3.4.2 PhFit

The PhFit tool [2] approximates data using acyclic phase-type distributions in CF-1 form. One major advantage is that the user can choose between different distance measures for the EM algorithm, in order to obtain an optimal fitting. The distance measures supported by PhFit are the relative entropy, PDF area distance, and CDF area distance, defined as

$$\int_0^\infty f(t)\log(\frac{f(t)}{\hat{f}(t)})dt, \int_0^\infty |\hat{F}(x) - F(x)|dt, \text{ and } \int_0^\infty |\hat{f}(x) - f(x)|dt, \text{ respectively,}$$

where $f(t)$ denotes the probability density function (PDF) of the original distribution and $\hat{f}(t)$ the PDF of the approximating distribution, $F(t)$ the cummulative distribution function (CDF) of the original distribution and $\hat{F}(t)$ the CDF of the fitted distribution. Among the fitting tools we discuss, PhFit is the only one with a graphical user interface. This feature is beneficial for finding appropriate fitting parameters and evaluation of results.

PhFit computes optimal values for distribution parameter $(\alpha, \mathbf{Q})$ starting with special initial values $(\alpha^{(0)}, \mathbf{Q}^{(0)})$ according to the distance measure. PhFit picks optimal values from 1000 random generated pairs of vectors. The distance measure defines the optimality criterion. The likelihood optimization problem is solved by using the iterative linearization method. The direction for optimization of the distance measure is determined by simplex algorithm. The algorithm stops computation once the relative difference between

$$(\alpha^{(i-1)}, \mathbf{A}^{(i-1)})$$

and

$$(\alpha^{(i)}, \mathbf{A}^{(i)})$$

for iteration i is less than the predefined value or once the number of maximum number of iterations is reached.

PhFit provides separate fitting for body and tail. The body is the part of distribution with the most mass, whereas the tail represents rare data points. The user can choose the boundary where the tail begins. The tail will be approximated with the heuristic method, that determines parameters for the mixture of m geometric- for discrete APH- and m exponential- for continious APH-distributions. Having parameters for

hyper-geometric or hyper-exponential distribution leads to the body fitting as described before. The result distribution is structured by the CF-1 and the mixtures.

PhFit requires as input a text file containing the data in ascending order. The output consists of the initial probability vector α and the diagonal of the subgenerator matrix. Note, however, that in contrast to the definition we gave in Definition 0.3, PhFit considers the 0th state to be absorbing, instead of state $(n+1)$. That is, the output of PhFit is reversed, compared to the notation used throughout this chapter.

4 Phase-type Distributions in Model Analysis: Matrix Analytic Methods

Figure 1 shows two ways of using a PH distribution in an evaluation. The memoryless property of the Markov chains allows the matrix representation of the phase-type distributions as it is given in Section 2. The matrix representation and accordingly the simple analytical formalism to define the properties of the PH distributions makes them very popular among researchers both for modeling (Section 3) and simulation (Section 5). Furthermore in the case of complex systems it turns out that the matrix representation of the PH distributions allows the use of the matrix analytic methods [19] in case of large Markov chains.

The matrix analytic methods utilize the structure of the Markov chain which, in this chapter, is two-dimensional. Both dimensions have their own characteristic. The first dimension represents the – usually finite – number of phases ($J(t)$) of the process. The second dimension is the infinite counting process ($N(t)$). This results in an infinite, but well-structured, Markov chain on the block level where the blocks describes the phase either with or without arrival. The same block structure appears also in the generator matrix of the Markov chain which can be upper block-bidiagonal or tridiagonal in our cases.

The examples of this section shows how the matrix analytic methods utilize the analytic given PH properties during the solution of complex Markov models. The result can be either the short-term or the steady-state behavior. The methods also allow to find the solution of infinite models by solving finite problems.

4.1 Processes with PH marginal distribution

Processes play an important role in stochastic modeling thus it comes natural to propose the process with PH marginal distribution. Here we investigate both the independent identical distributed (iid) and the correlated arrival process with PH marginal. These are the PH renewal process and the Markov arrival process (MAP) respectively.

4.1.1 PH renewal process

Given a phase-type distribution represented by the initial vector, generator matrix pair $(\alpha, \mathbf{A})$ – denoted as $\mathrm{PH}(\alpha, \mathbf{A})$ – it is the marginal distribution of the PH renewal process defined by the generator matrix

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A} & \mathbf{a}\alpha & 0 & \ldots \\ 0 & \mathbf{A} & \mathbf{a}\alpha & 0 \\ 0 & 0 & \mathbf{A} & \mathbf{a}\alpha \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}, \tag{6}$$

where $\mathbf{a} = -\mathbf{A}\mathbb{1}$ is the vector of absorption of the marginal distribution if $\mathbb{1}$ is the column vector of ones. The diagonal block describes the phase transitions of the PH marginal and the block in the upper co-diagonal describes the phase transitions belonging to the renewal instances. The graph of the corresponding continuous time Markov chain (CTMC) is depicted in Figure 7.

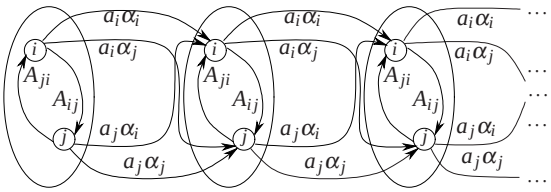


Fig. 7 The graph of the PH renewal process

The product in the upper co-diagonal blocks expresses that the initial distribution of the next interarrival is always the same (α) after arrival ("absorption" in the PH marginal) regardless of any of the other interarrivals, i.e., the process is uncorrelated.

The generator matrix of the phase process is $\mathbf{H} = \mathbf{A} + \mathbf{a}\alpha$. The steady state phase distribution (π) is the solution of the linear system of equations

$$\begin{aligned} \pi\mathbf{H} &= 0 \\ \pi\mathbb{1} &= 1. \end{aligned} \tag{7}$$

The transient phase distribution is

$$\pi(t) = \pi(0)\mathrm{e}^{\mathbf{H}t} \tag{8}$$

which is a vector of elements $\pi_i(t) = \Pr(J(t) = i)$ giving the probability that the process is in phase i at time t. Using the transient phase behavior the remaining time to the next arrival, at time t, is $\mathrm{PH}(\pi(t), \mathbf{A})$.

Let $\pi(n,t) = (\Pr(N(t) = n, J(t) = j))$ be the number of arrival (n) and the phase (j) distribution at time t. With initial conditions $\pi(0,0) = \alpha$ and $\pi(i,0) = 0, (i > 0)$ the transient number of arrivals is given by the differential equation

$$\frac{d\pi(i,t)}{dt} = \pi(i,t)\mathbf{A} + \pi(i-1,t)\mathbf{a}\alpha \tag{9}$$

and its z-transform, with initial condition $\pi(z,0) = \alpha$, is

$$\frac{d\pi(z,t)}{dt} = \pi(z,t)\mathbf{A} + z\pi(z,t)\mathbf{a}\alpha = \pi(z,t)\left(\mathbf{A} + z\mathbf{a}\alpha\right). \tag{10}$$

The solution of the differential equation, i.e., the transient distribution of the number of arrivals, is

$$\pi(z,t) = \alpha\mathrm{e}^{(\mathbf{A}+z\mathbf{a}\alpha)t}. \tag{11}$$

4.1.2 Markov arrival process

Compared to the PH renewal process the Markov arrival process (MAP) is the correlated arrival process with PH marginal distribution, i.e., the phase distribution is not restricted to be the same after every arrival – it is arbitrary. Its two dimensional CTMC is also defined by the phase process $J(t)$, describing the phase of the marginal distribution, and by the counting process $N(t)$, meaning the number of arrivals. Its graph is depicted in Figure 8 and its generator matrix is

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{D}_1 & 0 & \ldots \\ 0 & \mathbf{D}_0 & \mathbf{D}_1 & 0 & . \\ 0 & 0 & \mathbf{D}_0 & \mathbf{D}_1 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}, \tag{12}$$

where the Markov arrival process is represented by $\mathbf{D}_0$ – the phase transitions without arrival – and $\mathbf{D}_1$ – the phase transitions with one arrival. Such a MAP is denoted as $\mathrm{MAP}(\mathbf{D}_0, \mathbf{D}_1)$.

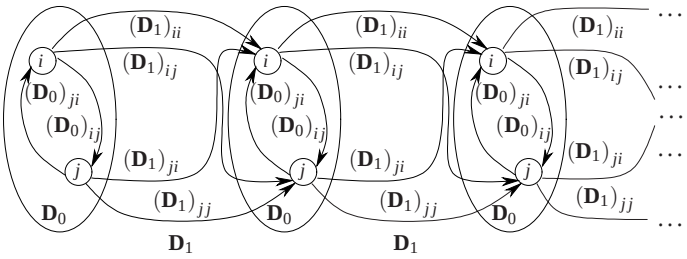


Fig. 8 The graph of the Markov arrival process

The interarrival times of the $\mathrm{MAP}(\mathbf{D}_0, \mathbf{D}_1)$ are $\mathrm{PH}(\alpha_0, \mathbf{D}_0)$, $\mathrm{PH}(\alpha_1, \mathbf{D}_0)\ldots$ The – correlated – phase distribution embedded at arrival instances forms a discrete time Markov chain (DTMC) with state transition probability matrix $\mathbf{P} = (-\mathbf{D}_0)^{-1}\mathbf{D}_1$.

The joint probability density function of the interarrival times, X_0 and X_k, is

$$f_{X_0,X_k}(x_0, x_k) = \pi \mathrm{e}^{\mathbf{D}_0 x_0}\mathbf{D}_1\mathbf{P}^{k-1}\mathrm{e}^{\mathbf{D}_0 x_k}\mathbf{D}_1\mathbb{1}, \tag{13}$$

where π is the embedded stationary phase distribution at arrival instances, i.e., it is the solution of the linear system of equations

$$\begin{aligned} \pi\mathbf{P} &= \pi \\ \pi\mathbb{1} &= 1. \end{aligned} \tag{14}$$

The stationary interarrival time distribution is $\mathrm{PH}(\pi, \mathbf{D}_0)$ with nth moment

$$E[X^n] = n!\pi(-\mathbf{D}_0)^{-n}\mathbb{1} \tag{15}$$

and the joint moment of two interarrivals is

$$\begin{aligned} E[X_0X_k] &= \int_{x_0}\int_{x_k} x_0x_k\pi\mathrm{e}^{\mathbf{D}_0x_0}\mathbf{D}_1\mathbf{P}^{k-1}\mathrm{e}^{\mathbf{D}_0x_k}\mathbf{D}_1\mathbb{1}\,dx_0dx_k \\ &= \pi(\mathbf{D}_0)^{-1}\mathbf{P}^k(\mathbf{D}_0)^{-1}\mathbb{1}. \end{aligned} \tag{16}$$

The covariance of two interarrivals is

$$\mathrm{cov}(X_0, X_k) = \mathrm{E}[X_0X_k] - \mathrm{E}^2[X] \tag{17}$$

and using (15), (16) and (17) the lag k correlation of the MAP is

$$\mathrm{corr}(X_0, X_k) = \frac{\mathrm{cov}(X_0, X_k)}{\mathrm{E}[X^2] - \mathrm{E}^2[X]}. \tag{18}$$

4.2 The quasi birth-death process

The quasi birth-death (QBD) process [19, 1] is also defined by the phase process ($J(t)$) and the counting precess ($N(t)$). But in case of the QBD process the counting, or the "level", process is allowed to be

decreased by one as well as to stay on the same level or to be increased by one. It is thus the "multiphase" extension of the birth death precess which is for example the solution of the M/M/1 queueing system. The generator matrix of the QBD process has block-tridiagonal form

$$\mathbf{Q} = \begin{pmatrix} \mathbf{L}' & \mathbf{F} & 0 & \dots \\ \mathbf{B} & \mathbf{L} & \mathbf{F} & 0 & . \\ 0 & \mathbf{B} & \mathbf{L} & \mathbf{F} & 0 \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}, \tag{19}$$

where the blocks or level transition matrices are

L′ local state transitions inside the first – irregular – block,
B backward (level) state transitions,
L local state transitions on the regular levels and
F forward (level) state transitions.

The graph of the QBD is depicted in Figure 9.

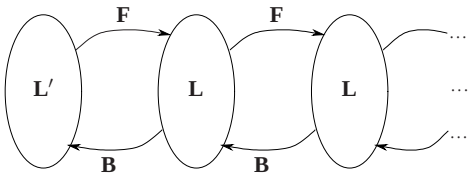


Fig. 9 The graph of the quasi birth-death process

We give the solution method of the QBD through the analysis of the MAP/PH/1 queueing system with arrival process MAP$(\mathbf{D}_0, \mathbf{D}_1)$ and service time PH$(\alpha, \mathbf{A})$. The level transition matrices are

$$\begin{aligned} \mathbf{L}' &= \mathbf{D}_0 \otimes \mathbf{I} \\ \mathbf{B} &= \mathbf{I} \otimes \mathbf{a}\alpha \\ \mathbf{L} &= \mathbf{D}_0 \oplus \mathbf{A} \\ \mathbf{F} &= \mathbf{D}_1 \otimes \mathbf{I}, \end{aligned}$$

where $\mathbf{a} = -\mathbf{A}\mathbb{1}$ and $\mathbf{I}$ is the appropriate size identity matrix. The operators $\otimes$ and $\oplus$ are the Kronecker product and sum respectively.

The generator matrix of the phase process is $\mathbf{H} = \mathbf{B} + \mathbf{L} + \mathbf{F}$ and if it is irreducible then the steady state phase distribution is the solution of the linear system of equations

$$\begin{aligned} \pi\mathbf{H} &= 0 \\ \pi\mathbb{1} &= 1. \end{aligned} \tag{20}$$

The QBD process is stable if its stationary drift is less than zero

$$d = \pi\mathbf{F}\mathbb{1} - \pi\mathbf{B}\mathbb{1} < 0. \tag{21}$$

The steady state solution of the QBD is the solution of the infinite system of linear equations

$$\begin{aligned} \nu\mathbf{Q} &= 0 \\ \nu\mathbb{1} &= 1. \end{aligned} \tag{22}$$

Partitioning ν according to the blocks of $\mathbf{Q}$ is

$$\nu = \begin{pmatrix} \nu_0 & \nu_1 & \nu_2 & \ldots \end{pmatrix}$$

and substituting the partitions into (22) we get

$$\nu_0 \mathbf{L}' + \nu_1 \mathbf{B} = 0 \tag{23}$$

and

$$\nu_{i-1}\mathbf{F} + \nu_i \mathbf{L} + \nu_{i+1}\mathbf{B} = 0 \quad \forall i \geq 1. \tag{24}$$

Assuming that the Markov chain is irreducible $\nu_i = \nu_{i-1}\mathbf{R} = \nu_0 \mathbf{R}^i \quad (\forall i)$, i.e., its solution is the matrix geometric distribution, the general equation (24) can be rewritten as

$$\begin{aligned} \nu_0 \mathbf{R}^{i-1}\mathbf{F} + \nu_0 \mathbf{R}^i \mathbf{L} + \nu_0 \mathbf{R}^{i+1}\mathbf{B} &= 0 \\ \nu_0 \mathbf{R}^{i-1}\left(\mathbf{F} + \mathbf{R}\mathbf{L} + \mathbf{R}^2\mathbf{B}\right) &= 0 \end{aligned}$$

with a solution determined by

$$\mathbf{F} + \mathbf{R}\mathbf{L} + \mathbf{R}^2\mathbf{B} = 0. \tag{25}$$

If the QBD is stable there is one of the solutions of $\mathbf{R}$ whose eigenvalues are within the unit circle on the complex plane.

As all the eigenvalues of the relevant $\mathbf{R}$ is within the unit circle there exists the limit of the sum $\sum_{i=0}^{\infty} \mathbf{R}^i = (\mathbf{I} - \mathbf{R})^{-1}$. Using the convergence the normalizing condition of ν can be expressed as

$$\nu \mathbb{1} = \sum_{i=0}^{\infty} \nu_i \mathbb{1} = \sum_{i=1}^{\infty} \nu_0 \mathbf{R}^i \mathbb{1} = \nu_0 \sum_{i=1}^{\infty} \mathbf{R}^i \mathbb{1} = \nu_0 (\mathbf{I} - \mathbf{R})^{-1} \mathbb{1} = 1. \tag{26}$$

Now substituting $\mathbf{R}$ into (23) and using (26) we have a linear system of equations

$$\begin{aligned} \nu_0 \left(\mathbf{L}' + \mathbf{R}\mathbf{B}\right) &= 0 \\ \nu_0 (\mathbf{I} - \mathbf{R})^{-1} \mathbb{1} &= 1 \end{aligned} \tag{27}$$

for the zeroth block of ν. All the other blocks can be calculated using ν_0 and $\mathbf{R}$ as

$$\nu_i = \nu_0 \mathbf{R}^i, \quad \forall i. \tag{28}$$

By these considerations the infinite problem of solving the QBD in (22) is reduced to be the solution of the finite problems in (25), (27) and (28). By this reduction the matrix analytic methods indirectly allows the utilization of the PH distribution in the solution of infinite systems.

5 Phase-type Distributions in Random-Variate Generation

While phase-type distributions enable efficient solutions for analytical models, they have applications beyond analytical approaches. For instance, PH distributions can be used to simulate realistic service-times in models that cannot be solved by analytical methods and thus require simulation, and for generating delays in test-beds. These applications require the efficient generation of random variates from phase-type distributions.

Phase-type distributed samples may be generated by playing the CTMC until absorption, and by numerical inversion of the distribution function [20]. In the following we focus on methods that 'play' the CTMC. Note that these methods require the Markovian representation.

The methods discussed in the following utilise random variates from the uniform, exponential, Erlang, and geometric distributions. We assume that random variates with uniform distribution on $(0,1)$ are given, and denote these by U. Using the inversion method, a sample with exponential distribution with rate λ is then drawn by

$$\mathrm{Exp}(\lambda) = -\frac{1}{\lambda}\ln(U).$$

A sample from the Erlang distribution with degree b and rate λ is generated by

$$\mathrm{Erl}(b,\lambda) = -\frac{1}{\lambda}\ln\left(\prod_{i=1}^{b} U_i\right).$$

Note that this way of sampling $\mathrm{Erl}(b,\lambda)$ is more efficient than the functional equivalent of drawing b exponentially distributed samples and summing them up, because the ln operation is applied only once. Finally, a sample from the geometric distribution (starting from 0) with parameter p is obtained by

$$\mathrm{Geo}(p) = \left\lfloor \frac{\ln(U)}{\ln(p)} \right\rfloor.$$

The most natural way to generate a PH-distributed sample by playing the CTMC proceeds as follows: First, we select a state i by drawing an integer sample distributed according to the initial probability vector α. Afterwards, in each step the next state is selected according to the next-state probability vector. The sojourn time for state i is obtained as a sample from the exponential distribution with rate $-\lambda_{ii}$. Letting e_i denote the row vector with 1 at position i, and 0 everywhere else, the `Play` method can be given in pseudocode as follows:

`Procedure Play:`

1) $x := 0$. Draw an α-distributed discrete sample i for the initial state.
2) The chain is in state i
   - draw an $e_i(-\mathrm{diag}\langle 1/a_{ii}, 0\rangle\overline{\mathbf{A}} + \mathbf{I})$-distributed discrete sample for the next state,
   - $x \mathrel{+}= \mathrm{Exp}(-a_{ii})$,
   - if the next state is the absorbing one ($i = n+1$) go to 3), otherwise go to 2)
3) Return x.

In [21], Neuts and Pagano observe that when traversing a state more than once, the `Play` method adds up multiple samples from the same exponential distribution. The sum of k_i exponential distributions of the same rate λ_{ii}, however, is the Erlang distribution with length k_i and rate λ_{ii}. As shown above, drawing a sample from the Erlang distribution of length k_i requires only one logarithm operation, as opposed to k_i logarithms when drawing individual exponential samples. Thus, Neuts and Pagano propose the following method, which, instead of drawing exponential samples for each visit to a state i, counts the number of visits and then draws one Erlang-distributed sample for each state:

`Procedure Count:`

1) $x := 0$, $k_i := 0$, $(i = 1,..,n)$, Draw an α-distributed discrete sample i for the initial state.
2) The chain is in state i
   - k_i += 1,
   - draw an $e_i(-\mathrm{diag}\langle 1/a_{ii}, 0\rangle\overline{\mathbf{A}} + \mathbf{I})$-distributed discrete sample for the next state,
   - if the next state is the absorbing one go to 3) otherwise to 2)

3) for $i = 1, \ldots, n$; do x += Erl$(k_i, -a_{ii})$; done
4) Return x.

If the distribution is in Monocyclic form, we can derive another method from the structural properties of the Monocyclic representation. Recall that this representation consists of a chain of Feedback-Erlang blocks. With such a chain, possible state transitions are predetermined by the structure in two ways: First, when we leave a Feedback-Erlang block j, the next state will be the first state of the next Feedback-Erlang block $j+1$. This implies that no new sample is required for choosing the successor block. Second, recall from Figure 5 that each FE block consists of a chain of $m_j - 1$ states with exactly one outgoing transition (to the next state), and only one state with two outgoing transitions (the feedback state). Thus, within each FE block the only state where the next state is not determined by the structure is the last one. Furthermore, as the last state has only two outgoing transitions, the choice of staying within block j or entering the next block $j+1$ corresponds to a Bernoulli experiment with parameter z_j. Consequently, the number of 'loops' in each block follows a geometric distribution with parameter z_j. Therefore, in order to generate the sample corresponding to the jth Feedback-Erlang block, we add a geometrically distributed number of exponentially distributed random variates with the same rate λ_j. As discussed when introducing the `Count` method, an efficient way of doing this is to draw a sample from an Erlang distribution of the appropriate length. These considerations lead to the following method:

`Procedure Monocyclic:`

1) $x := 0$. Draw an α-distributed discrete sample for the initial state,
2) the chain is in state l of block i (for the left-most state of the block, $l = b_i$)
   - $c = \text{Geo}(z_i)$,
   - $x += \text{Erl}(cb_i + l, \lambda_i)$
   - if the next block is the absorbing state go to 3), otherwise $l = b_{i+1}$, $i = i+1$ and go to 2)
3) Return x.

The first three methods are applicable to general PH distributions. If we restrict our attention to subclasses, more efficient methods can be designed. First, consider the APH class in CF-1 form. As a special case of the Monocyclic form, the CF-1 form is a chain of states, where each state has exactly one successor state (cf. Figure 4(a)), and thus the next state is not chosen randomly. Hence, once an initial state has been selected, the random variate is simply the sum of exponentially distributed samples from each of the successor states:[2]

`Procedure SimplePlay:`

1) $x := 0$. Draw an α-distributed discrete sample for the initial state.
2) The chain is in state i.
   - $x += \text{Exp}(-a_{ii})$,
   - $i += 1$,
   - if the next state is the absorbing state go to 3), otherwise go to 2).
3) Return x.

---

[2] Note that the transition rates in the CF-1 form are usually not identical, hence we cannot simply draw an Erlang-distributed sample.

If we assume a Hyper-Erlang distribution, represented as shown in Figure 3(a), we can simplify the procedure `Count`, by using our knowledge that each of the branches is an Erlang distribution:

```
Procedure SimpleCount:
```

1) Draw a β-distributed discrete sample to choose an Erlang branch i.
2) Return $\mathrm{Erl}(b_i, \lambda_i)$.

5.1 Costs of generating PH-distributed numbers

In the previous section we argue that the methods for generating random variates differ in their efficiency. We will now treat the costs of random number generation from phase-type distributions in a more formal way. All of the algorithms use exponential random variates for the sojourn times and uniform random variates for choosing the initial state. `Play` and `Count` additionally use uniform random variates for choosing successor states, while the `Monocyclic` algorithm needs geometrically distributed numbers for the number of loops in each Feedback-Erlang block. In order to draw from an exponential or geometric distribution, we need uniform random variates and logarithm operations. Therefore, we define the following two metrics for measuring algorithm complexity:

Definition 0.6. Let *#uni* and *#ln* be the number of uniform variates and logarithm operations, respectively, that are required for generating one PH-distributed random variate from a given PH distribution $(\alpha, \mathbf{A})$.

Using these metrics, we can compare the complexity of the algorithms. We consider both worst-case and average costs.

5.1.1 Worst-Case Costs

Let $\tilde{n}$ denote the length of the longest possible path through the CTMC. For the `Play` method, we draw one exponentially distributed random variate for each traversed state, and hence need one logarithm and one uniform random variate per step, as well as an additional uniform for choosing the next state. For this method, *#uni* and *#ln* are proportional to $\tilde{n}$. However, $\tilde{n}$ is not defined if there are cycles in the CTMC. Therefore, worst-case costs are not defined for `Play`.

The same problem with the unknown maximum number of state traversals occurs with the `Count` method. However, in this case we only draw Erlang-distributed samples (one for each state). Therefore, the maximum number of logarithm operations is bounded by the number of states: $\#ln = n$. Similarly, for the `Monocyclic` method we draw one Erlang-distributed and one geometrically-distributed sample for each Feedback-Erlang block. The latter requires another two logarithm operations, in addition to the one for generating the Erlang sample. As the worst case occurs when we start in the first block, the worst-case number of traversed FE blocks is m, and thus $\#ln = 3m$.

For APH in CF-1 form and using the `SimplePlay` method, the worst case is if the chain is entered at state $i = 1$, since in that case we have to traverse the whole chain. Thus, $\tilde{n} = n$. Obviously, for a Hyper-Erlang distribution in CF-1 form, $\tilde{n} = n$ holds as well. However, if we consider the Hyper-Erlang form and simulation using the `SimpleCount` method, the worst case is equivalent to choosing the longest Erlang branch. In that case, $\tilde{n} = \max b_i \leq n$. The worst-case costs can be computed as follows: With every class, we need one uniform random variate to choose the initial state. When using the $\mathrm{APH}(n)$ class in CF-1 form

Table 2 Theoretical Costs of generating PH distributed random variates from different PH classes and using different PH representations (where $\nu = (n, n-1, \ldots, 1)$, $n^* = \alpha(\mathrm{diag}\langle 1/a_{ii}\rangle \mathbf{A})^{-1}\mathbb{1}$).

PH Class	Worst Case #uni	Worst Case #ln	Average Case #uni	Average Case #ln
HEx(n) `SimpleCount`	2	1	2	1
HErD(n) `SimpleCount`	$\max b_i + 1$	1	$\beta b^{\mathsf{T}} + 1$	1
APH(n) `SimplePlay`	$n+1$	n	$\alpha\nu^{\mathsf{T}} + 1$	$\alpha\nu^{\mathsf{T}}$
PH(n) `Play`	∞	∞	$2\bar{n}+1$	$\bar{n}$
PH(n) `Count`	∞	n	$2\bar{n}+1$	n
`Monocyclic`	∞	$3m$	$\omega\varphi^{\mathsf{T}} + \alpha\psi^{\mathsf{T}}\omega\vartheta^{\mathsf{T}}$	

we need $\tilde{n} = n$ uniforms and $\tilde{n} = n$ logarithms for the consecutive phases. With the HErD class and the `SimpleCount` method we need $\tilde{n} = \max b_i$ additional random variates and one logarithm to obtain an Erlang-distributed random number. We summarise these results in the left half of Table 2.

5.1.2 Average Costs

In general, we do not expect to have worst-case behaviour, but are more interested in average costs. This measure is based on the average number of state transitions up to absorption,

$$\bar{n} = \alpha(\mathrm{diag}\langle 1/a_{ii}\rangle \mathbf{A})^{-1}\mathbb{1}.$$

Applying the `Play` method for the general PH class, in each step we need two uniform random variates (one for the exponential sample and one for choosing the next state, see above), and one logarithm operation. As before, applying the `Count` procedure instead, the number of logarithms is $\#ln = n$, while the number of uniforms stays $\#uni = \bar{n}$.

Canonical forms enable explicit expressions for $\bar{n}$. For Mono$(\alpha, m, b, \lambda, z)$ we introduce vector ω of size m, whose ith element is the probability of starting from Feedback-Erlang block i (e.g. $\omega_1 = \sum_{j=1}^{b_1} \alpha_j$), vector φ of size m, whose ith element is $\varphi_i = \frac{z_i b_i}{1-z_i} + \sum_{j=i+1}^{m} \frac{b_j}{1-z_j}$ (the mean number of steps spent in a Feedback-Erlang block from the first feedback, i.e. excluding the steps from the initial state to the feedback state in the first passage through the initial block), vector ψ of size n whose ith element indicates how many phases are needed to reach the next Feedback-Erlang block (e.g. if $b_1 \geq 2$ then $\psi_1 = b_1, \psi_2 = b_1 - 1$).

Using these notations the mean number of steps till absorption is

$$\bar{n} = \omega\varphi^{\mathsf{T}} + \alpha\psi^{\mathsf{T}},$$

where $\alpha\psi^{\mathsf{T}}$ contains the number of steps if there is no feedback (i.e., if $z_i = 0$, for $i = 1, \ldots, m$) and $\omega\varphi^{\mathsf{T}}$ contains the additional number of steps due to the loops in the Feedback-Erlang block.

The mean number of ln operations is

$$\ell^* = \omega\vartheta^{\mathsf{T}},$$

where ϑ is a row vector of size m whose ith element indicates the number of required ln operations starting from block i. $\vartheta_i = \sum_{j=i}^{m}(1 + 2\ \mathrm{sgn}(z_j))$, since a degenerate Feedback-Erlang block with $z_i = 0$ is Erlang(l, λ_i) distributed which requires one ln operation and a non degenerate ($z_i > 0$) Feedback-Erlang block requires three ln operations, two ln operations for $c = \mathrm{Geo}(z_i)$ and one for $\mathrm{Erl}(cb_i + l, \lambda_i)$.

For the APH class in CF-1 form, there exists an even simpler expression, as the number of traversed states depends only on the initial state, which in turn is determined by the initial probability vector α. Thus, for APH in CF-1 form,

$$\bar{n} = \alpha\nu^{\mathsf{T}}, \text{ where } \nu = (n, n-1, \ldots, 1).$$

Equivalently, for the HErD class, $\bar{n}$ is a weighted sum of the lengths of the Erlang branches:

$$\bar{n} = \alpha b^{\mathsf{T}}.$$

5.2 Optimisation

Considering the costs for the different methods discussed in the previous sections, it becomes clear that both the representation of the distribution and the method have an impact on the efficiency of PH random variate generation. One immediate question is then: What is the optimal representation to generate random variates efficiently? While the answer to this question is not yet available for the general PH case, [22] presents the following result for APH in CF-1 form:

Lemma 0.7. *[22] Given a Markovian representation $(\alpha, \mathbf{A})$ in CF-1 form, the representation $(\alpha^*, \mathbf{A}^*)$ that reverses the order of the rates is optimal with respect to $\bar{n}$ if α^* is a stochastic vector. In this case, all bi-diagonal representations are Markovian.*

The proof given in [22] relies on the observation that swapping two adjacent rates a_i, a_{i+1} moves probability mass towards the end of the chain only if $a_i < a_{i+1}$. Thus, reversing the CF-1 order (where $a_i \leq a_{i+1}$ for all i) gives an initial probability vector α where probability mass is concentrated at the higher indices. Recalling from above, $\bar{n} = \alpha v^{\mathsf{T}}$ for APH, i.e. high probability for states close to absorption implies low average costs.

Note, however, that reversing the CF-1 form may result in α with negative entries [22]. In this case, the tuple $(\alpha^*, \mathbf{A}^*)$ still represents the same distribution, but the representation does not have a Markovian interpretation anymore, and thus $\bar{n}$ is not defined, nor can `SimplePlay` be applied. The optimal ordering can then be found by exhaustive search over all $n!$ possible orderings, or by heuristics that try to find a Markovian representation that is as similar as possible to the reversed CF-1. The heuristics presented in [22] either start from the CF-1 form and apply pair-wise swappings until the result would be non-Markovian, or start from the reversed CF-1 and try to reach a Markovian representation.

6 Conclusion

In this chapter we introduced the basics of using phase-type distributions as tools in resilience evaluation, discussing the complete workflow from fitting to application in both analytical and simulation methods.

References

1. Neuts, M.F.: Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach. Dover Publications, Inc., New York (1981)
2. Horváth, A., Telek, M.: PhFit: A General Phase-Type Fitting Tool. In: TOOLS '02: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools, London, UK, Springer-Verlag (2002) 82–91
3. Telek, M., Heindl, A.: Matching Moments for Acyclic Discrete and Continous Phase-Type Distributions of Second Order. International Journal of Simulation Systems, Science & Technology **3**(3–4) (December 2002) 47–57
4. O'Cinneide, C.A.: Characterization of Phase-Type Distributions. Stochastic Models **6** (1990) 1–57
5. Cumani, A.: On the Canonical Representation of Homogeneous Markov Processes Modelling Failure-time Distributions. Microelectronics and Reliability **22** (1982) 583–602
6. Mocanu, S., Commault, C.: Sparse Representations of Phase-type Distributions. Commun. Stat., Stochastic Models **15**(4) (1999) 759 – 778
7. O'Cinneide, C.A.: Phase-Type Distributions and Invariant Polytopes. Advances in Applied Probability **23**(3) (1991) 515–535
8. He, Q.M., Zhang, H.: Spectral Polynomial Algorithms for Computing Bi-Diagonal Representations for Phase Type Distributions and Matrix-Exponential Distributions. Stochastic Models **22** (2006) 289–317
9. Pulungan, R.: Reduction of Acyclic Phase-Type Representations. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany (2009)

10. Aldous, D., Shepp, L.: The least variable phase-type distribution is erlang. Stochastic Models **3** (1987) 467–473
11. Telek, M., Heindl, A.: Moment bounds for acyclic discrete and continuous phase-type distributions of second order. In: Proc. of UK Performance Evaluation Workshop. (2002)
12. Horváth, G., Miklós, T.: A canonical representation of order 3 phase type distributions. In Wolter, K., ed.: Formal methods and stochastic models for Performance Evaluation. Volume 4748 of LNCS., Berlin, Germany, Springer (2007) 48–62
13. Horváth, G., Miklós, T.: A minimal representation of Markov arrival processes and a moments matching method. Performance Evaluation **64(9-12)**(1153-1168) (August 2007)
14. van de Liefvoort, A.: The moment problem for continuous distributions. Technical Report WP-CM-1990-02, University of Missouri, Kansas City (1990)
15. Lang, A., Arthur, J.: Parameter Approximation for Phase-Type Distributions. Matrix-Analytic Methods in Stocastic Modells **183** (1996) 151–206
16. Bodrog, L., Horváth, A., Telek, M.: Moment characterization of matrix exponential and markovian arrival processes. Annals OR **160**(1) (2008) 51–68
17. Thümmler, A., Buchholz, P., Telek, M.: A Novel Approach for Phase-Type Fitting with the EM Algorithm. IEEE Trans. Dependable Secur. Comput. **3**(3) (2006) 245–258
18. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39** (1997) 1–38
19. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling. Series on statistics and applied probability. ASA-SIAM (1999)
20. Brown, E., Place, J., de Liefvoort, A.V.: Generating Matrix Exponential Random Variates. Simulation **70** (April 1998) 224–230
21. Neuts, M.F., Pagano, M.E.: Generating random variates from a distribution of phase type. In: WSC '81: Proceedings of the 13th Winter Simulation Conference, Piscataway, NJ, USA, IEEE Press (1981) 381–387
22. Reinecke, P., Telek, M., Wolter, K.: Reducing the costs of generating aph-distributed random numbers. In Müller-Clostermann, B., Echtle, K., Rathgeb, E., eds.: MMB & DFT 2010. Number 5987 in LNCS, Springer-Verlag Berlin Heidelberg (2010) 274–286