

PH FITTING ALGORITHM AND ITS APPLICATION TO RELIABILITY ENGINEERING

Hiroyuki Okamura Tadashi Dohi
Hiroshima University

(Received September 2, 2015; Revised December 21, 2015)

Abstract This article presents parameter estimation of phase-type (PH) distribution. PH distribution is widely used in model-based performance evaluation such as reliability and queueing system, and is defined by an absorbing continuous-time Markov chain (CTMC). Since non-Markovian model can be approximated by a CTMC by replacing general distributions with PH distributions, the parameter estimation is a challenging issue in stochastic modeling. We focus on the statistical inference algorithms to estimate not only the parameters of PH distribution with grouped, truncated and missing data but also the probability density function, and give two examples of PH fitting in reliability engineering.

Keywords: Reliability, phase-type distribution, maximum likelihood estimation, EM algorithm, transient analysis, continuous-time Markov chain

1. Introduction

Stochastic models are frequently used for evaluation of system performance indices in queueing and reliability analyses. In particular, dynamic system behavior of computer systems, telecommunication network systems, or production systems, is described as a state-dependent stochastic process. In such model-based performance evaluation, discrete-time and continuous-time Markov chains (DTMCs/CTMCs) are quite popular to model the system behavior [12]. The time-homogeneous DTMC and CTMC are mathematically tractable and can provide both stationary and transient system performance indices through analytical approaches. However, when some of state transitions are represented by non-Markovian-type distributions such as Weibull, normal, log-normal distributions, it becomes difficult to take analytical approaches in many cases. In such cases, although Monte-Carlo (MC) simulation approach may be practically useful in some situations, it may not be often suitable when the high accuracy of system indices is required. For example, in reliability analysis of mission-critical systems, five through nine nines of quantitative reliability are demanded in practice. Hence, the analytical approach, if possible, is more preferred to simulation approach in quantitative reliability assessment.

Phase-type (PH) distribution is defined for a non-negative random variable which is an absorbing time on time-homogeneous DTMC or CTMC process. It is known that the stochastic model described by PH distributions is reduced to a time-homogeneous Markov process with DTMC or CTMC. Furthermore the PH distribution can get close to any probability distribution with any precision if the number of states of the underlying DTMC or CTMC infinitely increases [5]. This property provides advantages to both approximation and estimation in model-based performance analysis. For example, PH expansion is regarded as one of the most useful approximation techniques for non-Markovian models, where non-Markovian-type distributions are replaced by PH distributions under the situ-

ation where the probability density function (p.d.f.) or cumulative distribution function (c.d.f.) of non-Markovian-type distribution is known. PH expansion approximates the non-Markovian models to Markov models which are analytically tractable, and the PH-expanded Markov model approximates the non-Markovian model accurately if the number of states of the underlying DTMC or CTMC is appropriate. In the estimation, even though the true distribution is unknown, PH distribution is expected to be close to the true distribution when the number of states of the underlying DTMC or CTMC infinitely increases*. That is, by assuming that the true distribution is a family of PH distributions, we develop a non-parametric-like estimation approach.

In both cases, key issues are how to determine the appropriate number of states of the underlying DTMC or CTMC and how to determine parameters of PH distributions. These are called the PH fitting problem. The article mainly focuses on the second issue, i.e., how to determine PH parameters when the number of states of the underlying DTMC or CTMC is fixed. Since the essential difference between approximation and estimation is whether we know the p.d.f. or c.d.f. of the true distribution or not, PH approximation is used to indicate the PH fitting when the p.d.f. or c.d.f. of the true distribution is known. On the other hand, PH estimation means the PH fitting when we do not know the true distribution. Furthermore, the terminology, PH fitting, is used when it is not necessary to distinguish whether we know the true distribution or not in the article.

Since PH distributions generally involve a large number of parameters, PH fitting is not so tractable. In past, there are two major approaches for PH fitting with moments and Kullback-Leibler (KL) divergence. The former determines PH parameters so as to fit theoretical moments of PH distribution to moments from samples or p.d.f. In PH estimation, the moments are estimated from samples. In PH approximation, the moments are calculated from the p.d.f. of distribution. On the other hand, KL-based PH fitting is to find PH parameters that minimize KL divergence from a PH distribution to the true distribution. In the context of PH estimation, this corresponds to maximum likelihood (ML) estimation of PH distribution.

This tutorial article presents a survey of PH fitting and introduces the state-of-the-art PH fitting algorithm for PH distributions defined on CTMC presented by [67]. More specifically, we give KL-based PH approximation and estimation procedures with the EM (expectation-maximization) algorithm [17, 87] and uniformization technique. We also refer to the numerical tool implemented with the statistical software R. Further, we give two examples of PH fitting; PH estimation for software reliability assessment and PH approximation for dynamic reliability modeling based on a Markov regenerative process (MRGP). We introduce the so-called phase-type software reliability model (PH-SRM) [59, 60, 63] and mention how to use the PH estimation algorithm to the parameter estimation of PH-SRM. In the second example, we apply the PH approximation (PH expansion) to describe a parallel system with single repair facility by means of the MRGP and evaluate the pointwise availability.

The organization of this article is as follows. In Section 2, we describe the mathematical definition and properties of PH distribution. Section 3 overviews PH fitting and introduces an efficient ML estimation algorithm when grouped, truncated and missing data are given. In Section 4, we formulate PH-SRM in the context of software reliability engineering, and present how to use the PH estimation algorithm to get model parameters of PH-SRM. Section 5 demonstrates the PH approximation of MRGP. We describe the definition of

*In practice, the bias of estimates should be considered.

MRGP and provide the Kronecker representation of the approximate MRGP. Finally, we conclude the article with future research directions of PH fitting in Section 6.

Abbreviation and acronym

PH: Phase type
 CTMC: Continuous-time Markov chain
 DTMC: Discrete-time Markov chain
 MC: Monte-Carlo
 p.d.f.: probability density function
 c.d.f.: cumulative distribution function
 KL: Kullback-Leibler
 ML: Maximum likelihood
 EM: Expectation maximization
 MRGP: Markov regenerative process
 PH-SRM: Phase-type software reliability model
 APH: Acyclic PH
 GPH: General PH
 CF1: Canonical form 1
 CF2: Canonical form 2
 CF3: Canonical form 3
 LS: Laplace-Stieltjes
 AIC: Akaike information criterion
 BIC: Bayesian information criterion
 MM: Moment matching
 LLF: Log-likelihood function
 MCMC: Markov chain Monte Carlo
 IID: independent and identically distributed
 NA: Not available
 MAP: Markovian arrival process
 SRM: Software reliability model
 NHPP: Non-homogeneous Poisson process
 CPH-SRM: PH-SRM with CF1
 HEr-SRM: PH-SRM with hyper-Erlang distribution
 GO: Goel-Okumot model
 ISS: Inflection S-shaped model
 Q-R ISS: Quasi-renewal time-delay model based on ISS
 Beta-Mix: NHPP-based SRM with beta mixture
 MSE: Mean squared error
 MLL: Maximum log-likelihood
 DF: Degrees of freedom
 EXP: Exponential (transition)
 GEN: General (transition)
 DE: Double exponential
 WEI: Weibull distribution
 LOG: Log-normal distribution
 CV²: Squared coefficient of variation
 MRSPN: Markov regenerative stochastic Petri net

2. PH Distribution

2.1. Definition

The PH distribution is defined as a probability distribution for time to absorption in a time-homogeneous CTMC with an absorbing state. Without loss of generality, we consider an absorbing CTMC on the finite state space $\{1, 2, \dots, m, m+1\}$ with the following infinitesimal generator:

$$\mathbf{Q}_{PH} = \begin{pmatrix} \mathbf{T} & \boldsymbol{\tau} \\ \mathbf{0} & 0 \end{pmatrix}, \quad (2.1)$$

where the state $m+1$ indicates the absorbing state. Also, \mathbf{T} is an infinitesimal generator between transient states $\{1, 2, \dots, m\}$, and $\boldsymbol{\tau}$ is a column vector for transition rates to the absorbing state. Let τ_i and $\lambda_{i,j}$ denote the i -th entry of $\boldsymbol{\tau}$ and the (i,j) -entry of \mathbf{T} , respectively. By using the column vector $\mathbf{1}$ whose all elements are 1, the vector $\boldsymbol{\tau}$ is given by $\boldsymbol{\tau} = -\mathbf{T}\mathbf{1}$. The above absorbing CTMC process is called a *phase process*, denoted by J_x where x indicates time, and the transient state of phase process is called *phase* in general. The PH random variable is defined by the first passage time to the absorbing state on the phase process J_x , i.e.,

$$X = \inf\{x \geq 0; J_x = m+1\}. \quad (2.2)$$

Suppose that the initial phase J_0 is determined by a probability (row) vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_m)$ over the transient states. Then the p.d.f. and c.d.f. of X can be written with the PH parameters $(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$:[†]

$$f_{PH}(x) = \boldsymbol{\alpha} \exp(\mathbf{T}x)\boldsymbol{\tau}, \quad 0 \leq x < \infty, \quad (2.3)$$

$$F_{PH}(x) = 1 - \boldsymbol{\alpha} \exp(\mathbf{T}x)\mathbf{1}, \quad 0 \leq x < \infty, \quad (2.4)$$

respectively.

2.2. Properties

It is known that PH distribution consists of the mixture/convolution of exponential distributions. Hence it involves some well-known probability distributions as the special cases. The exponential distribution is the simplest PH distribution when the number of phases (the number of transient states in the phase process) is 1. Similarly, since the Erlang distribution, hyper-exponential distribution, hypo-exponential distribution and mixture of Erlang distributions can be obtained from the mixture/convolution of exponential distributions, they are also subclasses of PH distributions. These subclasses do not have any cyclic structure between transient states, and then belong to a class of acyclic PH distribution (APH). Since the APH distribution is mathematically tractable and a superclass of commonly-used distributions such as the exponential and Erlang distributions, the APH distribution is quite important in applications. Also the PH distribution whose parameters do not have any constraint is called the general PH distribution (GPH). The relationship among subclasses of PH distributions is summarized in Figure 1.

Cumani [16] derived three canonical forms of APH distribution; CF1, CF2 and CF3, and proved that all of APH distributions can be reconfigured to any one of the canonical forms

[†]In fact, since the PH representation $(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$ is redundant, the PH distribution is essentially defined only by $(\boldsymbol{\alpha}, \mathbf{T})$. In our PH fitting, it is noted that the diagonals of \mathbf{T} are computed from non-diagonals of \mathbf{T} and $\boldsymbol{\tau}$, so we use the PH representation $(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$.

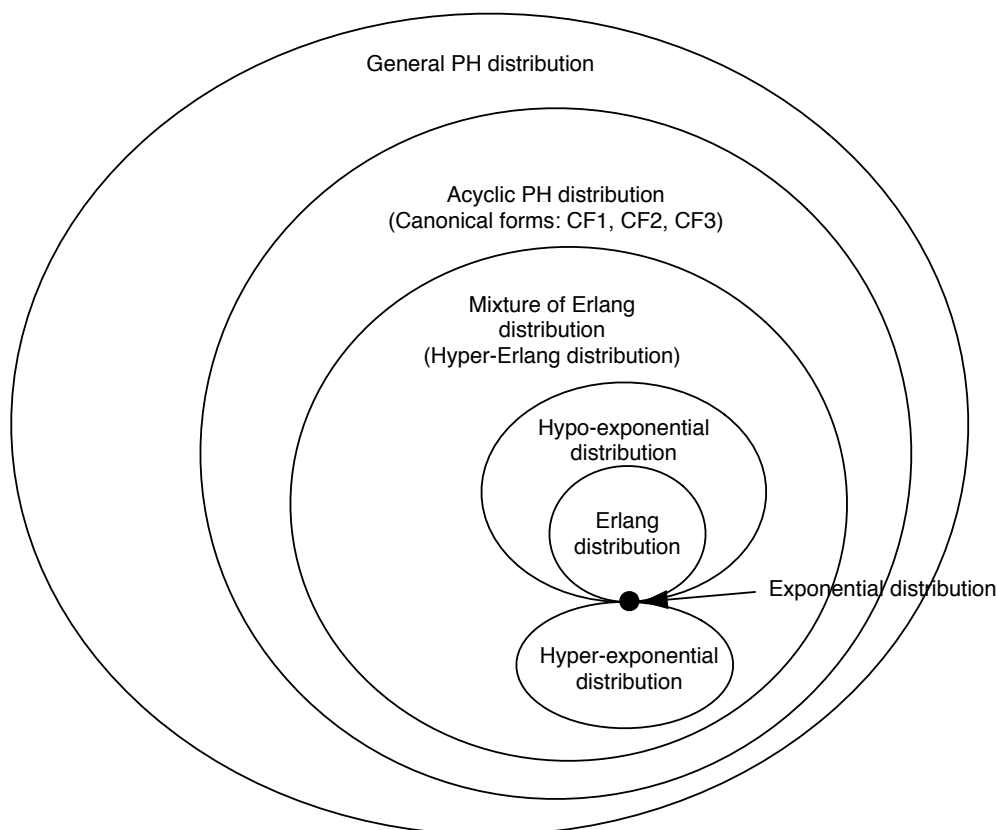


Figure 1: Relationship among subclasses of PH distributions

which have $2m - 1$ free parameters. Specifically, CF1 (canonical form 1) is defined by

$$\boldsymbol{\alpha} = (\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_m), \quad (2.5)$$

$$\mathbf{T} = \begin{pmatrix} -\beta_1 & \beta_1 & & & & \\ & -\beta_2 & \beta_2 & & & \\ & & \ddots & \ddots & & \\ & & & -\beta_{m-1} & \beta_{m-1} & \\ & & & & -\beta_m & \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_m \end{pmatrix}, \quad (2.6)$$

where $0 < \beta_1 \leq \cdots \leq \beta_m$. Throughout the article, blanks in a matrix indicate zero entries. When the parameter constraint $0 < \beta_1 \leq \cdots \leq \beta_m$ does not hold in Equation (2.6), the resulting PH distribution is called a bidiagonal PH distribution [28] to distinguish it from CF1.

On the other hand, CF2 and CF3 are given as variants of CF1. Concretely speaking, by using CF1 parameters, CF2 can be represented in the form;

$$\boldsymbol{\alpha} = (1 \quad 0 \quad \cdots \quad 0), \quad (2.7)$$

$$\mathbf{T} = \begin{pmatrix} -\beta_m & \alpha_1\beta_m & \alpha_2\beta_m & \cdots & \alpha_{m-1}\beta_m \\ & -\beta_1 & \beta_1 & & \\ & & \ddots & \ddots & \\ & & & -\beta_{m-2} & \beta_{m-2} \\ & & & & -\beta_{m-1} \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} \alpha_n\beta_n \\ 0 \\ \vdots \\ 0 \\ \beta_{m-1} \end{pmatrix}. \quad (2.8)$$

Also, CF3 is given by

$$\boldsymbol{\alpha} = (1 \ 0 \ \cdots \ 0), \tag{2.9}$$

$$\mathbf{T} = \begin{pmatrix} -\beta_m & (1 - \gamma_m)\beta_m & & & & \\ & -\beta_{m-1} & (1 - \gamma_{m-1})\beta_{m-1} & & & \\ & & \ddots & \ddots & & \\ & & & -\beta_2 & (1 - \gamma_2)\beta_2 & \\ & & & & -\beta_1 & \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} \gamma_m\beta_m \\ \vdots \\ \gamma_2\beta_2 \\ \gamma_1\beta_1 \end{pmatrix}, \tag{2.10}$$

where $\gamma_i = \alpha_i / (1 - \sum_{l=i+1}^m \alpha_l)$. When the parameter of CF3 are allowed to be complex numbers, the structure of CF3 reduces Coxian distribution [19,85]. That is, CF3 is a special case of Coxian distribution in which the parameters are restricted to be real numbers.

PH distribution has closure properties for convolution, mixture, minimum and maximum operations. Let X_A and X_B be independent PH random variables with parameters $(\boldsymbol{\alpha}_A, \mathbf{T}_A, \boldsymbol{\tau}_A)$ and $(\boldsymbol{\alpha}_B, \mathbf{T}_B, \boldsymbol{\tau}_B)$, respectively. Then $X_C = X_A + X_B$ becomes a PH random variable with the following parameters:

- convolution

$$\boldsymbol{\alpha}_C = (\boldsymbol{\alpha}_A \ \mathbf{0}), \quad \mathbf{T}_C = \begin{pmatrix} \mathbf{T}_A & \boldsymbol{\tau}_A \boldsymbol{\alpha}_B \\ & \mathbf{T}_B \end{pmatrix}, \quad \boldsymbol{\tau}_C = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau}_B \end{pmatrix}. \tag{2.11}$$

A mixture of X_A and X_B is expressed as $X_C = \chi X_A + (1 - \chi)X_B$ where χ is the indicator random variable with probability p . This means the random variable X_C chooses either of X_A or X_B with probability p . The random variable X_C becomes a PH random variable having the parameters:

- mixture

$$\boldsymbol{\alpha}_C = (p\boldsymbol{\alpha}_A \ (1 - p)\boldsymbol{\alpha}_B), \quad \mathbf{T}_C = \begin{pmatrix} \mathbf{T}_A & \\ & \mathbf{T}_B \end{pmatrix}, \quad \boldsymbol{\tau}_C = \begin{pmatrix} \boldsymbol{\tau}_A \\ \boldsymbol{\tau}_B \end{pmatrix}. \tag{2.12}$$

In a fashion similar to the previous discussion, let $X_C^{\min} = \min(X_A, X_B)$ and $X_C^{\max} = \max(X_A, X_B)$ be minimum and maximum of X_A and X_B , respectively. Then they are PH random variables with the following parameters:

- minimum

$$\boldsymbol{\alpha}_C = \boldsymbol{\alpha}_A \otimes \boldsymbol{\alpha}_B, \quad \mathbf{T}_C = \mathbf{T}_A \oplus \mathbf{T}_B, \quad \boldsymbol{\tau} = \boldsymbol{\tau}_A \otimes \mathbf{1} + \mathbf{1} \otimes \boldsymbol{\tau}_B, \tag{2.13}$$

- maximum

$$\boldsymbol{\alpha}_C = (\boldsymbol{\alpha}_A \otimes \boldsymbol{\alpha}_B \ \mathbf{0} \ \mathbf{0}), \quad \mathbf{T}_C = \begin{pmatrix} \mathbf{T}_A \oplus \mathbf{T}_B & \boldsymbol{\tau}_A \otimes \mathbf{I} & \mathbf{I} \otimes \boldsymbol{\tau}_B \\ & \mathbf{T}_B & \\ & & \mathbf{T}_A \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\tau}_B \\ \boldsymbol{\tau}_A \end{pmatrix}, \tag{2.14}$$

where \otimes and \oplus are the Kronecker product and sum, respectively.

Furthermore, let X_1, \dots, X_n be independent PH random variables. Then we suppose that \tilde{X}_A and \tilde{X}_B are random variables that are generated by convolution, mixture, minimum and maximum operations of X_1, \dots, X_n . According to the above properties, \tilde{X}_A and \tilde{X}_B are also PH random variables but they may not be independent. However, it is clear

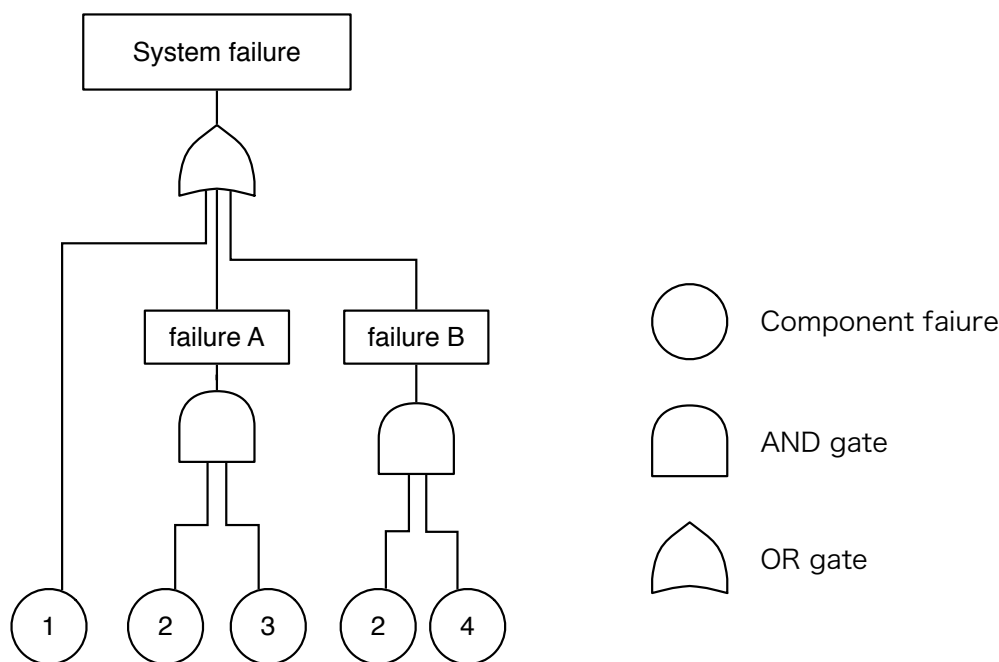


Figure 2: An example of fault tree

that a random variable generated by convolution, mixture, minimum and maximum operations of \tilde{X}_A and \tilde{X}_B becomes a PH random variable generated by independent PH random variables X_1, \dots, X_n . Thus, although the expressions of p.d.f. are more complex, the closure properties hold even in the class of \tilde{X}_A and \tilde{X}_B .

In the reliability engineering, it is commonly supposed that a system failure is caused by multiple component failures. This causal relationship is often modeled by fault trees. The fault tree analysis is a kind of root-cause analysis which represents the condition that causes an event occurrence by using AND/OR conditions [7]. Figure 2 illustrates an example of fault tree. In this tree, leaf nodes correspond to component failure events, and AND/OR gates express the failure conditions for upper events. The AND gate means that the upper event occurs only when all the leaf events occur. The OR gate indicates that the upper event occurs when at least one event occurs.

In AND gate, X_A and X_B denote random variables of event occurrence time for lower events A and B . Then the upper event occurrence time is given by $\max(X_A, X_B)$. Similarly, in the case of OR gate, the upper event occurrence time is given by $\min(X_A, X_B)$. That is, the top event occurrence time is given by min-max operations of component failure times. In Figure 2, e.g., the system failure (top event) occurs at $\min(X_1, \max(X_2, X_3), \max(X_2, X_4))$. From this insight, when X_1, \dots, X_4 are mutually independent, we have the following proposition.

Proposition 2.1. *In the fault tree model, when component failure times are assumed to be independent PH random variables, the system failure time also becomes a PH random variable which is constructed by min-max operations of component failure times.*

By combining the above proposition with Equations (2.13) and (2.14), it is easily seen that when component failure times follow independent APH random variables, the system failure time also becomes an APH random variable. Furthermore, since any APH distribution can be transformed to the canonical forms, we have the following proposition.

Proposition 2.2. *In the fault tree model, when component failure times are assumed to be independent APH random variables, the probability distribution of system failure time is expressed as the canonical forms.*

This property implies that even if we do not know the exact failure mechanism for the system failure such as fault trees, the system failure time distribution is represented by a canonical form of PH distributions. In particular, since component failure times are often independent exponential random variables in the reliability engineering, the canonical forms are appropriate for representing the probability distribution of system failure time.

3. PH Fitting

3.1. Historical survey

The purpose of PH fitting is to determine PH parameters $(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$ so as to approximate the target distribution by the fitted PH distribution. In particular, the PH estimation is the PH fitting from data under the situation where we do not know the true distribution and the PH approximation is the PH fitting from p.d.f. or c.d.f. of the true distribution. In both cases, the accuracies of estimation and approximation depend on the number of phases and the subclasses. For example, if the number of phases is 1, the corresponding PH distribution is the exponential distribution. Needless to say, the exponential distribution does not possess the relatively high fitting ability. Hence we encounter the following three research questions on PH fitting:

- (i) What is the appropriate subclass of PH distributions?
- (ii) How many phases are required?
- (iii) How should the estimates of $(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$ be determined?

On the research question (i), three kinds of phase structures, GPH, APH and mixture of exponential/Erlang distributions, are typically used. In particular, since APH distribution can be transformed to the canonical forms, PH fitting for APH distribution is discussed in the context of parameter estimation of canonical forms.

It is worth mentioning that the research question (ii) has been still open as an unresolved problem for PH fitting. Several approaches have been proposed in past researches. Particularly, in the PH approximation, the number of phases is determined by the criterion of approximation. For example, when the criterion is given by the moments of a target distribution, the accuracy of approximation can be defined by how many moments should be used. Osogami and Harchol-Balter [71] and Bobbio et al. [10] gave the moment-based method for APH distribution with the first three moments, and revealed the minimum number of phases that is necessary to match the first three moments. A more sophisticated approach is based on the Laplace-Stieltjes (LS) transform. O’Cinneide [54–57] discussed the characteristics of PH distribution, and provided the lower bounds on the number of phases of PH distribution by means of poles of LS transform. He and Zhang [28] derived the phase reduction algorithm to find the minimal representation of Coxian distribution based on O’Cinneide’s works. On the other hand, in the PH estimation, the theoretically valid approach in the sense of statistics would be to use information criteria such as AIC (Akaike information criterion) [2] and BIC (Bayesian information criterion) [78]. They are used in many applications of statistical model selection problem. The information criterion essentially handles the penalty term incurred by increasing the number of model parameters. That is, this method leads to find the appropriate number of phases by means of information criteria.

On the research question (iii), there are many variations proposed in the past literature, where the methods can be classified into three types. The first method is moment-based approach, which is often called moment matching (MM) method and is mainly used in the PH approximation. The concept of MM method is to find PH parameters so that the population moments can fit the moments derived from samples or p.d.f. of the true distribution. As mentioned before, the accuracy of MM method depends on how many moments is used. In general, it is difficult to develop the MM method with a large number of moments. In fact, many authors focused on only the first two or three moments to determine PH parameters.

The main concern for the earliest PH fitting was the MM method with special structure such as a mixture of Erlang distributions and CF3 with 2 phases using the first two moments in application of Markov model [48]. Altiok [4] presented a method to approximate general distributions by PH distributions via LS transform of the mixture of exponential distributions. van der Heijden [85] gave a simple formula for the MM method of CF3 with the first three moments. Aldous and Shepp [3] showed that the number of phases is easily determined by the coefficient variation for Erlang distribution. Johnson and Taaffe [34–37] and Johnson [33] conducted several works on PH fitting for the mixture of Erlang distributions with non-linear programming techniques in terms of the difference of p.d.f. as well as moments. In the context of the MM method for APH distribution, Telek and Heindl [80] presented a method for 2-phase APH distributions. Osogami and Harchol-Balter [71] and Bobbio et al. [10] proposed the MM methods that provide the minimum number of phases to fit to the first three moments. van de Liefvoort [84] and Telek and Horvath [81] extended the PH distribution to the matrix-exponential distribution, and discussed the MM methods with high-order moments. In PH fitting, the problem of how to approximate a heavy-tailed distribution was often discussed. The heavy-tailed distributions are distributions whose tails are not exponentially decayed. Feldmann and Whittle [20] discussed the method to approximate the heavy-tailed distribution with the mixture of exponentials.

Here we remark on the computational risk in the use of MM method in the PH estimation. It has no doubt that the MM method works well for PH approximation, because moments are exactly computed from the p.d.f. However, in the PH estimation, we should always consider the estimation errors for moments. In general, an estimate of high-order moment is rather sensitive to errors caused by random samples. Thus it is difficult to obtain the accurate estimates of high-order moments when the number of samples is small. In addition, there is the situation where we cannot know even the first moment exactly in the case where some data are missing.

The second approach is to use KL (Kullback-Leibler) divergence which is called KL-based approach in this article. The KL-based approach reduces the maximum likelihood (ML) estimation when we do not know the true distribution. The ML estimation is a commonly used technique to estimate model parameters by maximizing the likelihood so that the observed data are drawn from the model. The advantage of ML estimation is that it can estimate parameters from any type of data and can enjoy several rich properties in statistics, such as asymptotic normality in PH fitting. Then this approach is superior to the moment-based approach in the PH estimation. Also, when we know the p.d.f. of the true distribution, the KL-based approach is to determine PH parameters that minimizes the KL divergence to the true distribution. Since the KL divergence is defined by the integral of the true distribution and a PH distribution, the KL-based approach is also used in the PH approximation through techniques of numerical quadrature. On the other hand, since the computation cost of KL-based approach is relatively high compared to moment-based

approach, the challenging issue is to reduce the computation cost. The EM (expectation-maximization) algorithm [17, 87] is frequently used for PH fitting.

Bobbio and Cumani [9] presented the ML estimation for canonical forms by solving the non-linear programming. In their method, the linear programming was iteratively applied to solve the non-linear programming to maximize the log-likelihood function (LLF). The method was almost similar to the steepest decent method but introduced constraints to ensure the parameter constraints of canonical forms. Also Bobbio and Telek [11] carried out experiments on ML estimation for APH distribution. Asmussen et al. [6] proposed an EM algorithm for GPH distribution, and also presented an idea of using numerical quadrature for PH approximation. Olsson [70] extended the EM algorithm for PH estimation with censored data. Okamura et al. [66] improved Asmussen's EM algorithm with respect to time complexity of the number of phases. Also, they proposed an improved algorithm for estimating APH distribution with EM algorithm. Moreover, in [67], the PH fitting with EM algorithm is extended to handle grouped, truncated and missing data. On the other hand, for PH fitting with the mixture of Erlang distributions (hyper-Erlang distribution), Thummler et al. [82] and Panchenko and Thumler [72] developed the EM algorithms which can determine the phase structure of hyper-Erlang distribution. Compared to APH distribution, the hyper-Erlang distribution is inferior to representation ability because the hyper-Erlang distribution is a sub-class of APH distribution. However, their algorithm is much faster than the EM algorithms for APH distributions. Based on the EM algorithms for hyper-Erlang distributions, Reinecke et al. [77] proposed the method to combine the data clustering and PH fitting to fit to heavy-tailed distribution according to the idea of Feldmann and Whitte [20].

The third method is Bayes estimation, which is only used for the PH estimation. In the Bayesian paradigm, it is assumed that unknown model parameters are random variables. By observing field data, the corresponding probability distribution of model parameter is updated according to the Bayes theorem. The updated probability distribution of model parameter is called the posterior distribution, and the probability distribution of model parameter before the update is called the prior distribution. The main feature of Bayes estimation is to compute the posterior distributions of parameters. The Bayes estimation enables us to evaluate not only interval estimation of parameters but also the predictive distribution in many cases. However, the computation of posterior distribution is often more expensive than moment-based and KL-based approaches as well.

Bladt et al. [8] proposed an MCMC (Markov chain Monte Carlo) algorithm for GPH distribution with Markov jump process to approximate the posterior distributions. Watanabe et al. [86] considered an improved MCMC algorithm by using the uniformization technique to improve the computation speed. Yamaguchi et al. [92] and Okamura et al. [68] proposed the variational Bayes algorithms for the mixture of Erlang distributions and GPH distribution. The presented methods are much faster than MCMC-based algorithms.

3.2. KL-based PH fitting

The KL-based approach is to determine PH parameters that minimize KL divergence. Consider the situation where a p.d.f. $f(x)$ is approximated by a PH distribution with p.d.f. $f_{PH}(x)$. The KL divergence $KL(f, f_{PH})$ between $f(x)$ and $f_{PH}(x)$ is defined as follows.

$$\begin{aligned} KL(f, f_{PH}) &= \int_0^{\infty} f(x) \log \frac{f(x)}{f_{PH}(x)} dx \\ &= \int_0^{\infty} f(x) \log f(x) dx - \int_0^{\infty} f(x) \log f_{PH}(x) dx. \end{aligned} \quad (3.1)$$

The KL divergence means the distance between $f(x)$ and $f_{PH}(x)$. If $f(x)$ and $f_{PH}(x)$ are exactly same, the KL divergence becomes 0. Since the first term of Equation (3.1) is a constant with respect to the PH parameters, we find $f_{PH}(t)$ maximizing $\int_0^\infty f(x) \log f_{PH}(x) dx$.

In the PH estimation, since the true distribution $f(x)$ is unknown, we estimate the second term of Equation (3.1) from observed data. For example, let x_1, \dots, x_n be IID (independent and identically distributed) samples from the true distribution. The second term can be estimated as the following LLF:

$$\int_0^\infty f(x) \log f_{PH}(x) dx \approx \sum_{i=1}^n \log f_{PH}(x_i), \quad (3.2)$$

and the PH parameters are determined to solve the maximization problem the above LLF.

In the PH approximation, the p.d.f. of the true distribution is known. When applying a suitable numerical quadrature to the second term of Equation (3.1), we have

$$\int_0^\infty f(x) \log f_{PH}(x) dx \approx \sum_{i=1}^n w_i \log f_{PH}(x_n). \quad (3.3)$$

The above equation implies that the KL-based PH approximation reduces the ML estimation with the weighted IID samples $(x_1, w_1), \dots, (x_n, w_n)$. Also, the accuracy of PH approximation is evaluated by the KL divergence between them.

3.3. EM algorithm

The EM algorithm is an iterative method for ML estimation under the situation where there exists unobserved data [17, 87]. Let \mathcal{D} and \mathcal{U} be observable and unobservable data vectors, respectively, and we estimate a model parameter vector $\boldsymbol{\theta}$ from only the observable data \mathcal{D} . Then the problem corresponds to finding a parameter vector that maximizes a marginal LLF:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}), \quad (3.4)$$

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \log p(\mathcal{D}; \boldsymbol{\theta}) = \log \int p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U}, \quad (3.5)$$

where $p(\cdot)$ is a probability density or mass function.

From Jensen's inequality, for any model parameter vector $\boldsymbol{\theta}'$, we have

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) &= \log \int p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U} \\ &= \log \int \frac{p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta})}{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}')} p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') d\mathcal{U} \\ &\geq \int p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') \log \frac{p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta})}{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}')} d\mathcal{U} \equiv \mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}'). \end{aligned} \quad (3.6)$$

Since the distribution of unobservable data is

$$p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}) = \frac{p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta})}{\int p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U}}, \quad (3.7)$$

the lower bound of the marginal LLF $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}')$ can be rewritten by

$$\begin{aligned}\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') &= \int p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') \log \frac{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}) \int p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U}}{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}')} d\mathcal{U} \\ &= \int p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') \log \frac{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta})}{p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}')} d\mathcal{U} + \log \int p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U}.\end{aligned}\quad (3.8)$$

Then we have the following formula:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) - \mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') = KL(p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}'), p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta})). \quad (3.9)$$

Also the difference $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) - \mathcal{L}(\boldsymbol{\theta}'; \mathcal{D})$ holds

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) - \mathcal{L}(\boldsymbol{\theta}'; \mathcal{D}) = \mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') - \mathcal{Z}(\boldsymbol{\theta}'; \boldsymbol{\theta}') + KL(p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}'), p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta})). \quad (3.10)$$

Since $KL(\cdot, \cdot) \geq 0$, the model parameter vector $\boldsymbol{\theta}$ that satisfies $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') - \mathcal{Z}(\boldsymbol{\theta}'; \boldsymbol{\theta}') > 0$ yields $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) - \mathcal{L}(\boldsymbol{\theta}'; \mathcal{D}) > 0$. In other words, when we find $\boldsymbol{\theta}$ that maximizes $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}')$ for a fixed $\boldsymbol{\theta}'$, the marginal LLF for $\boldsymbol{\theta}$ is greater than that for $\boldsymbol{\theta}'$.

Let $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ denote the expected LLF of the complete data vector $(\mathcal{D}, \mathcal{U})$ with respect to the distribution for unobservable data vector under a provisional parameter $\boldsymbol{\theta}'$, i.e.,

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}') = \int p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') \log p(\mathcal{D}, \mathcal{U}; \boldsymbol{\theta}) d\mathcal{U}. \quad (3.11)$$

Then the lower bound $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}')$ is rewritten in the form:

$$\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') = Q(\boldsymbol{\theta}|\boldsymbol{\theta}') - \int p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') \log p(\mathcal{U}|\mathcal{D}; \boldsymbol{\theta}') d\mathcal{U}. \quad (3.12)$$

Since the second term of the above equation is constant with respect to $\boldsymbol{\theta}$, we obtain $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}') - \mathcal{Z}(\boldsymbol{\theta}'; \boldsymbol{\theta}') = Q(\boldsymbol{\theta}|\boldsymbol{\theta}') - Q(\boldsymbol{\theta}'|\boldsymbol{\theta}')$. Thus the maximization of $\mathcal{Z}(\boldsymbol{\theta}; \boldsymbol{\theta}')$ is reduced to the maximization $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$.

According to the above discussions, the EM algorithm consists of E-step and M-step. E-step computes the expected LLF $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ with a given provisional parameter vector $\boldsymbol{\theta}'$. In the M-step, we find a new parameter vector $\boldsymbol{\theta}$ that maximizes the expected LLF:

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}}{\operatorname{argmax}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}'). \quad (3.13)$$

Since the marginal LLF for $\boldsymbol{\theta}$ is surely greater than that for $\boldsymbol{\theta}'$, we can perform the same E- and M-steps by replacing $\boldsymbol{\theta}'$ with $\boldsymbol{\theta}$. These steps are repeatedly executed until the parameters converge. It should be noted that the EM algorithm does not guarantee that the parameters converge to ML estimates. That is, the EM algorithm may converge to a local maximum of the marginal LLF. To avoid the convergence to a local maximum, it is important to choose appropriate initial guesses of parameters.

3.4. PH estimation with grouped, truncated and missing data

Here we introduce a PH estimation algorithm proposed by [67] based on ML estimation with EM algorithm. It can be applied to estimating GPH parameters with grouped, truncated and missing data.

The grouped data consists of several time intervals and the number of events that occur in the time intervals. The time points that generate time intervals are called *break points*

Table 1: Examples of grouped, truncated and missing data for PH fitting (break points: 0, 10, 20, . . . , 90, 100)

Data (a)		Data (b)	
Time interval	The number of events	Time interval	The number of events
[0, 10]	1	[0, 10]	1
[10, 20]	NA	[10, 20]	NA
[20, 30]	4	[20, 30]	4
[30, 40]	10	[30, 40]	10
[40, 50]	NA	[40, 50]	NA
[50, 60]	30	[50, 60]	30
[60, 70]	10	[60, 70]	10
[70, 80]	12	[70, 80]	12
[80, 90]	4	[80, 90]	4
[90, 100]	0	[90, 100]	0
[100, ∞]	5	[100, ∞]	NA

in this article. Table 1 (a) presents an example of the grouped, truncated and missing data on break points 0, 10, 20, . . . , 100. In this table, the values in intervals [10, 20] and [40, 50] are missing and denoted as NA (not available). Furthermore, the bottom row means that 5 samples are truncated at 100. Similarly, Table 1 (b) gives another example. In this case, several samples are truncated at 100 but we cannot know the exact number of truncated samples. In this way, it is worth noting that the incomplete data such grouped, truncated and missing data frequently appear in field data analysis.

Consider the grouped data on the break points $0 = x_0 < x_1 < \dots < x_K$, i.e., the data consist of the number of samples for $K + 1$ intervals $[0, x_1), [x_2, x_3), \dots, [x_K, \infty)$. For the sake of notational convenience, let $X^{(k,l)}$ be the l -th sample in the k -th interval. In addition, we assume two kinds of intervals; observable and unobservable intervals. Let \mathcal{I} and $\bar{\mathcal{I}}$ be the sets of indexes of observable and unobservable intervals. Note that \mathcal{I} and $\bar{\mathcal{I}}$ are disjoint, and that $\mathcal{I} \cup \bar{\mathcal{I}}$ gives all the indexes, i.e., $\mathcal{I} \cup \bar{\mathcal{I}} = \{1, 2, \dots, K + 1\}$. We can count the numbers of samples in only the observable intervals, but the numbers of samples in the unobservable intervals are unknown, i.e., the data are missing in the unobservable intervals.

Let $\mathcal{D} = \{n_k\}_{k \in \mathcal{I}}$ be the numbers of samples in observable intervals. Define the following unobserved variables:

- U_k : the number of samples in the k -th unobserved interval $[x_{k-1}, x_k)$, $k \in \bar{\mathcal{I}}$.
- $B_i^{(k,l)}$: an indicator random variable for the event that the phase process of $X^{(k,l)}$ begins with phase i .
- $Z_i^{(k,l)}$: the total sojourn time of phase i on the phase process of $X^{(k,l)}$.
- $Y_i^{(k,l)}$: an indicator random variable for the event that the phase process of $X^{(k,l)}$ is i just before the absorption.
- $M_{i,j}^{(k,l)}$: the total number of phase transitions from i to j on the phase process of $X^{(k,l)}$.

Since the estimates of CTMC parameters are given by frequencies and sojourn time for each state under the situation where the CTMC process is completely observed, the M-step

formulas of PH parameters are obtained from Equation (3.13) in the following:

$$\alpha_i \leftarrow \frac{\sum_{k \in \mathcal{I}} \mathbb{E} \left[\sum_{l=1}^{n_k} B_i^{(k,l)} \middle| \mathcal{D} \right] + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E} \left[\sum_{l=1}^{U_k} B_i^{(k,l)} \middle| \mathcal{D} \right]}{\sum_{k \in \mathcal{I}} n_k + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E}[U_k | \mathcal{D}]}, \quad (3.14)$$

$$\tau_i \leftarrow \frac{\sum_{k \in \mathcal{I}} \mathbb{E} \left[\sum_{l=1}^{n_k} Y_i^{(k,l)} \middle| \mathcal{D} \right] + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E} \left[\sum_{l=1}^{U_k} Y_i^{(k,l)} \middle| \mathcal{D} \right]}{\sum_{k \in \mathcal{I}} \mathbb{E} \left[\sum_{l=1}^{n_k} Z_i^{(k,l)} \middle| \mathcal{D} \right] + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E} \left[\sum_{l=1}^{U_k} Z_i^{(k,l)} \middle| \mathcal{D} \right]}, \quad (3.15)$$

$$\lambda_{i,j} \leftarrow \frac{\sum_{k \in \mathcal{I}} \mathbb{E} \left[\sum_{l=1}^{n_k} M_{i,j}^{(k,l)} \middle| \mathcal{D} \right] + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E} \left[\sum_{l=1}^{U_k} M_{i,j}^{(k,l)} \middle| \mathcal{D} \right]}{\sum_{k \in \mathcal{I}} \mathbb{E} \left[\sum_{l=1}^{n_k} Z_i^{(k,l)} \middle| \mathcal{D} \right] + \sum_{k \in \bar{\mathcal{I}}} \mathbb{E} \left[\sum_{l=1}^{U_k} Z_i^{(k,l)} \middle| \mathcal{D} \right]}. \quad (3.16)$$

Note that all the expected values are computed by the provisional PH parameters $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau})$.

Next we derive the analytical forms of the expected values in Equations (3.14)–(3.16). Since it is known that the posterior distribution of unobserved samples $\mathbf{U} = \{U_k\}_{k \in \bar{\mathcal{I}}}$ becomes a negative multinomial distribution [49], the expected value of U_k , $k \in \bar{\mathcal{I}}$ becomes the similar expression of the expected value of negative multinomial distribution:

$$\mathbb{E}[U_k | \mathcal{D}] = \frac{N \int_{x_{k-1}}^{x_k} f_{PH}(x) dx}{\sum_{i \in \mathcal{I}} \int_{x_{i-1}}^{x_i} f_{PH}(x) dx}, \quad k \in \bar{\mathcal{I}}, \quad (3.17)$$

where $N = \sum_{k \in \mathcal{I}} n_k$, and $f_{PH}(x)$ is the p.d.f. of PH distribution.

Define the following row and column vectors for a PH random variable X and its phase process J_x :

$$[\mathbf{f}(x)]_i = P(J_x = i), \quad [\mathbf{b}(x)]_i = P(X = x | J_0 = i), \quad (3.18)$$

where $i \neq 0$ and, for the sake of notational convenience, $P(X = x)$ means the p.d.f.

Since $X^{(k,l)}$ are IID samples, the expected value of $B_i^{(k,l)}$ in the observable interval is given by

$$\begin{aligned} \mathbb{E} \left[\sum_{l=1}^{n_k} B_i^{(k,l)} \middle| \mathcal{D} \right] &= n_k \mathbb{E}[B_i | x_{k-1} < X \leq x_k] = \frac{n_k P(J_0 = i, x_{k-1} < X \leq x_k)}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} P(J_0 = i) P(X = x | J_0 = i) dx}{\int_{x_{k-1}}^{x_k} P(X = x) dx} = \frac{n_k \int_{x_{k-1}}^{x_k} \alpha_i [\mathbf{b}(x)]_i dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx}. \end{aligned} \quad (3.19)$$

In the case of unobservable interval, by using the posterior distribution of U_k , the expected value becomes

$$\begin{aligned} \mathbb{E} \left[\sum_{l=1}^{U_k} B_i^{(k,l)} \middle| \mathcal{D} \right] &= \sum_{n=0}^{\infty} n \mathbb{E}[B_i | U_k = n, x_{k-1} < X \leq x_k] P(U_k = n | \mathcal{D}) \\ &= \sum_{n=0}^{\infty} \frac{n \int_{x_{k-1}}^{x_k} \alpha_i [\mathbf{b}(x)]_i dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx} P(U_k = n | \mathcal{D}) = \frac{\mathbb{E}[U_k | \mathcal{D}] \int_{x_{k-1}}^{x_k} \alpha_i [\mathbf{b}(x)]_i dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx} \\ &= \frac{N \int_{x_{k-1}}^{x_k} \alpha_i [\mathbf{b}(x)]_i dx}{\sum_{i \in \mathcal{I}} \int_{x_{i-1}}^{x_i} f_{PH}(x) dx}. \end{aligned} \quad (3.20)$$

Similarly, we have

$$\begin{aligned} \mathbb{E} \left[\sum_{l=1}^{n_k} Y_i^{(k,l)} \middle| \mathcal{D} \right] &= \frac{n_k \int_{x_{k-1}}^{x_k} P(X = x, J_x^- = i) dx}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} P(J_x^- = i) P(J_x = 0 | J_x^- = i) dx}{P(x_{k-1} < X \leq x_k)} = \frac{n_k \int_{x_{k-1}}^{x_k} [\mathbf{f}(x)]_i \tau_i dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx}, \end{aligned} \quad (3.21)$$

$$\mathbb{E} \left[\sum_{l=1}^{U_k} Y_i^{(k,l)} \middle| \mathcal{D} \right] = \frac{N \int_{x_{k-1}}^{x_k} [\mathbf{f}(x)]_i \tau_i dx}{\sum_{i \in \mathcal{I}} \int_{x_{i-1}}^{x_i} f_{PH}(x) dx}, \quad (3.22)$$

where $J_x^- = \lim_{x' \rightarrow 0^+} J_{x-x'}$.

For $Z_i^{(k,l)}$ and $M_{i,j}^{(k,l)}$, the expected values in the observable interval are

$$\begin{aligned} \mathbb{E} \left[\sum_{l=1}^{n_k} Z_i^{(k,l)} \middle| \mathcal{D} \right] &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x P(X = x, J_u = i) du dx}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x P(J_u = i) P(X = x, | J_u = i) du dx}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x [\mathbf{f}(u)]_i [\mathbf{b}(x-u)]_i du dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx}, \end{aligned} \quad (3.23)$$

$$\begin{aligned} \mathbb{E} \left[\sum_{l=1}^{n_k} M_{i,j}^{(k,l)} \middle| \mathcal{D} \right] &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x P(X = x, J_u^- = i, J_u = j) du dx}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x P(J_u^- = i) P(J_u = j | J_u^- = i) P(X = x, | J_u = j) du dx}{P(x_{k-1} < X \leq x_k)} \\ &= \frac{n_k \int_{x_{k-1}}^{x_k} \int_0^x [\mathbf{f}(u)]_i \lambda_{i,j} [\mathbf{b}(x-u)]_j du dx}{\int_{x_{k-1}}^{x_k} f_{PH}(x) dx}. \end{aligned} \quad (3.24)$$

Also the expected values in the unobservable interval become

$$\mathbb{E} \left[\sum_{l=1}^{U_k} Z_i^{(k,l)} \middle| \mathcal{D} \right] = \frac{N \int_{x_{k-1}}^{x_k} \int_0^x [\mathbf{f}(u)]_i [\mathbf{b}(x-u)]_i du dx}{\sum_{i \in \mathcal{I}} \int_{x_{i-1}}^{x_i} f_{PH}(x) dx}, \quad (3.25)$$

$$\mathbb{E} \left[\sum_{l=1}^{U_k} M_{i,j}^{(k,l)} \middle| \mathcal{D} \right] = \frac{N \int_{x_{k-1}}^{x_k} \int_0^x [\mathbf{f}(u)]_i \lambda_{i,j} [\mathbf{b}(x-u)]_j du dx}{\sum_{i \in \mathcal{I}} \int_{x_{i-1}}^{x_i} f_{PH}(x) dx}. \quad (3.26)$$

To compute the expected values, we define the following vectors and matrices based on

$\mathbf{f}(x) = \boldsymbol{\alpha} \exp(\mathbf{T}x)$ and $\mathbf{b}(x) = \exp(\mathbf{T}x)\boldsymbol{\tau}$:

$$\bar{\mathbf{f}}_k = \int_{x_k}^{\infty} \mathbf{f}(x)dx = \boldsymbol{\alpha}(-\mathbf{T})^{-1} \exp(\mathbf{T}t_k), \quad (3.27)$$

$$\tilde{\mathbf{f}}_k = \int_{x_{k-1}}^{x_k} \mathbf{f}(x)dx = \bar{\mathbf{f}}_{k-1} - \bar{\mathbf{f}}_k, \quad (3.28)$$

$$\bar{\mathbf{b}}_k = \int_{x_k}^{\infty} \mathbf{b}(x)dx = \exp(\mathbf{T}x_k)\mathbf{1}, \quad (3.29)$$

$$\tilde{\mathbf{b}}_k = \int_{x_{k-1}}^{x_k} \mathbf{b}(x)dx = \bar{\mathbf{b}}_{k-1} - \bar{\mathbf{b}}_k, \quad (3.30)$$

$$\begin{aligned} \bar{\mathbf{H}}_k &= \int_{x_k}^{\infty} \int_0^u \mathbf{b}(x-u)\mathbf{f}(u)dudx = \int_0^{x_k} \exp(\mathbf{T}(x_k-u))\mathbf{1}\boldsymbol{\alpha} \exp(\mathbf{T}u)du + \mathbf{1}\bar{\mathbf{f}}_k \\ &= \bar{\boldsymbol{\Upsilon}} + \mathbf{1}\bar{\mathbf{f}}_k, \end{aligned} \quad (3.31)$$

$$\tilde{\mathbf{H}}_k = \int_{x_{k-1}}^{x_k} \int_0^u \mathbf{b}(x-u)\mathbf{f}(u)dudx = \bar{\boldsymbol{\Upsilon}}_{k-1} - \bar{\boldsymbol{\Upsilon}}_k + \mathbf{1}\tilde{\mathbf{f}}_k, \quad (3.32)$$

where $\bar{\boldsymbol{\Upsilon}}_k = \int_0^{x_k} \exp(\mathbf{T}(x_k-u))\mathbf{1}\boldsymbol{\alpha} \exp(\mathbf{T}u)du$. By using the above vectors and matrices, the expected values are rewritten as

$$\mathbb{E}[U_k | \mathcal{D}] = \frac{N\boldsymbol{\alpha}\tilde{\mathbf{b}}_k}{\sum_{i \in \mathcal{I}} \boldsymbol{\alpha}\tilde{\mathbf{b}}_i}, \quad (3.33)$$

$$\mathbb{E}\left[\sum_{l=1}^{n_k} B_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{n_k \alpha_i [\tilde{\mathbf{b}}_k]_i}{\boldsymbol{\alpha}\tilde{\mathbf{b}}_k}, \quad \mathbb{E}\left[\sum_{l=1}^{U_k} B_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{N\alpha_i [\tilde{\mathbf{b}}_k]_i}{\sum_{u \in \mathcal{I}} \boldsymbol{\alpha}\tilde{\mathbf{b}}_u}, \quad (3.34)$$

$$\mathbb{E}\left[\sum_{l=1}^{n_k} Y_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{n_k [\tilde{\mathbf{f}}_k]_i \tau_i}{\boldsymbol{\alpha}\tilde{\mathbf{b}}_k}, \quad \mathbb{E}\left[\sum_{l=1}^{U_k} Y_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{N[\tilde{\mathbf{f}}_k]_i \tau_i}{\sum_{u \in \mathcal{I}} \boldsymbol{\alpha}\tilde{\mathbf{b}}_u}, \quad (3.35)$$

$$\mathbb{E}\left[\sum_{l=1}^{n_k} Z_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{n_k [\tilde{\mathbf{H}}_k]_{i,i}}{\boldsymbol{\alpha}\tilde{\mathbf{b}}_k}, \quad \mathbb{E}\left[\sum_{l=1}^{U_k} Z_i^{(k,l)} \middle| \mathcal{D}\right] = \frac{N[\tilde{\mathbf{H}}_k]_{i,i}}{\sum_{u \in \mathcal{I}} \boldsymbol{\alpha}\tilde{\mathbf{b}}_u}, \quad (3.36)$$

$$\mathbb{E}\left[\sum_{l=1}^{n_k} M_{i,j}^{(k,l)} \middle| \mathcal{D}\right] = \frac{n_k \lambda_{i,j} [\tilde{\mathbf{H}}_k]_{j,i}}{\boldsymbol{\alpha}\tilde{\mathbf{b}}_k}, \quad \mathbb{E}\left[\sum_{l=1}^{U_k} M_{i,j}^{(k,l)} \middle| \mathcal{D}\right] = \frac{N\lambda_{i,j} [\tilde{\mathbf{H}}_k]_{j,i}}{\sum_{u \in \mathcal{I}} \boldsymbol{\alpha}\tilde{\mathbf{b}}_u}. \quad (3.37)$$

It can be seen that $\tilde{\mathbf{f}}_k$, $\tilde{\mathbf{b}}_k$, $\tilde{\mathbf{H}}_k$ are replaced by $\bar{\mathbf{f}}_k$, $\bar{\mathbf{b}}_k$, $\bar{\mathbf{H}}_k$, respectively, at $k = K + 1$, because t_{K+1} goes to infinity.

Much computation time of the EM algorithm for PH distribution is spent for the calculation of convolution integral of matrix exponential. In [61, 66], the effective computation algorithm for the convolution integral of matrix exponential was proposed based on the uniformization. Let $\boldsymbol{\Upsilon}(t; \mathbf{v}_1, \mathbf{v}_2)$ be the convolution integral of matrix exponential;

$$\boldsymbol{\Upsilon}(x; \mathbf{v}_1, \mathbf{v}_2) = \int_0^x \exp(\mathbf{T}(x-u))\mathbf{v}_1\mathbf{v}_2 \exp(\mathbf{T}u)du, \quad (3.38)$$

where \mathbf{v}_1 and \mathbf{v}_2 are arbitrary column and row vectors, respectively. Algorithm 1 presents a pseudo code to compute $\boldsymbol{\Upsilon}(x; \mathbf{v}_1, \mathbf{v}_2)$. In this algorithm, ε is an error tolerance for the computation of Poisson probability. The function $\text{PoiBound}(\mu, \varepsilon)$ gives the first point where the c.d.f. of Poisson distribution with mean μ becomes more than $1 - \varepsilon$. The function $\text{PoiProb}(n, \mu)$ is the Poisson probability mass function with mean parameter μ , i.e.,

Algorithm 1 Uniformization-based convolution integral of matrix exponential

```

function convint( $x, \mathbf{v}_1, \mathbf{v}_2$ ) : return value  $\Upsilon$ 
 $r \leftarrow \max(\text{abs}(\text{diag}(\mathbf{T})))$ ;  $\mathbf{P} \leftarrow \mathbf{I} + \mathbf{T}/r$ ;  $R \leftarrow \text{PoiBound}(rx, \varepsilon)$ 
 $\mathbf{v}_1(0) \leftarrow \mathbf{v}_1$ 
for  $i = 1 : R$  do
     $\mathbf{v}_1(i) \leftarrow \mathbf{P}\mathbf{v}_1(i-1)$ 
end for
 $\mathbf{v}_2(R) \leftarrow \text{PoiProb}(R+1, rx)\mathbf{v}_2$ 
for  $i = R-1 : 0$  do
     $\mathbf{v}_2(i) \leftarrow \mathbf{v}_2(i+1)\mathbf{P} + \text{PoiProb}(i+1, rx)\mathbf{v}_2$ 
end for
 $\Upsilon \leftarrow \mathbf{0}$ 
for  $i = 0 : R$  do
     $\Upsilon \leftarrow \Upsilon + \mathbf{v}_1(i)\mathbf{v}_2(i)/r$ 
end for
end function

```

$e^{-\mu}\mu^n/n!$. Also \mathbf{I} means the identity matrix. The time complexity of Algorithm 1 is proportional to a square of the number of phases. In particular, when \mathbf{T} is a sparse matrix, the time complexity is proportional to the number of non-zero elements of \mathbf{T} .

Applying the algorithm to the computation of expected values, we define

$$\mathbf{H} = \sum_{k=1}^{K+1} w_k \tilde{\mathbf{H}}_k, \quad (3.39)$$

where

$$w_k = \begin{cases} n_k/\pi \tilde{\mathbf{b}}_k, & k \in \mathcal{I}, \\ N/\sum_{u \in \mathcal{I}} \pi \tilde{\mathbf{b}}_u, & k \in \bar{\mathcal{I}}. \end{cases} \quad (3.40)$$

From $\tilde{\mathbf{H}} = \bar{\mathbf{H}}_{k-1} - \bar{\mathbf{H}}_k$ and $\bar{\mathbf{H}}_k = \Upsilon(x_k; \mathbf{1}, \boldsymbol{\alpha}) + \mathbf{1}\tilde{\mathbf{f}}_k$, we get

$$\mathbf{H} = \sum_{k=1}^{K+1} w_k \mathbf{1}\tilde{\mathbf{f}}_k + \sum_{k=1}^K \Upsilon(\Delta x_k; \bar{\mathbf{b}}_{k-1}, \mathbf{c}_k), \quad (3.41)$$

where $\Delta x_k = x_k - x_{k-1}$ and

$$\mathbf{c}_k = \sum_{l=k}^K (w_{l+1} - w_l) \boldsymbol{\alpha} \exp(\mathbf{T}(t_l - t_k)). \quad (3.42)$$

Algorithm 2 indicates our E-step procedure for PH distributions under grouped, truncated and missing data according to the above formulas. In this algorithm, the function $\text{convint}(x, \mathbf{v}_1, \mathbf{v}_2)$ denotes the convolution integral of matrix exponential presented in Algorithm 1. By using the results of Algorithm 2, the parameters are updated as $\alpha_i \leftarrow \alpha_i[\mathbf{B}]_i/(N+U)$, $\tau_i \leftarrow \tau_i[\mathbf{Y}]_i/[\mathbf{H}]_{i,i}$ and $\lambda_{i,j} \leftarrow \lambda_{i,j}[\mathbf{H}]_{j,i}/[\mathbf{H}]_{i,i}$.

As mentioned before, any APH distribution can be transformed to three types of canonical forms [16]. In the M-step, the parameters are updated by using the results of E-step;

$$\alpha_i \leftarrow \frac{\alpha_i[\mathbf{B}]_i}{N+U}, \quad \lambda_i \leftarrow \frac{\lambda_i[\mathbf{H}]_{i+1,i}}{[\mathbf{H}]_{i,i}} \quad \text{for } i = 1, \dots, m-1, \quad \lambda_m \leftarrow \frac{\lambda_m[\mathbf{Y}]_m}{[\mathbf{H}]_{m,m}}. \quad (3.43)$$

Algorithm 2 E-step procedure for PH distributions with grouped, truncated and missing data

```

B ← 0; Y ← 0; H ← O
f0 ← α(-T)-1; b0 ← 1; N ← 0; U ← 0
for k = 1 : K do
  fk ← fk-1 exp(TΔxk); fk ← fk-1 - fk; bk ← exp(TΔxk)bk-1; bk ← bk-1 - bk
  if k ∈  $\mathcal{I}$  then
    N ← N + nk; U ← U + αbk
  end if
end for
if K + 1 ∈  $\mathcal{I}$  then
  N ← N + nK+1; U ← U + αbK
end if
for k = 1 : K do
  if k ∈  $\mathcal{I}$  then
    w(k) ← nk/αbk;
  else
    w(k) ← N/U
  end if
  B ← B + w(k)bk; Y ← Y + w(k)fk
end for
if K + 1 ∈  $\mathcal{I}$  then
  w(K + 1) ← nK+1/αbK
else
  w(K + 1) ← N/U
end if
B ← B + w(K + 1)bK; Y ← Y + w(K + 1)fK
cK ← (w(K + 1) - w(K))α
for k = K - 1 : 1 do
  ck ← ck+1 exp(TΔxk+1) + (w(k + 1) - w(k))α
end for
for k = 1 : K do
  H ← H + w(k)1fk + convint(Δxk, bk-1, ck)
end for
H ← H + w(K + 1)1fK

```

Algorithm 3 Re-sorting algorithm

```

for  $i = 1 : (m - 1)$  do
  for  $j = 1 : (m - i)$  do
    if  $\lambda_j > \lambda_{j+1}$  then
       $\alpha_j \leftarrow \alpha_j + \alpha_{j+1}(1 - \lambda_{j+1}/\lambda_j)$ ;
       $\alpha_{j+1} \leftarrow \alpha_{j+1}\lambda_{j+1}/\lambda_j$ ;
      swap( $\lambda_j, \lambda_{j+1}$ );
    end if
  end for
end for

```

However, this procedure does not ensure that the restriction of CF1 holds. Okamura et al. proposed Algorithm 3 as the re-sorting algorithm of $\lambda_1, \dots, \lambda_m$ for the bidiagonal PH distribution. After every M-step execution, Algorithm 3 is performed to ensure the restriction.

3.5. Tools

Several PH fitting software tools are available. *EMpht* [18] is a C program tool for PH fitting with EM algorithms by Asmussen et al. [6] and Olsson [70]. *PhFit* [29, 76] consists of a Java-based interference and computation engines written by C language. This tool deals with continuous and discrete PHs. Furthermore, it can be applied to PH fitting from a p.d.f. *momentmatching* [50] is a set of MATLAB files to execute three moment matching algorithms [71]. *BuTools* [13] are program packages for Mathematica and MATLAB/Octave. The tool provides MAP (Markovian arrival process) fitting as well as PH fitting based on the MM method. *G-FIT* [23] is a command line tool to provide EM algorithms for the hyper-Erlang distribution [72, 82]. *jPhaseFit* [38] is a library for Java to handle PHs. In this library, both of PH fitting algorithms with MM and EM algorithm are implemented. In particular, EM algorithm for hyper-Erlang distribution [82] was implemented in this tool. *HyperStar* [31] is a Java-based GUI tool to estimate hyper-Erlang distribution and to plot graphs. It implements the cluster-based algorithm [77]. *mapfit* [47, 62] is a package of R for PH/MAP fitting, which is distributed by CRAN. This package provides the fast EM algorithms for PH/MAP [61, 66] and PH/MAP fitting with grouped data [65, 67].

4. Phase-Type Software Reliability Model

In this section, we present the phase-type software reliability model (PH-SRM) proposed by [59, 60, 63] as an application example of PH fitting in the reliability engineering. The software reliability model (SRM) is a well-known stochastic model for estimating the number of bugs and software reliability quantitatively [46, 51, 52, 74]. It is also used to estimate the number of remaining bugs before/after testing with the bug detection data collected in testing process. During the last four decades, non-homogeneous Poisson processes (NHPPs) based SRMs have gained much popularity for modeling the behavior of software-fault detection process. The NHPP-based SRMs are mathematically tractable, and then a number of NHPP-based SRMs have been proposed in the literature. Goel and Okumoto [26], Littlewood [44, 45], Yamada et al. [90], Musa and Okumoto [53], Ohba [58], Goel [25], Laprie et al. [43] and Gokhale and Trivedi [27] presented typical NHPP-based SRMs. Even in recent years, many researchers still proposed NHPP-based SRMs with different mean value functions under different modeling scenarios such as imperfect debugging, change points, testing efforts and fault correction process [30, 32, 64, 73, 88].

Table 2: Relationship between fault detection time distributions and NHPP-based SRMs

Distribution	NHPP-based SRM
Exponential	Goel-Okumoto model [26]
2-stage Erlang	Delayed S-shaped model [90]
k -state Erlang	k -stage Erlangian NHPP [39, 94]
Pareto	Littlewood [44, 45]
Weibull	Goel [25]
Gompertz	Yamada [89]
Hyperexponential	Hyper-exponential model [43]
Hypoexponential	Time-domain dependent model [22]
Truncated logistic	Infection S-shaped model [58]
Log-logistic	Gokhale and Trivedi [27]
Truncated normal	Okamura et al. [64]
Log-normal	Achcar et al. [1]

4.1. NHPP-based SRM

Langberg and Singpurwalla [42] showed that almost all of the NHPP-based SRMs follow a simple debugging scenario, which causes a unified modeling framework for NHPP-based SRMs. Suppose that

(A-1) The total number of software faults is a Poisson random variable with mean ω .

(A-2) All of the software fault detection times are IID random variables having a c.d.f. $F(x)$.

Let $M(x)$ be the cumulative number of software faults before time x . Provided that the total number of software faults is given by N , the number of faults detected before time t follows the binomial distribution with success probability $F(x)$. Then we have

$$P(M(x) = y | M(\infty) = N) = \binom{N}{y} F(x)^y \bar{F}(x)^{N-y}, \quad (4.1)$$

where $\bar{F}(x) = 1 - F(x)$. Since N is a Poisson random variable with mean ω , the probability mass function (p.m.f.) of $M(x)$ is given by

$$\begin{aligned} P(M(x) = y) &= \sum_{N=y}^{\infty} e^{-\omega} \frac{\omega^N}{N!} \binom{N}{y} F(x)^y \bar{F}(x)^{N-y}, \\ &= \frac{(\omega F(x))^y}{y!} e^{-\omega F(x)}. \end{aligned} \quad (4.2)$$

The above p.m.f. is equivalent to the NHPP with mean value function $\omega F(x)$ [42].

Equation (4.2) implies that each of NHPP-based SRMs can be characterized by a particular fault detection time distribution. Table 2 presents relationship between typical fault detection time distributions and corresponding NHPP-based SRMs.

4.2. PH-SRM

Okamura and Dohi [59, 60, 63] proposed PH-SRM (phase-type software reliability model) by substituting the PH distribution to the fault detection time distribution in NHPP-based

model. Then the p.m.f. of $M(t)$ is given by

$$P(M(x) = y) = \frac{(\omega F_{PH}(x))^y}{y!} e^{-\omega F_{PH}(x)}, \quad (4.3)$$

$$F_{PH}(x) = 1 - \boldsymbol{\alpha} \exp(\mathbf{T}x)\boldsymbol{\tau}. \quad (4.4)$$

Since PH distribution can express any distribution approximately, PH-SRM can also express all of the NHPP-based SRMs. Okamura and Dohi [59, 60, 63] considered two specific PH-SRM; CPH-SRM and HEr-SRM that use CF1 and hyper-Erlang distribution, respectively. The fundamental procedure of software reliability assessment with NHPP-based SRMs is (i) to collect the fault data in testing phase, (ii) to estimate model parameters from the collected data, (iii) to compute the reliability measures from the estimated models. Since PH-SRM involves many parameters, we have to say that Step (ii) is not an easy task. Okamura and Dohi [63] developed the EM algorithm for PH-SRM when grouped fault detection data are given.

Let $\mathcal{D} = \{(x_1, y_1), \dots, (x_K, y_K)\}$ be grouped data, where y_i are the cumulative number of faults in time interval $[0, x_i)$. Here we define $0 < X_1 < X_2 < \dots < X_N$ as all of ordered fault detection times, where N is the number of total faults detected in $[0, \infty)$. Note that X_i and N are unobserved variables. Based on the EM principle, we formulate EM-step of PH-SRM as follows.

$$\omega \leftarrow E[N|\mathcal{D}], \quad (4.5)$$

$$(\boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\tau}) \leftarrow E \left[\sum_{k=1}^N \log f_{PH}(X_{[k]}) \middle| \mathcal{D} \right]. \quad (4.6)$$

By applying the result of [69] to Equation (4.5), we have

$$\omega \leftarrow y_K + \omega \bar{F}_{PH}(x_K). \quad (4.7)$$

On the other hand, Equation (4.6) implies the EM-step for ordinary PH distribution with grouped and truncated data, provided that the total number of faults follows a Poisson distribution. Since N/U is the expected total number of events in Algorithm 2, we obtain the E-step of Equation (4.6) by replacing N/U with ω in Algorithm 2.

In the PH-SRM, the data fitting ability depends on the number of phases. As the number of phases increases, the data fitting ability of PH-SRM also increases. On the other hand, the overfitting problem may arise when the number of phases is excessively large. Information criteria offer plausible measures to prevent the overfitting. The most well-known information criterion is AIC [2], which is defined by

$$\text{AIC} = -2(\text{maximum log-likelihood}) + 2(\text{degrees of freedom}), \quad (4.8)$$

where the degrees-of-freedom is generally identical to the number of free model parameters. The model with the smaller AIC can be regarded as the better model. Although AIC is quite convenient for the model selection problem, we must carefully treat the definition of degrees-of-freedom. In particular, when dealing with the models involving redundant parameters like mixture models, the degrees-of-freedom is not identical to the number of free parameters. The PH-SRM is the case. It should be noted that there are some open problems to be solved for degrees-of-freedom of PH distribution.

Table 3: Data fitting results

Model	MSE	MLL	DF	AIC	CTIME
GO	990.20	-359.88	2	723.76	—
ISS	295.33	-317.93	3	641.85	—
Q-R ISS	230.60	-296.44	5	602.88	—
Beta-Mix	278.47	-327.65	2	659.30	—
31-CPH-SRM	44.10	-246.43	23	538.86	1.2
20-HEr-SRM	275.22	-302.10	8	620.20	237.0

4.3. Numerical experiment

We compare PH-SRMs with two NHPP-based SRMs in terms of data fitting ability. Many researchers proposed NHPP-based SRMs under different modeling scenarios. Hwang and Pham [30] proposed an NHPP-based SRM under the scenario that the fault removal is delayed. Based on this scenario, they presented a piecewise continuous mean value function. Kim et al. [40] also discussed a different NHPP-based SRM from the statistical point of view. They incorporated the Beta distribution to represent unknown probability in the model, and unified both finite and infinite failure models. In the above papers, the authors examined the data fitting ability with the exactly same data, and concluded that both models were superior to other well-known SRMs such as Goel-Okumoto model [26], inflection S-shaped model [58], and some others [91], [75], [93]. Here we demonstrate the data fitting ability of our PH-SRM by compared with these latest SRMs under the same data set used in [30, 40]. The data was originally reported by Tohma et al. [83], which consists of 111 observations of the number of detected faults (grouped data) in actual software testing. The tested programs were for the monitoring and real-time control system with about 200 modules, where each module has around 1000 lines of code. Hwang and Pham [30] and Kim et al. [40] derived ML estimates of their SRMs for this data set. The estimated model parameters can be cited in [30, 40].

To evaluate the data fitting ability of our PH-SRM, we estimate model parameters of CPH-SRM and HEr-SRM with the same data. In PH-SRM, we determined the best number of phases from the view point of AIC, and chose 31-CPH-SRM as the best model from 2-CPH-SRM through 50-CPH-SRM. On the other hand, the computation time of parameter estimation for HEr-SRM exponentially increases with the number of phases. Thus, in the case of HEr-SRM, we examined AICs for up to 20-HEr-SRM. As a result, 20-HEr-SRM was selected as the best one among them.

Table 3 presents the summary of fitting results for all the SRMs. GO and ISS denote Goel-Okumoto model [26] and inflection S-shaped model [58], respectively. Q-R ISS means the quasi-renewal time-delay model based on inflection S-shaped mean value function in [30], which was the best model on the original paper [30]. Beta-Mix is an NHPP-based SRM with beta mixture proposed by [40]. Although the above paper just focused on the simplest two mixture models, we chose the best mixture model among others. As the data fitting criteria, we compute mean squared error (MSE), maximum log-likelihood (MLL), degrees of freedom (DF) and AIC [‡]. Note that the definition of MSE here is also different from [30, 40],

[‡]Note that AIC of Q-R ISS is different from the original paper [30]. The AIC computed in [30] seems to be incorrect.

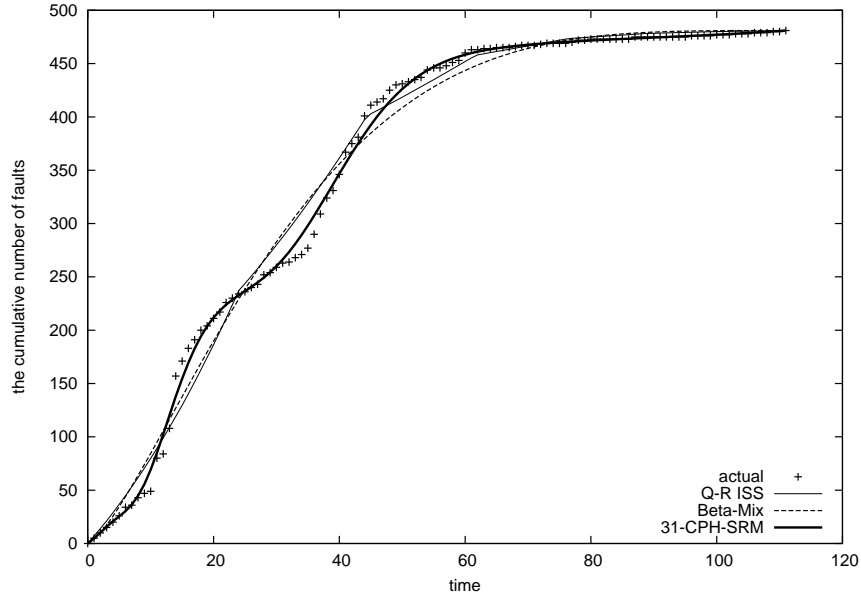


Figure 3: The estimated mean value functions

so, for the grouped data $\{(x_1, y_1), \dots, (x_K, y_K)\}$, MSE is computed by

$$\text{MSE} = \frac{1}{K} \sum_{k=1}^K \left(\hat{\Lambda}(x_k) - y_k \right)^2, \quad (4.9)$$

where $\hat{\Lambda}(x)$ is the estimated mean value function of NHPP-based SRM. The column CTIME indicates the computation time to obtain ML estimates for 31-CPH-SRM and 20-HEr-SRM. Figure 3 also illustrates the estimated mean value functions for Q-R ISS, Beta-Mix and 31-CPH-SRM. From the results, 31-CPH-SRM has extremely high data fitting ability compared to others. In fact, although DF of 31-CPH-SRM is higher than others, the mean behavior of 31-CPH-SRM is very close to the actual data in Figure 3. The same observation can be obtained from the result of MSE. This is an evidence that the SRM with high degrees of freedom requires to represent the fault detection processes with many change points on the data trend. Although PH-SRM is built by applying PH distribution to software fault detection time distribution in the framework of Langberg and Singpurwalla [42], PH-SRM is not based on any modeling scenario dissimilar to Q-R ISS, and is superior to the others in terms of data fitting ability. From this simple example, we understand that it is enough to select an appropriate fault detection time distribution in NHPP-based software reliability modeling.

5. PH Approximation for Markov Regenerative Process

Markov modeling is one of the most important techniques to evaluate reliability and performance of systems. A Markov regenerative process (MRGP) is known as one of the widest class of stochastic counting processes that are mathematically tractable. MRGP consists of several discrete states and time sequence of state transitions, and is an extension of both CTMC and the renewal process. Since MRGP allows state transitions triggered by general distributions, it is often used to analyze stochastic Petri nets. The Markov regenerative stochastic Petri net (MRSPNs), which are governed by MRGP, are heavily used in computer science [14, 24, 41]. In this section, we discuss the transient analysis of MRGP by

applying PH approximation. The fundamental idea is to replace general distributions of MRGP with PH distributions that approximate the general distributions. Since the original MRGP is well approximated by a time-homogeneous CTMC, we can derive numerically transient probabilities with commonly-used techniques for CTMC such as uniformization. In particular, this section presents a systematic way to approximate MRGP with a CTMC via PH approximation.

5.1. Markov regenerative process

Consider a stochastic process $\{S(t); t \geq 0\}$ with discrete state space. If $S(t)$ has time points at which the process probabilistically restarts itself, the process is called regenerative. When state transitions at the regeneration points are governed by a DTMC, the process $S(t)$ is an MRGP. Define a regeneration time sequence $T_1 < T_2 < \dots$ and their time intervals $\Delta T_i = T_i - T_{i-1}$, $i = 1, 2, \dots$. Then the time intervals can be represented by a Markov renewal sequence [15]. Suppose that the time sequence is time-homogeneous, i.e.,

$$\begin{aligned} P(S(T_n) = j, \Delta T_n < t \mid S(T_{n-1}) = i) \\ = P(S(T_1) = j, \Delta T_1 < t \mid S(T_0) = i) \equiv K_{i,j}(t). \end{aligned} \quad (5.1)$$

The state probability of MRGP is given by

$$\begin{aligned} V_{i,j}(t) &= P(S(t) = j \mid S(0) = i) \\ &= P(S(t) = j, \Delta T_1 \leq t \mid S(0) = i) + P(S(t) = j, \Delta T_1 > t \mid S(0) = i) \\ &= \sum_l \int_0^t P(S(t-u) = j \mid S(0) = l) dK_{i,l}(u) \\ &\quad + P(S(t) = j, \Delta T_1 > t \mid S(0) = i). \end{aligned} \quad (5.2)$$

In general, $\mathbf{K}(t)$, $\mathbf{V}(t)$ and $\mathbf{E}(t)$ are matrices whose elements are given by $K_{i,j}(t)$, $V_{i,j}(t)$ and $P(S(t) = j, \Delta T_1 > t \mid S(0) = i)$, respectively. Then we have the Markov renewal equation for MRGP [14, 21];

$$\mathbf{V}(t) = \mathbf{E}(t) + \int_0^t d\mathbf{K}(u)\mathbf{V}(t-u), \quad (5.3)$$

where $\mathbf{E}(t)$ and $\mathbf{K}(t)$ are called local and global kernels.

To apply PH approximation, we consider the following MRGP whose space space is finite and each state is classified to subspaces:

- The state space is separated into subspaces \mathcal{S}^E and $\mathcal{S}_1^G, \dots, \mathcal{S}_K^G$.
- The subspace \mathcal{S}^E (EXP state) is a set of the states in which there exist only exponential (EXP) transitions. \mathbf{C}_0 is defined as an infinitesimal generator of CTMC for EXP transitions in the subspace \mathcal{S}^E . Also $\mathbf{C}_1, \dots, \mathbf{C}_K$ are transition rate matrices for EXP transitions that transfer the state from the subspace \mathcal{S}^E to the subspaces $\mathcal{S}_1^G, \dots, \mathcal{S}_K^G$, respectively.
- The subspace \mathcal{S}_i^G (GEN states) is a set of the states where there exist both EXP transitions and general (GEN) transitions that are triggered by general distributions.
 - In $\mathcal{S}_1^G, \dots, \mathcal{S}_K^G$, there is no EXP transition that transfers the state to other subspaces, i.e., all the EXP transitions of \mathcal{S}_i^G are the transitions between two states of \mathcal{S}_i^G . \mathbf{Q}_i denotes the infinitesimal generator of EXP transitions of \mathcal{S}_i^G .
 - In \mathcal{S}_i^G , there are M_i GEN transitions that are triggered by general distributions; $G_i^{(l)}(t)$, $l = 1, \dots, M_i$. These transitions are competitive. That is, when one GEN transition occurs, the ages of the other distributions are renewed.

- The GEN transitions have respective probability matrices to determine the outgoing state just after the transition occurs. The probability matrices are defined by $\mathbf{P}_{i,0}^{(l)}, \dots, \mathbf{P}_{i,K}^{(l)}$. The subscript corresponds to the state of subspaces \mathcal{S}^E (state 0), $\mathcal{S}_1^G, \dots, \mathcal{S}_K^G$.

In this section, we refer to the MRGPs where each state has only one general transition and two or more transitions as single-GEN and multiple-GEN, respectively.

5.2. PH approximation

In this section, we consider the KL-based PH approximation. As mentioned in Section 3, the KL-based PH approximation is reduced to ML estimation of PH distribution with weighted IID samples, i.e.,

$$\hat{\boldsymbol{\theta}} \leftarrow \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^n w_i \log f_{PH}(x_i; \boldsymbol{\theta}), \quad (5.4)$$

where $(x_1, w_1), (x_2, w_2), \dots, (x_K, w_K)$ are weighted IID samples. If we obtain the weighted IID samples, the PH parameters are obtained from the EM algorithm for PH distribution efficiently. However, since the computation time of EM algorithm depends on the number of samples, the number of samples is preferred to be as small as possible.

This article uses the double exponential (DE) formula [79] to generate the weighted IID samples. This method achieves the highly-accurate quadrature with relatively small integration points. The DE formula is based on the fact that the trapezoidal quadrature is the best in terms of precision for the functions whose tails are expressed by double exponential functions. Thus the first step of DE formula is to change the original integration to an infinite integration of the function which decays according to double exponential function. Concretely, we can use the following function in the case of PH approximation:

$$\phi(x) = \exp\left(\frac{\pi}{2}(x - e^{-x})\right). \quad (5.5)$$

By substituting the above function to $\int_0^\infty f(x) \log f_{PH}(x) dx$, the integration is transformed to

$$\int_0^\infty f(x) \log f_{PH}(x) dx = \int_{-\infty}^\infty f(\phi(x)) \log f_{PH}(\phi(x)) \phi'(x) dx, \quad (5.6)$$

where $\phi'(x)$ is the first derivative of $\phi(x)$. The second step of DE formula is to apply the trapezoidal rule to the above integration. Then we have

$$\int_{-\infty}^\infty f(\phi(x)) \log f_{PH}(\phi(x)) \phi'(x) dx \approx \sum_{i=K^-}^{K^+} h \phi'(ih) f(\phi(ih)) \log f_{PH}(\phi(ih)), \quad (5.7)$$

where h is a step size and $K^+ (= -K^-)$ is a upper (lower) limit of integration points. According to Equation (5.7), the weighted samples $(x_1, w_1), \dots, (x_n, w_n)$ can be generated by

$$x_{i-K^-+1} = \phi(ih), \quad (5.8)$$

$$w_{i-K^-+1} = h \phi'(ih) f(\phi(ih)), \quad i = K^-, \dots, 0, \dots, K^+, \quad (5.9)$$

where $n = K^+ - K^- + 1$. Based on the above weighted samples, the EM algorithm for PH distribution is applied to obtain PH parameters. Also the accuracy of PH approximation

can be obtained the KL divergence. When we apply the DE formula, the KL divergence is given by

$$KL(f, f_{PH}) = \sum_{i=1}^K w_i \log f(x_i) - \max_{\theta} \sum_{i=1}^K w_i \log f_{PH}(x_i; \theta). \quad (5.10)$$

Finally we derive a Kronecker representation for the approximate MRGP. To obtain the Kronecker representation, we focus on the fact that regeneration point processes becomes PH renewal processes via the PH approximation. Suppose that the general distribution $G_i(t)$ approximates a PH distribution with $(\alpha_i, \mathbf{T}_i, \tau_i)$. Let $M(t)$ be the number of regenerations experienced before time t . Then a pair of $(M(t), S(t))$ becomes a MAP (Markovian arrival process) under the approximate MRGP. Define \mathbf{D}_0 and \mathbf{D}_1 as infinitesimal generators of the MAP without and with arrivals, respectively. By taking superposition of the non-regeneration point process and the PH renewal processes, they can be expressed as

$$\mathbf{D}_0 = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 \otimes \alpha_1 & \cdots & \mathbf{C}_K \otimes \alpha_K \\ & \mathbf{Q}_1 \oplus \mathbf{T}_1 & & \\ & & \ddots & \\ & & & \mathbf{Q}_k \oplus \mathbf{T}_K \end{pmatrix}, \quad (5.11)$$

$$\mathbf{D}_1 = \begin{pmatrix} \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{P}_{1,0} \otimes \xi_1 & \mathbf{P}_{1,1} \otimes \Lambda_{1,1} & \cdots & \mathbf{P}_{1,K} \otimes \Lambda_{1,K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{K,0} \otimes \xi_K & \mathbf{P}_{K,1} \otimes \Lambda_{K,1} & \cdots & \mathbf{P}_{K,K} \otimes \Lambda_{K,K} \end{pmatrix}, \quad (5.12)$$

where $\Lambda_{i,j} = \tau_i \alpha_j$. The state process $S(t)$ is represented by a CTMC with infinitesimal generator $\mathbf{D}_0 + \mathbf{D}_1$. Then, the transient solution of $S(t)$ can be obtained from the commonly-used technique for CTMC such as the uniformization. In the case of multiple-GEN, the Kronecker representation can be rewritten:

$$\mathbf{C}_i \otimes \alpha_i \rightarrow \mathbf{C}_i \otimes \alpha_i^{(1)} \otimes \cdots \otimes \alpha_i^{(l)}, \quad (5.13)$$

$$\mathbf{Q}_i \oplus \mathbf{T}_i \rightarrow \mathbf{Q}_i \oplus \mathbf{T}_i^{(1)} \oplus \cdots \oplus \mathbf{T}_i^{(l)}, \quad (5.14)$$

$$\begin{aligned} \mathbf{P}_{i,j} \otimes \tau_i &\rightarrow \mathbf{P}_{i,j}^{(1)} \otimes \tau_i^{(1)} \otimes \mathbf{1} \otimes \cdots \otimes \mathbf{1} \\ &+ \mathbf{P}_{i,j}^{(2)} \otimes \mathbf{1} \otimes \tau_i^{(2)} \otimes \cdots \otimes \mathbf{1} \\ &\vdots \\ &+ \mathbf{P}_{i,j}^{(l)} \otimes \mathbf{1} \otimes \mathbf{1} \otimes \cdots \otimes \tau_i^{(l)}, \end{aligned} \quad (5.15)$$

where $\alpha_i^{(l)}$, $\mathbf{T}_i^{(l)}$ and $\tau_i^{(l)}$ are the parameters of PH approximation for $G_i^{(l)}(t)$.

5.3. Numerical experiment

In the numerical experiment, we first investigate the accuracy of KL-based PH approximation from the p.d.f. information, where the PH fitting for Weibull (WEI) and log-normal (LOG) distributions are conducted. For simplicity, suppose that all the first moments are given by unity. Given the squared coefficient of variation (CV^2), we determine the parameters of the above probability distributions. Table 4 presents CV^2 and the third moment

Table 4: Distribution types, moments and the number of weighted samples

dist.	CV ²	3rd	# of samples
WEI1	0.1	1.3008	238
WEI2	2.0	16.4203	600
LOG1	0.1	1.3310	182
LOG2	2.0	27.0000	571

Table 5: PH fitting results

	# of phases	CV ²	3rd	time (sec.)
WEI1	10	0.1168	1.3695	2.96
	50	0.1001	1.3013	5.39
	100	0.1000	1.3009	10.43
	200	0.1000	1.3008	26.36
WEI2	10	1.9851	15.9087	10.37
	50	1.9977	16.3110	59.86
	100	1.9983	16.3349	158.55
	200	1.9999	16.4147	395.07
LOG1	10	0.1000	1.3201	3.61
	50	0.1000	1.3311	7.02
	100	0.1000	1.3310	18.55
	200	0.1000	1.3310	55.55
LOG2	10	1.9418	21.9953	63.07
	50	1.9801	24.4015	119.25
	100	1.9481	22.2787	350.49
	200	1.9811	24.4760	748.60

for each distribution function used in this experiment. The weighted samples are generated with the DE formula with $K = 5000$ and $h = 0.01$. Since the number of samples directly affects the computation of PH fitting, we remove the samples whose weights are less than $1.0e-13$. Table 4 also gives the number of generated weighted samples for each distribution function. Table 5 presents the result of EM algorithm with the generated weighted samples. This table shows CV² and third moments of approximate PH distributions. The computation times are measured in seconds on Intel CoreDuo 2.0GHz. It can be seen that the approximate moments get close to the theoretical ones as the number of phases increases. This observation implies that the number of phases is an important factor to guarantee the accuracy of approximation.

Next we give a reliability example to use PH approximation of MRGP. Consider a parallel system with two machines A and B. Suppose that two machines share a single repair facility working on FCFS (first-come first-serve) schedule, and that the failure rates of machine A and machine B are constant, i.e., the failure times are exponentially distributed with mean times $1/\lambda_A$ and $1/\lambda_B$, respectively. When either machine A or B fails, the failed machine undergoes repair operation. The times for repairing machines A and B obey general distributions $G_A(t)$ and $G_B(t)$, respectively. The system is available whenever at least one machine is operative. Based on this simple model, we compute the pointwise (transient) system availability defined by the probability that the system is available at a single time

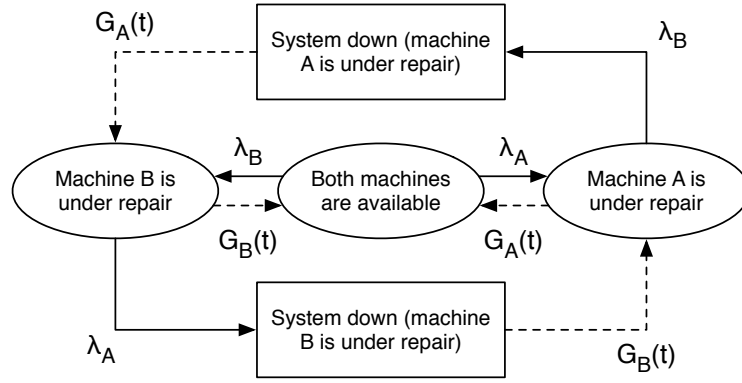


Figure 4: Transition diagram of parallel system with single repair facility

point.

Figure 4 depicts the state transition diagram for the parallel system with single repair facility [21]. The dotted arrows indicate regenerative transitions. This model contains two different regenerative transitions denoted by $G_A(t)$ and $G_B(t)$, and is represented by a single-GEN MRGP. Then the parameters corresponding to the structured MRGP are given by

$$\mathbf{C}_0 = -(\lambda_A + \lambda_B), \quad \mathbf{C}_1 = (\lambda_A, 0), \quad \mathbf{C}_2 = (\lambda_B, 0), \quad (5.16)$$

$$\mathbf{Q}_1 = \begin{pmatrix} -\lambda_B & \lambda_B \\ 0 & 0 \end{pmatrix}, \quad (5.17)$$

$$\mathbf{P}_{1,0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{1,1} = \mathbf{O}, \quad \mathbf{P}_{1,2} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad (5.18)$$

$$\mathbf{Q}_2 = \begin{pmatrix} -\lambda_A & \lambda_A \\ 0 & 0 \end{pmatrix}, \quad (5.19)$$

$$\mathbf{P}_{2,0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{2,2} = \mathbf{O}, \quad \mathbf{P}_{2,1} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \quad (5.20)$$

Note that the MRGP becomes the multiple-GEN case if the failure time distributions are general distributions.

Since the behavior of pointwise system availabilities depends on the initial state of system, we consider two kinds of initial states of system:

- Init 1: both machines are available.
- Init 2: one machine fails.

Figures 5 through 12 show the behavior of point availabilities where the repair distribution is assumed to be WEI1, WEI2, LOG1 or LOG2. Throughout this example, we assume that $\lambda_A = 0.1$ and $\lambda_B = 0.1$ and that an identical distribution with mean 1 for $G_A(t)$ and $G_B(t)$. It is also assumed that the number of phases for KL-based PH approximation (KL-PH) is fixed as 10, 50, 100 or 200. To compare with KL-based PH approximation, we conduct the moment-based PH approximation (MM) [10]. In all of the figures, we observe that the approximate pointwise availabilities take almost same values as the number of phases increases. Since there is no remarkable difference between the approximate pointwise availability values with 100 and 200 phases in all the results, we conclude that the pointwise availability values with 100 and 200 phases are almost similar to the exact one. In the cases of WEI2 and LOG2, the results of moment-based PH approximation are far from

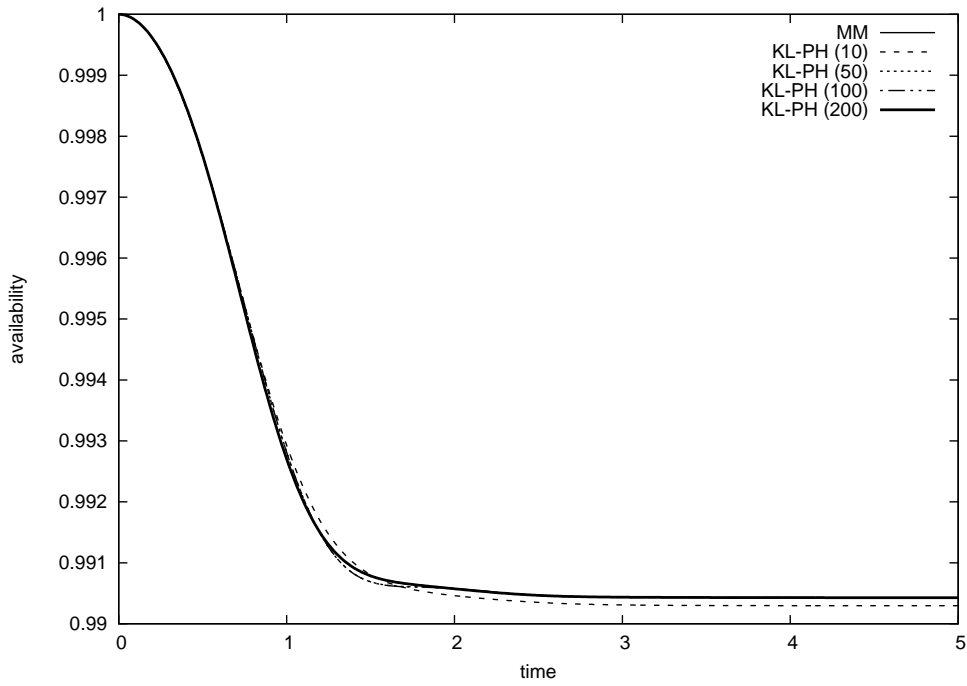


Figure 5: Pointwise system availabilities for parallel system (Init 1 & WEI1)

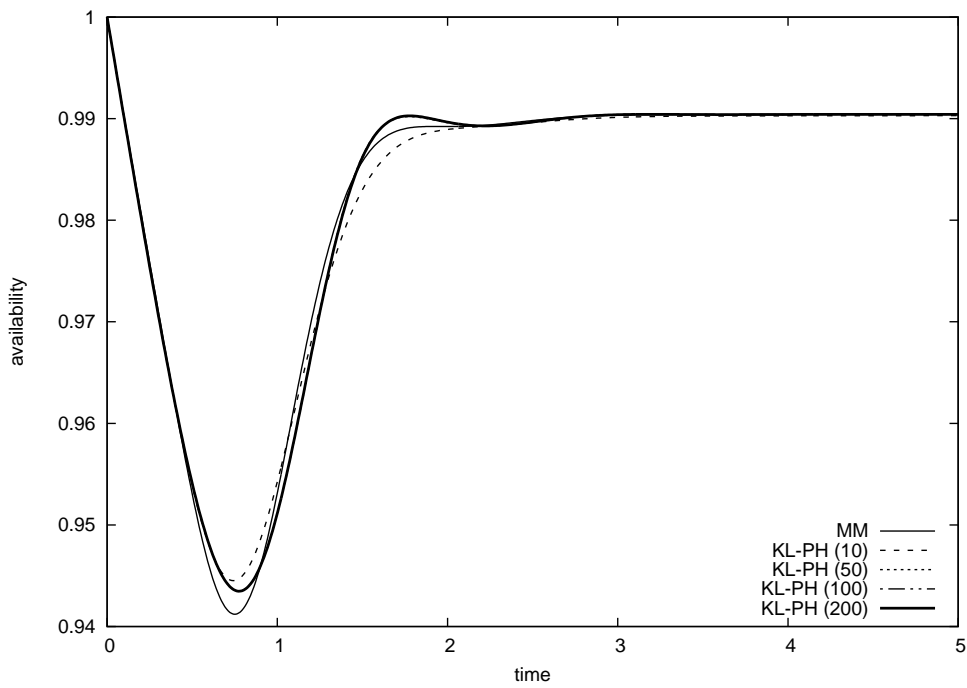


Figure 6: Pointwise system availabilities for parallel system (Init 2 & WEI1)

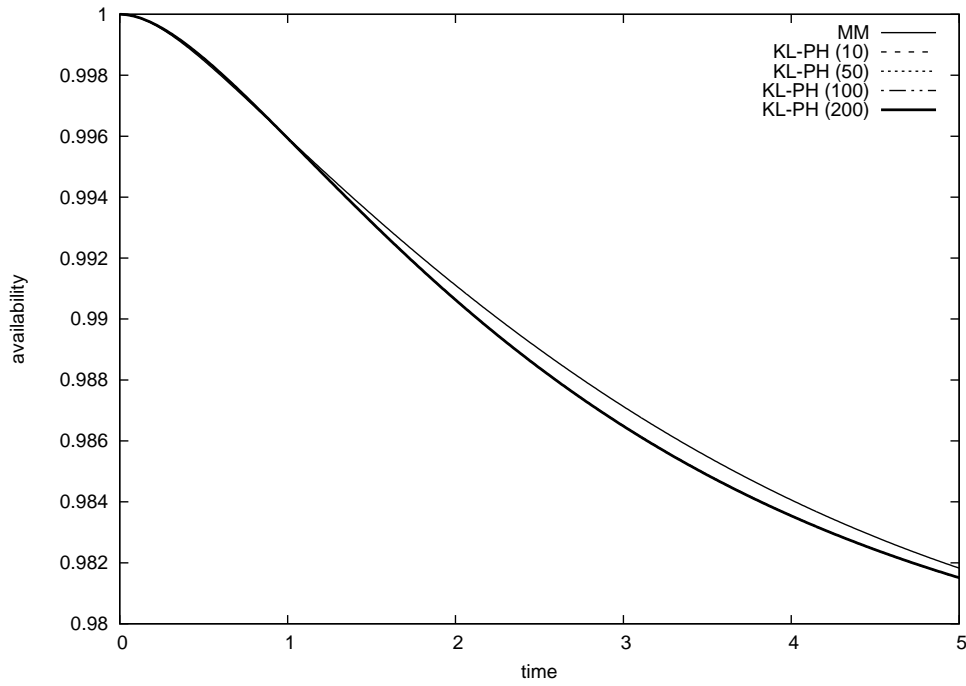


Figure 7: Pointwise system availabilities for parallel system (Init 1 & WEI2)

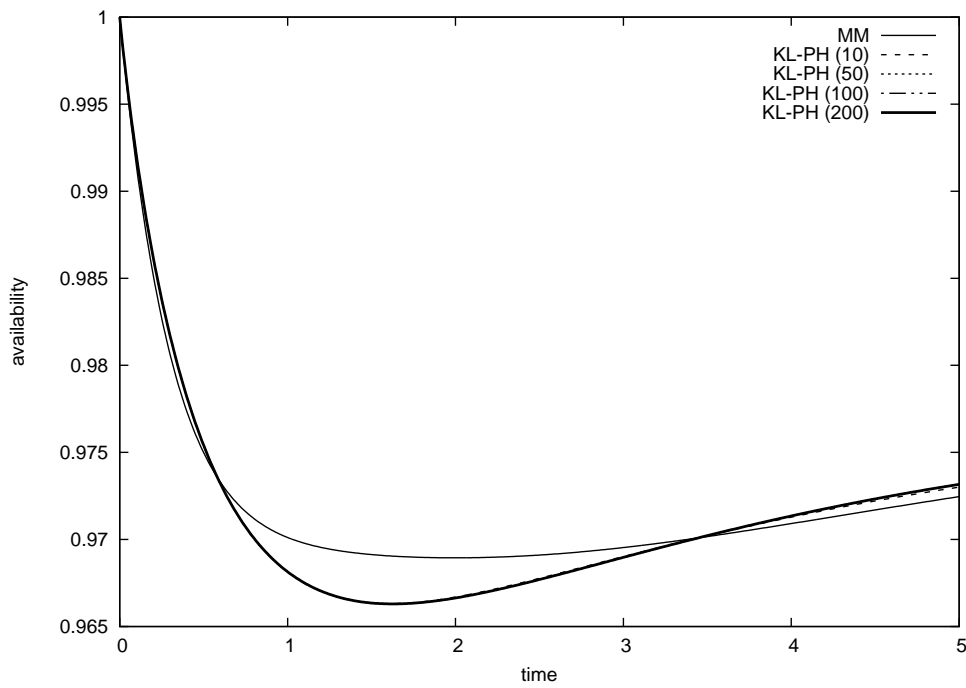


Figure 8: Pointwise system availabilities for parallel system (Init 2 & WEI2)

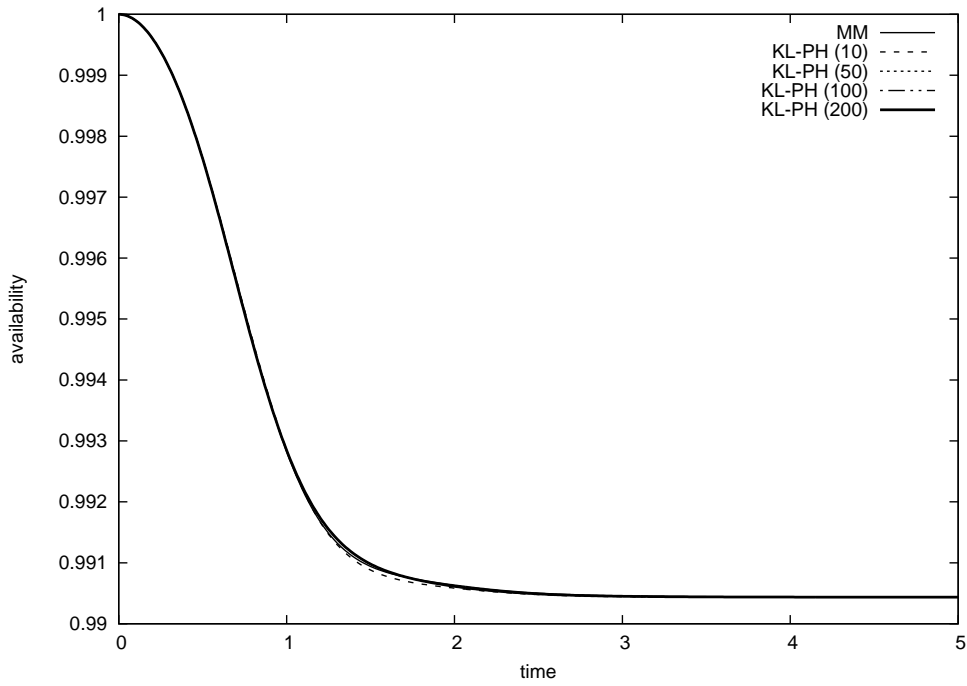


Figure 9: Pointwise system availabilities for parallel system (Init 1 & LOG1)

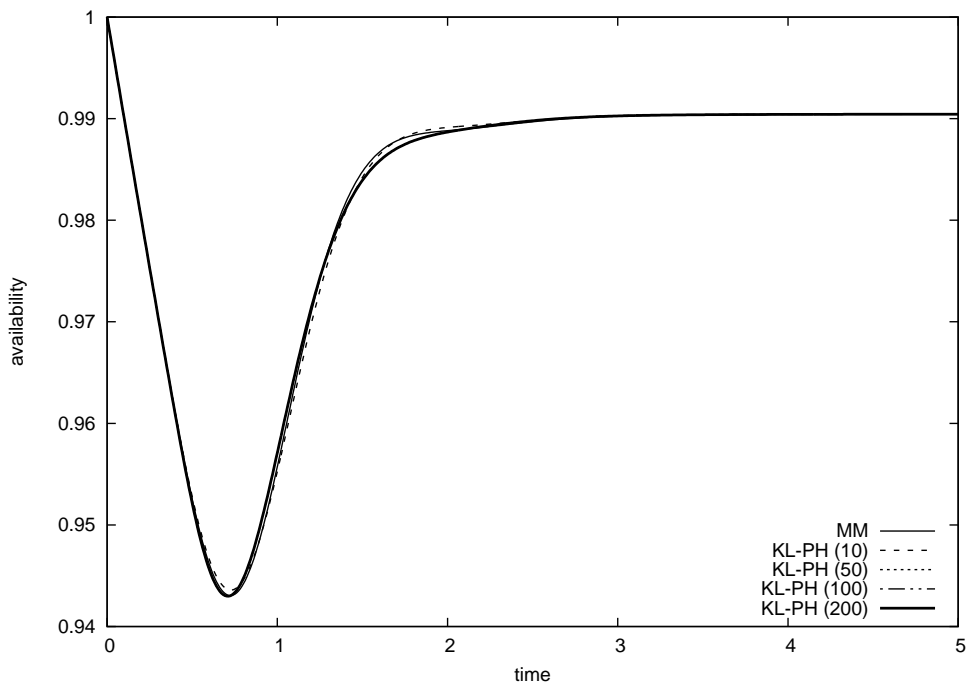


Figure 10: Pointwise system availabilities for parallel system (Init 2 & LOG1)

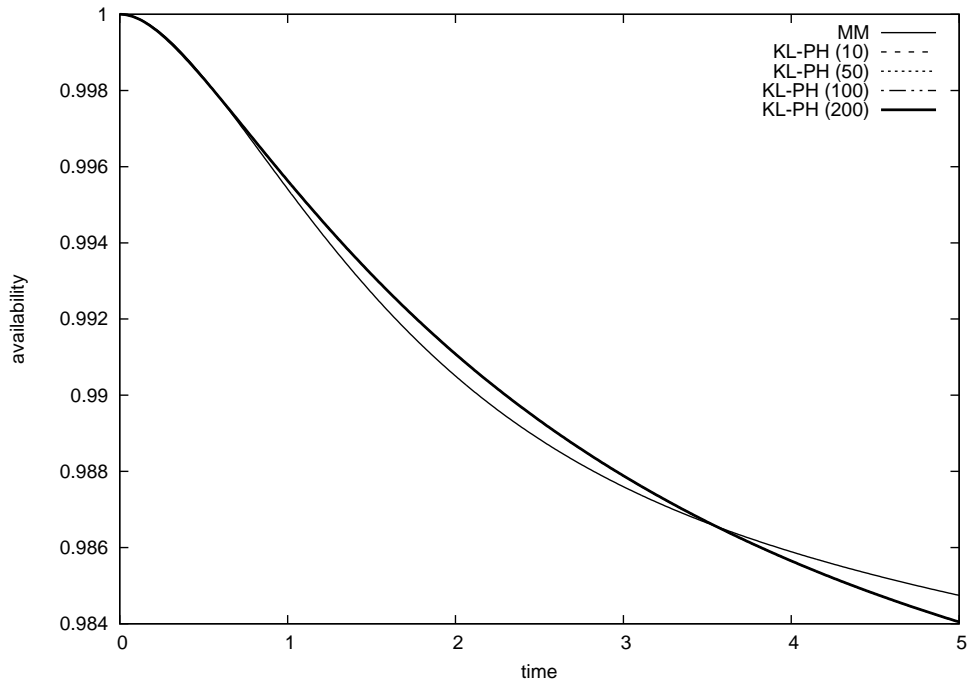


Figure 11: Pointwise system availabilities for parallel system (Init 1 & LOG2)

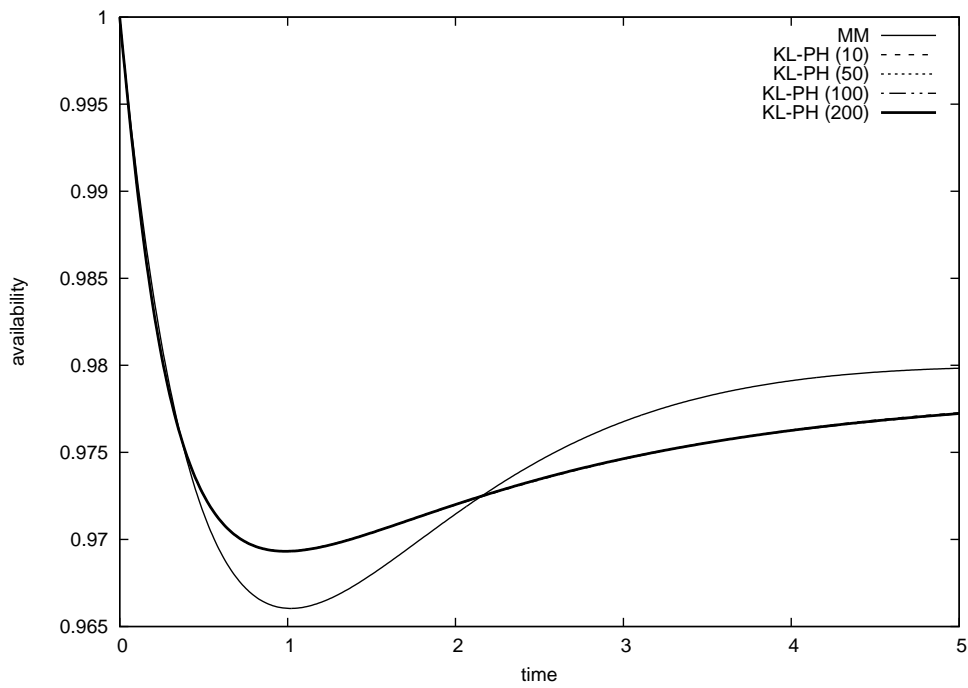


Figure 12: Pointwise system availabilities for parallel system (Init 2 & LOG2)

those of KL-based PH approximation. This implies that the accuracy of moment-based PH approximation is insufficient to analyze the transient solution. In particular, since WEI2 and LOG2 have relatively large CVs, the KL-based PH approximation is more efficient in these cases.

6. Conclusion

In this article, we have summarized the PH estimation and PH approximation based on KL divergence. We have also shown the efficiency of our algorithm in numerical experiments with two reliability examples. The great advantage of PH distribution is to approximate arbitrary probability distribution with positive support. In the first example on software reliability modeling, we have built PH-SRM which provided the best goodness-of-fit performance among a large number of NHPP-based SRMs. Moreover, by applying a state-of-the-art PH fitting, we have determined model parameters of PH-SRM even if the number of phases is rather large. This enables us to estimate quantitative software reliability in software testing phase with high accuracy. In the second example on system availability modeling, we have demonstrated how to apply PH approximation to an MRGP. Compared to the moment-based approach, KL-based approach has been able to provide the high accurate results on pointwise availability.

In future, we will have a plan to improve the computation effort of the current PH fitting algorithm. As shown in Section 5, EM algorithm for PH distribution takes much computation time in the case where the coefficient variation of probability distribution is large. This is caused by the fact that the estimated PH distribution involves a stiff CTMC in its phase structure. In such a case, the algorithm presented in Section 3 may not work well in some patterns. Hence, we will consider a more effective computation method even if the underlying CTMC is stiff. Moreover, it is known that PH approximation causes a state explosion leading to a large CTMC. We will try to develop numerical techniques to handle the large CTMC analysis.

References

- [1] J.A. Achcar, D.K. Dey and M. Niverthi: A Bayesian approach using nonhomogeneous Poisson processes for software reliability models. In A.P. Basu, K.S. Basu and S. Mukhopadhyay (eds.): *Frontiers in Reliability* (World Scientific, Singapore, 1998), 1–18.
- [2] H. Akaike: Information theory and an extension of the maximum likelihood principle. In B.N. Petrov and F. Csaki (eds.): *Proceedings of 2nd International Symposium on Information Theory* (1973), 267–281.
- [3] D. Aldous and L. Shepp: The least variable phase-type distribution is Erlang. *Stochastic Models*, **3** (1987), 467–473.
- [4] T. Altiok: On the phase-type approximations of general distributions. *IIE Transactions*, **17** (1985), 110–116.
- [5] S. Asmussen and G. Koole: Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, **30** (1993), 365–372.
- [6] S. Asmussen, O. Nerman and M. Olsson: Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, **23** (1996), 419–441.
- [7] R.E. Barlow and R. Campo: Total time on test processes and applications to failure data analysis. In R.E. Barlow, J. Fussell and N.D. Singpurwalla (eds.): *Reliability and Fault Tree Analysis* (SIAM, Philadelphia, 1975), 451–481.

- [8] M. Bladt, A. Gonzalez and S.L. Lauritzen: The estimation of phase-type related functionals using Markov chain Monte Carlo methods. *Scandinavian Actuarial Journal*, **4** (2003), 280–300.
- [9] A. Bobbio and A. Cumani: ML estimation of the parameters of a PH distribution in triangular canonical form. In G. Balbo and G. Serazzi (eds.): *Computer Performance Evaluation* (Elsevier Science Publishers, 1992), 33–46.
- [10] A. Bobbio, A. Horváth and M. Telek: Matching Three Moments with Minimal Acyclic Phase Type Distributions. *Stochastic Models*, **21** (2005), 303–326.
- [11] A. Bobbio and M. Telek: A benchmark for PH estimation algorithms: Result for acyclic-PH. *Stochastic Models*, **10** (1994), 661–677.
- [12] G. Bolch, S. Greiner, H. de Meer and K.S. Trivedi: *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, 2nd edition (John Wiley & Sons, New York, 2006).
- [13] BuTools: <http://webspn.hit.bme.hu/~telek/tools/butools/butools.html> (Accessed Sep 2, 2015).
- [14] H. Choi, V.G. Kulkarni and K.S. Trivedi: Markov regenerative stochastic Petri nets. *Performance Evaluation*, **20** (1994), 337–357.
- [15] E. Cinlar: *Introduction to Stochastic Processes* (Prentice-Hall, Englewood Cliffs, 1975).
- [16] A. Cumani: On the canonical representation of homogeneous Markov processes modelling failure-time distributions. *Microelectronics and Reliability*, **22** (1982), 583–602.
- [17] A.P. Dempster, N.M. Laird and D.B. Rubin: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **B-39** (1977), 1–38.
- [18] EMpht: <http://home.math.au.dk/asmus/pspapers.html> (Accessed Sep 2, 2015).
- [19] M.J. Faddy: On inferring the number of phases in a Coxian phase-type distribution. *Stochastic Models*, **14** (1998), 407–417.
- [20] A. Feldmann and W. Whitt: Fitting mixtures of exponentials to long-tail distributes to analyze network performance models. *Performance Evaluation*, **31** (1998), 245–258.
- [21] R. Fricks, M. Telek, A. Puliafito and K.S. Trivedi: Markov renewal theory applied to performability evaluation. In K. Bagchi and G. Zobrist (eds.): *State-of-the Art in Performance Modeling and Simulation. Modeling and Simulation of Advanced Computer Systems: Applications and Systems* (Gordon and Breach Publishers, Newark, NJ, 1998), 193–236.
- [22] T. Fujiwara and S. Yamada: Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit. In *Proceedings of 7th ISSAT International Conference on Reliability and Quality in Design* (2001), 36–40.
- [23] G-FIT: <http://ls4-www.cs.tu-dortmund.de/home/thummler/mainE.html> (Accessed Sep 2, 2015).
- [24] R. German, D. Logothetis and K.S. Trivedi: Transient analysis of Markov regenerative stochastic Petri nets: a comparison of approaches. In *Proceedings of 6th International Conference on Petri Nets and Performance Models* (1995), 103–112.
- [25] A.L. Goel: Software Reliability Models: Assumptions, Limitations and Applicability. *IEEE Transactions on Software Engineering*, **SE-11** (1985), 1411–1423.
- [26] A.L. Goel and K. Okumoto: Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Transactions on Reliability*, **R-28** (1979), 206–211.

- [27] S.S. Gokhale and K.S. Trivedi: Log-logistic software reliability growth model. In *Proceedings of 3rd IEEE International High-Assurance Systems Engineering Symposium* (1998), 34–41.
- [28] Q.-H. He and H. Zhang: An algorithm for computing minimal Coxian representations. *INFORMS Journal on Computing*, **20** (2008), 179–190.
- [29] A. Horváth and M. Telek: PhFit: A general phase-type fitting tool. In *Proceedings of Computer Performance Evaluation/TOOLS*, Lecture Notes in Computer Science 2324 (Springer, Berlin, 2002), 82–91.
- [30] S. Hwang and H. Pham: Quasi-renewal time-delay fault removal consideration in software reliability modeling. *IEEE Transactions on Systems, Man, and Cybernetics–Part A*, **39** (2009), 200–209.
- [31] HyperStar:
<http://www.mi.fu-berlin.de/inf/groups/ag-tech/projects/HyperStar/index.html>
(Accessed Sep 2, 2015).
- [32] S. Inoue, S. Hayashida and S. Yamada: Toward practical software reliability assessment with change-point based on hazard rate models. In *37th Annual IEEE Computer Software and Applications Conference* (2013), 268–273.
- [33] M.A. Johnson: Selecting parameters of phase distributions: Combining nonlinear programming, heuristics, and Erlang distributions. *ORSA Journal on Computing*, **5** (1993), 69–83.
- [34] M.A. Johnson and M.R. Taaffe: Matching moments to phase distributions: Mixtures of Erlang distribution of common order. *Stochastic Models*, **5** (1989), 711–743.
- [35] M.A. Johnson and M.R. Taaffe: Matching moments to phase distributions: Density function shapes. *Stochastic Models*, **6** (1990), 283–306.
- [36] M.A. Johnson and M.R. Taaffe: Matching moments to phase distributions: Nonlinear programming approaches. *Stochastic Models*, **6** (1990), 259–281.
- [37] M.A. Johnson and M.R. Taaffe: An investigation of phase-distribution moment-matching algorithms for use in queueing models. *Queueing Systems*, **8** (1991), 129–148.
- [38] jPhaseFit: <http://copa.uniandes.edu.co/?p=141> (Accessed Sep 2, 2015).
- [39] T.M. Khoshgoftaar: Nonhomogeneous Poisson processes for software reliability growth. In *Proceedings of the International Conference on Computational Statistics* (1988), 13–14.
- [40] D.K. Kim, D.H. Park and I.K. Yeo: Mean functions based on meta-mixtures in non-homogeneous Poisson processes. *Journal of the Korean Statistical Society*, **39** (2010), 237–244.
- [41] V.G. Kulkarni: *Modeling and Analysis of Stochastic Systems* (Chapman and Hall, New York, 1995).
- [42] N. Langberg and N.D. Singpurwalla: Unification of some software reliability models. *SIAM journal on scientific and statistical computing*, **6** (1985), 781–790.
- [43] J.C. Laprie, K. Kanoun, C. Béounes and M. Kaâniche: The KAT (Knowledge-Action-Transformation) approach to the modeling and evaluation of reliability and availability growth. *IEEE Transactions on Software Engineering*, **17** (1991), 370–382.
- [44] B. Littlewood: Stochastic reliability growth: a model for fault removal in computer programs and hardware designs. *IEEE Transactions on Reliability*, **R-30** (1981), 313–320.

- [45] B. Littlewood: Rationale for a modified Duane model. *IEEE Transactions on Reliability*, **R-33** (1984), 157–159.
- [46] M.R. Lyu (ed.): *Handbook of Software Reliability Engineering* (McGraw-Hill, New York, 1996).
- [47] mapfit: <http://cran.r-project.org/web/packages/mapfit/index.html> (Accessed Sep 2, 2015).
- [48] R. Marie: Calculating equilibrium probabilities for $\lambda(n)/C_k/1/N$ queues. In *Proceedings of the 1980 International Symposium on Computer Performance Modelling, Measurement and Evaluation* (ACM Press, New York, 1980), 117–125.
- [49] G.J. McLachlan and P.N. Jones: Fitting mixture models to grouped and truncated data via the EM algorithm. *Biometrics*, **44** (1988), 571–578.
- [50] momentmatching:
<http://www.cs.cmu.edu/~osogami/code/momentmatching/index.html> (Accessed Sep 2, 2015).
- [51] J.D. Musa: *Software Reliability Engineering* (McGraw-Hill, New York, 1999).
- [52] J.D. Musa, A. Iannino and K. Okumoto: *Software Reliability, Measurement, Prediction, Application* (McGraw-Hill, New York, 1987).
- [53] J.D. Musa and K. Okumoto: A Logarithmic Poisson Execution Time Model for Software Reliability Measurement. In *Proceedings of 7th International Conference on Software Engineering* (1984), 230–238.
- [54] C.A. O’Cinneide: On nonuniqueness of representations of phase-type distributions. *Stochastic Models*, **5** (1989), 322–330.
- [55] C.A. O’Cinneide: Characterization of phase-type distributions. *Stochastic Models*, **6** (1990), 1–57.
- [56] C.A. O’Cinneide: Phase-type distributions and invariant polytopes. *Advances in Applied Probability*, **23** (1991), 515–535.
- [57] C.A. O’Cinneide: Triangular order of triangular phase-type distributions. *Stochastic Models*, **9** (1993), 507–529.
- [58] M. Ohba: Inflection S-shaped software reliability growth model. In S. Osaki and Y. Hatoyama (eds.): *Stochastic Models in Reliability Theory* (Springer-Verlag, Berlin, 1984), 144–165.
- [59] H. Okamura and T. Dohi: Building phase-type software reliability models. In *Proceedings of 17th International Symposium on Software Reliability Engineering* (2006), 289–298.
- [60] H. Okamura and T. Dohi: Hyper-Erlang software reliability model. In *Proceedings of 14th Pacific Rim International Symposium on Dependable Computing* (2008), 232–239.
- [61] H. Okamura and T. Dohi: Faster maximum likelihood estimation algorithms for Markovian arrival processes. In *Proceedings of 6th International Conference on Quantitative Evaluation of Systems* (2009), 73–82.
- [62] H. Okamura and T. Dohi: mapfit: An R-based tool for PH/MAP parameter estimation. In *Proceedings of 12th International Conference on Quantitative Evaluation of Systems*, Lecture Notes in Computer Science 9259 (Springer-Verlag, Berlin, 2015), 105–112.
- [63] H. Okamura and T. Dohi: Phase-Type Software Reliability Model: Parameter Estimation Algorithms with Grouped Data. *Annals of Operations Research*, in press.

- [64] H. Okamura, T. Dohi and S. Osaki: Software reliability growth models with normal failure time distributions. *Reliability Engineering and System Safety*, **116** (2013), 135–141.
- [65] H. Okamura, T. Dohi and K.S. Trivedi: Markovian arrival process parameter estimation with group data. *IEEE/ACM Transactions on Networking*, **17** (2009), 1326–1339.
- [66] H. Okamura, T. Dohi and K.S. Trivedi: A refined EM algorithm for PH distributions. *Performance Evaluation*, **68** (2011), 938–954.
- [67] H. Okamura, T. Dohi and K.S. Trivedi: Improvement of EM algorithm for phase-type distributions with grouped and truncated data. *Applied Stochastic Models in Business and Industry*, **29** (2013), 141–156.
- [68] H. Okamura, R. Watanabe and T. Dohi: Variational Bayes for phase-type distribution. *Communications in Statistics – Simulation and Computation*, **43** (2014), 2013–2044.
- [69] H. Okamura, Y. Watanabe and T. Dohi: An iterative scheme for maximum likelihood estimation in software reliability modeling. In *Proceedings of 14th International Symposium on Software Reliability Engineering* (2003), 246–256.
- [70] M. Olsson: Estimation of phase-type distributions from censored data. *Scandinavian Journal of Statistics*, **23** (1996), 443–460.
- [71] T. Osogami and M. Harchol-Balter: Closed Form Solutions for Mapping General Distributions to Minimal PH Distributions. *Performance Evaluation*, **63** (2006), 524–552.
- [72] A. Panchenko and A. Thümmler: Efficient phase-type fitting with aggregated traffic traces. *Performance Evaluation*, **64** (2007), 629–645.
- [73] R. Peng, Y.F. Li, W.J. Zhang and Q.P. Hu: Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. *Reliability Engineering and System Safety*, **126** (2014), 37–43.
- [74] H. Pham: *Software Reliability* (Springer, Singapore, 2000).
- [75] H. Pham and L. Nordmann: A generalized NHPP software reliability model. In *Proceedings of 3rd ISSAT international conference on reliability and quality in design* (1997), 116–120.
- [76] PhFit: <http://webspn.hit.bme.hu/~telek/tools.htm> (Accessed Sep 2, 2015).
- [77] P. Reinecke, T. Kraub and K. Wolter: Cluster-based fitting of phase-type distributions to empirical data. *Computers and Mathematics with Applications*, **64** (2012), 3840–3851.
- [78] G. Schwarz: Estimating the Dimension of a Model. *Annals of Statistics*, **6** (1978), 461–464.
- [79] H. Takahasi and M. Mori: Double exponential formulas for numerical integration. *Publications of the Research Institute for Mathematical Sciences*, **9** (1974), 721–741.
- [80] M. Telek and A. Heindl: Matching moments for acyclic discrete and continuous phase-type distributions of second order. *International Journal of Simulation Systems, Science & Technology*, **3** (2002), 47–57.
- [81] M. Telek and G. Horváth: A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*, **64** (2007), 1153–1168.
- [82] A. Thümmler, P. Buchholz and M. Telek: A novel approach for phase-type fitting with the EM algorithm. *IEEE Transactions on Dependable Secure Computing*, **3** (2006), 245–258.

- [83] Y. Tohma, Y. Yamano, H. Ohba and R. Jacoby: The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model. *IEEE Transactions on Software Engineering*, **17** (1991), 483–489.
- [84] A. van de Liefvoort: The moment problem for continuous distributions, Technical report (University of Missouri, WP-CM-1990-02, Kansas City, 1990).
- [85] M.C. van der Heijden: On the three-moment approximation of a general distribution by a Coxian distribution. *Probability in the Engineering and Informational Sciences*, **2** (1988), 257–261.
- [86] R. Watanabe, H. Okamura and T. Dohi: An efficient MCMC algorithm for countinuous PH distributions. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose and A. M. Uhrmacher (eds.): *Proceedings of The 2012 Winter Simulation Conference* (2012), 12 pages.
- [87] C.F.J. Wu: On the convergence properties of the EM algorithm. *Annals of Statistics*, **11** (1983), 95–103.
- [88] M. Xie, Q.P. Hu, Y.P. Wu and S.H. Ng: A study of the modeling and analysis of software fault-detection and fault-correction processes. *Quality and Reliability Engineering International*, **23** (2007), 459–470.
- [89] S. Yamada: A stochastic software reliability growth model with Gompertz curve. *Transactions of Information Processing Society of Japan* (in Japanese), **37** (1992), 964–969.
- [90] S. Yamada, M. Ohba and S. Osaki: S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability*, **R-32** (1983), 475–478.
- [91] S. Yamada, K. Tokuno and S. Osaki: Imperfect debugging models with fault introduction rate for software reliability assessment. *International Journal of Systems Science*, **23** (1992), 2241–2252.
- [92] Y. Yamaguchi, H. Okamura and T. Dohi: Variational Bayesian approach for estimating parameters of a mixture of Erlang distributions. *Communications in Statistics – Theory and Methods*, **39** (2010), 2333–2350.
- [93] X. Zhang and H. Pham: A software cost model with error removal times and risk costs. *International Journal of Systems Science*, **29** (1998), 435–442.
- [94] M. Zhao and M. Xie: On maximum likelihood estimation for a general non-homogeneous Poisson process. *Scandinavian Journal of Statistics*, **23** (1996), 597–607.

Hiroyuki Okamura
Department of Information Engineering
Hiroshima University
1-4-1 Kagamiyama, Higashi-Hiroshima
Hiroshima 739-8527, Japan
E-mail: okamu@rel.hiroshima-u.ac.jp