数理解析研究所講究録 1252 巻 2002 年 96-102

Phase–Type Software Reliability Models

岡村寛之，渡部保博，土肥正
Hiroyuki Okamura, Yasuhiro Watanabe and Tadashi Dohi
Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan

1 Introduction

Software reliability models (SRMs) are classified into *time domain models* and *counting process models*. The time domain model is the stochastic model based on the sequence of inter-failure times. Jelinski and Moranda model [4] and Schick and Wolverton model [13] are the most classical models belonging to this class. On the other hand, the counting process models have gained popularity for describing the stochastic behavior of the number of software failures observed in the testing phase. The most well-known and tractable models are non-homogeneous Poisson process (NHPP) models. Goel and Okumoto [3], Yamada, Ohba and Osaki [15], Musa and Okumoto [9] develop representative NHPP models. These SRMs are based on the different debugging scenarios, and can catch qualitatively typical (but not general) reliability growth phenomena observed in the testing phase of software products. It should be noted, however, that the SRMs based on past observations may not be always useful in the software testing process. Because one cannot catch the global trend of the software failure occurrence phenomena in the initial testing phase. In other words, a unified modeling framework comprising some typical reliability growth patterns should be developed for robust software reliability assessment.

Langberg and Singpurwalla [7] show that several SRMs can be comprehensively viewed by adopting a Bayesian point of view. Miller [8] extends the Langberg and Singpurwalla's idea and considers an exponential order statistics model. Raftery [12], Kuo and Yang [5] investigate the modeling framework based on the generalized order statistics (GOS), and discuss several parameter estimation methods from the standpoint of both Bayesian and non-Bayesian statistics. In the GOS modeling framework, the SRMs can be characterized by only the fault detection time distribution.

This article proposes phase–type SRMs based on the GOS of software failure data. To unify some existing SRMs, the phase–type distribution [10], which represents the software fault detection time distribution, is used to represent the GOS of software failure data. Also, we provide a unified estimation method for model parameters in the phase–type SRMs. The usual estimation method, such as the maximum likelihood estimation (MLE) based on the Newton's method, does not function well in many cases, since the phase–type SRMs often have many model parameters. To overcome this problem, we develop the EM (expectation-maximization) algorithms [1, 2, 11] to compute the maximum likelihood estimates of model parameters.

2 BASIC SRMS

Suppose that the number of software faults detected before time t, $N(t)$, obeys the following Poisson distribution:

$$\Pr\{N(t)=k\}=\frac{\{\Lambda(t)\}^k}{k!}\exp\{-\Lambda(t)\}, \tag{1}$$

where $\Lambda(t)=\mathrm{E}[N(t)]$ is the mean value function and $\lambda(t)=d\Lambda(t)/dt$ is the intensity function of the NHPP. If the mean value function is bounded, *i.e.*

$$\lim_{t\to\infty}\Lambda(t)<\infty, \tag{2}$$

the corresponding counting process models are called NHPP-I or finite failure models. For instance, Goel and Okumoto [3] assume that the expected number of faults detected per unit time is proportional to the expected number of remaining faults, that is, $\lambda(t)=b\{a-\Lambda(t)\}$, where a (>0) and b (>0) are the expected initial number of faults and the fault detection rate per unit time, respectively. By solving this differential equation with initial condition $\Lambda(0)=0$, we have

$$\Lambda(t)=a(1-\exp\{-bt\}). \tag{3}$$

On the other hand, if the mean value function is not bounded, *i.e.*

$$\lim_{t\to\infty} \Lambda(t) \to \infty, \tag{4}$$

the corresponding counting process models are called NHPP-II or infinite failure models. The logarithmic Poisson execution time model by Musa and Okumoto model [9]:

$$\Lambda(t) = \frac{1}{\theta}\log(\lambda_0\theta t + 1) \tag{5}$$

belongs to this class, where θ (> 0) is the reduction rate of instantaneous software intensity and λ_0 (> 0) is the initial intensity.

In this way, the SRMs based on the NHPP can be characterized by the mean value function $\Lambda(t)$. In other words, the existing SRMs are modeled by taking account of only the deterministic behavior of the mean value function. However, as shown later, the deterministic modeling does not validate the stochastic phenomenon behind the software debugging. In the following section, we introduce a modeling framework based on the GOS and further develop more general and robust SRMs.

3 Unification of SRMs

3.1 GOS Models

Langberg and Singpurwalla [7] propose a modeling framework to unify some SRMs. They make the following assumptions:

Assumption A: Software failures caused by software faults occur at independently and identically distributed (i.i.d.) random times.

Assumption B: The initial number of software faults is finite.

Let $\{N(t);\ t \geq 0\}$ and $\{X_i;\ i = 1, 2, \ldots\}$ denote the number of faults detected before time t (> 0) and the sequence of fault detection times, respectively. From Assumption A, X_i, $i = 1, 2, \ldots$, are i.i.d. random variables with (absolutely continuous) probability distribution function $F(t)$ and probability density function $f(t) = dF(t)/dt$. If the initial number of software faults is known as a constant N (> 0), the probability mass function of the number of faults detected before time t is given by

$$\Pr\{N(t) = n\} = \binom{N}{n} F(t)^n \overline{F}(t)^{N-n}, \tag{6}$$

where $\overline{F}(\cdot) = 1 - F(\cdot)$. For the GOS models, we suppose that the initial number of faults obeys the Poisson distribution with parameter ω (> 0). Then the number of faults detected before time t follows

$$\Pr\{N(t) = k\} = \frac{\{\omega F(t)\}^k}{k!} \exp\{-\omega F(t)\}. \tag{7}$$

Since Eq. (7) is equivalent to the probability mass function of NHPP having the mean value function, $\omega F(t)$, we can represent some existing SRMs by substituting a typical probability distribution into $F(t)$ in Eq. (7). Ultimately, it can be shown that the GOS models explain stochastically the physical debugging processes from the finite population and provide most existing SRMs.

3.2 Phase–Type SRMs

The problem in the GOS models is how to select a suitable fault detection time distribution $F(t)$. For instance, if the fault detection time is an exponentially distributed random variable, the resulting SRM is reduced to Jelinski and Moranda model [4] or Goel and Okumoto model [3]. If the fault detection time obeys the 2-Erlang distribution, the resulting model is the S-shaped SRM [15]. For the sake of simplification, we focus only on the counting process model, namely, the NHPP model in this paper. Of course, the time domain model, as shown before, are special model with constant N in the GOS models, so that we can discuss the time domain model in the similar way to the NHPP model. Before describing the unification of NHPP models, we consider two representative NHPP models: the S-shaped SRM and the hyperexponential SRM. The main difference between them is that the S-shaped model consists of the convolution of two exponential distributions, but the hyperexponential SRM does of the superposition of multiple exponential distributions. Although their structures are quite different from each other, it can be seen that both of them are specific Markov processes. In other words, their structures present the respective debugging scenarios. Thus it is straightforward to see that the SRMs are unified under the assumption that the fault detection process is governed by a general Markov process.

Consider a Markov process with state space $\{1, 2, \cdots, m+1\}$, where each state is called a *phase*. The phase indicates a specific activity in the software testing, for example, inspection, review, debugging, *etc.* Without any loss of generality, the phases $\{1, 2, \cdots, m\}$ correspond to the transient states on the Markov process and the phase $\{m+1\}$ does to a detection of a software fault, *i.e.*, it is an absorbing state. The initial probability vector for the Markov process is given by $(\boldsymbol{\alpha}, 0)$, where $\boldsymbol{\alpha}$ is the $1 \times m$ probability vector. Until detection of a fault (absorption), the phase changes according to the Markov process with infinitesimal generator $\boldsymbol{Q}^*$, where

$$\boldsymbol{Q}^* = \left(\begin{array}{cccc|c} -t_{11} & t_{12} & \cdots & t_{1m} & \xi_1 \\ t_{21} & -t_{22} & \cdots & t_{2m} & \xi_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{m1} & t_{m2} & \cdots & -t_{mm} & \xi_m \\ \hline 0 & 0 & \cdots & 0 & 0 \end{array} \right). \tag{8}$$

In Eq. (8), t_{ij} is the transition rate from the phase i to j, ξ_i is the transition rate into the absorbing state representing the detection of a software fault. Furthermore, the diagonal elements, t_{ii}, are given by

$$t_{ii} = -\sum_{j \neq i} t_{ij} - \xi_i, \qquad \text{for all } i. \tag{9}$$

From the elementary argument on Markov processes, the fault detection time distribution characterizing the SRMs is given by

$$\begin{aligned} F(t) &= 1 - \boldsymbol{\alpha} \exp(\boldsymbol{Q}t)\boldsymbol{e}^t, & (10)\\ \boldsymbol{\alpha} &= (\alpha_1, \cdots, \alpha_m)', & (11)\\ \boldsymbol{Q} &= \begin{pmatrix} -t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & -t_{22} & \cdots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & -t_{mm} \end{pmatrix}, & (12) \end{aligned}$$

where $\boldsymbol{e}^t$ is a column vector of 1s. The probability distribution given by Eq (10) is called the ***phase-type distribution*** with parameter $(\boldsymbol{\alpha}, \boldsymbol{Q})$ [10]. The phase–type distribution represents every probability distribution belonging to the exponential family, such as exponential distribution, k-Erlang distribution, hyperexponential distribution, Coxian–type distribution and so on. More generally, the phase–type distribution characterizes every series/parallel system with exponentially distributed failure time and approximates arbitrary probability distribution with arbitrary accuracy if the number of phases is large. Substituting the phase–type distribution into $F(t)$ in the GOS models yields ***phase-type SRMs***. These are obviously the widest class of SRMs which can be treated mathematically.

4 Parameter Estimation

Let $\Lambda(t; \boldsymbol{\theta})$ be the mean value function of an NHPP, where $\boldsymbol{\theta}$ is a set of parameters. If the fault detection time data, $\boldsymbol{x} = (x_1, \ldots, x_n)$, is available, the model parameters are determined so as to maximize the logarithmic likelihood function of the NHPP:

$$\log L_{\mathrm{NHPP}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \lambda(x_i; \boldsymbol{\theta}) - \Lambda(x_n; \boldsymbol{\theta}), \tag{13}$$

where $\lambda(\cdot)$ is the intensity function of the NHPP:

$$\lambda(t; \boldsymbol{\theta}) = \frac{d}{dt}\Lambda(t; \boldsymbol{\theta}). \tag{14}$$

Since Eq. (13) is a nonlinear function of $\boldsymbol{\theta}$ in many cases, we have to solve numerically the following simultaneous equations:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log L_{\mathrm{NHPP}}(\boldsymbol{\theta}) = \mathbf{0}. \tag{15}$$

This is the first-order condition of optimality for the MLE. The Newton's method may be used to obtain the solution of the above logarithmic likelihood equations. However, it does not function well in terms of computation cost and convergence property. The main reason is that the logarithmic likelihood function often becomes a multi-modal function. This implies that the estimates based on the Newton's method may not converge to the desired values due to the dependence of the initial value. For this problem, some authors (*e.g.* [6]) propose the improved methods, but none of them realizes the effective computation for the maximum likelihood estimates with suitable effort. In this section, we develop iteration algorithms to estimate the model parameters based on the EM (expectation-maximization) principle [2, 14]. Although the EM algorithm tends to converge more slowly than the Newton's method does, the associated likelihood becomes surely higher as the estimates are updated in the EM algorithm. This property is fairly effective even if the logarithmic likelihood function is a multi-modal function. Thus, the proposed algorithms provide robust methods and plausible solutions in many types of SRMs.

4.1 EM Algorithms for NHPP-I Models

Let $X_1, X_2, \ldots, X_N$ and $X_{[1]} < X_{[2]} < \cdots < X_{[N]}$ be fault detection times and their order statistics, respectively, where N is the initial number of faults and the Poisson distributed random variable with parameter ω (> 0). If one can observe all the fault detection times $\boldsymbol{x} = (x_1, \ldots, x_n)$, which is the complete data, the likelihood function is given by

$$L(\omega, \boldsymbol{\theta}) = \Pr\{N = n\} n! \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}). \tag{16}$$

From Assumption B, we have

$$L(\omega, \boldsymbol{\theta}) = \omega^n \exp\{-\omega\} \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}) \tag{17}$$

and hence

$$\log L(\omega, \boldsymbol{\theta}) = n \log \omega - \omega + \sum_{i=1}^{n} \log f(x_i; \boldsymbol{\theta}). \tag{18}$$

In this situation, the standard argument on the MLE gives the following estimators:

$$\hat{\omega} = n \tag{19}$$

and

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{n} \log f(x_i; \boldsymbol{\theta}) \right\}. \tag{20}$$

The EM algorithm is an iterative method for an estimation problem with incomplete data. Let X and $Y = u(X)$ be the unobserved random variable with probability density $f(\cdot; \boldsymbol{\theta})$ and the observed random variable, respectively. Given the observed experiment, $\boldsymbol{y}$, we estimate the parameter set $\boldsymbol{\theta}$. The $(n+1)$-st step in the EM algorithm consists of finding $\boldsymbol{\theta}_{n+1}$ which maximizes the expected logarithmic likelihood function for the complete data, provided that the incomplete data is observed. That is,

$$\hat{\boldsymbol{\theta}}_{n+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \mathrm{E}[\log f(X; \boldsymbol{\theta}) | u(X) = \boldsymbol{y}; \hat{\boldsymbol{\theta}}_n] \right\}, \tag{21}$$

where $\hat{\boldsymbol{\theta}}_n$ is the estimated parameter set at the n-th step in the EM algorithm and $\mathrm{E}[\cdot\,; \boldsymbol{\theta}]$ denotes an expected value provided that the probability density f has the parameter set $\boldsymbol{\theta}$.

Let us now return our argument to the estimation problem for the NHPPs. Given the fault detection time data at time t (≥ 0), $\boldsymbol{y} = (x_1, \ldots, x_k)$, $k < N$, it is seen that $\boldsymbol{y}$ is the incomplete data instead of the complete data $\boldsymbol{x} = (x_1, \ldots, x_n)$. Then the EM algorithm is developed as follows. At the $(n+1)$-st step, the estimates of the model parameters are calculated as

$$\hat{\omega}_{n+1} = \mathrm{E}[N | X = \boldsymbol{y}; \hat{\omega}_n, \hat{\boldsymbol{\theta}}_n] \tag{22}$$

and

$$\hat{\boldsymbol{\theta}}_{n+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \mathrm{E}\left[\sum_{i=1}^{N} \log f(X_i;\boldsymbol{\theta}) \,\middle|\, X = \boldsymbol{y}; \hat{\omega}_n, \hat{\boldsymbol{\theta}}_n \right] \right\}. \tag{23}$$

Notice that $X = \boldsymbol{y}$ denotes an event; $\{X_{[1]} = x_1, X_{[2]} = x_2, \ldots, X_{[k]} = x_k\}$.

4.2 EM Algorithms for Phase–Type SRMs

Next, we consider the EM algorithms in the phase–type SRMs. The phase–type SRMs are strictly consistent with the GOS models if the fault detection process is modeled by a Markov process. In general, the estimation procedure for phase–type distribution is more complex than those for exponential distribution, 2-Erlang distribution and so on, since the phase–type distribution usually has many parameters. The EM algorithms for the phase–type distribution are considered by some authors. Asmussen *et al.* [1] obtain the EM algorithm in the case of no censored observations. Olsson [11] also derives the EM algorithms in both cases of right-censored and interval-censored observations. Applying these results, we can develop the EM algorithms for the phase–type SRMs.

Suppose that the probability distribution of software fault detection time $F(t)$ is the phase–type distribution with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{Q}$ in Eqs. (10)–(12). Let $X_1, X_2, \ldots, X_N$ and $X_{[1]} < X_{[2]} < \cdots < X_{[N]}$ denote the software fault detection times and their order statistics, respectively, where N is the initial number of software faults (Poisson distributed random variable with parameter ω (> 0)). The i-th software fault detection time, X_i, consists of two random variable sequences: an embedded Markov chain on the phases

$$I_0^{[i]}, I_1^{[i]}, \ldots, I_{M-1}^{[i]}; \quad (I_M^{[i]} = 0)$$

and its sojourn time

$$S_0^{[i]}, S_1^{[i]}, \ldots, S_{M-1}^{[i]}; \quad (S_M^{[i]} = \infty),$$

where M is the number of transitions until detection of a software fault. The fault detection time X_i can be represented as the sum of sojourn times, *i.e.*,

$$X_i = \sum_{j=0}^{M-1} S_j^{[i]}. \tag{24}$$

It is assumed that one observes a complete data set for fault detection times $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, where

$$\boldsymbol{x}_\nu = (i_0^{[\nu]}, \ldots, i_{m[\nu]-1}^{[\nu]}, s_0^{[\nu]}, \ldots, s_{m[\nu]-1}^{[\nu]}), \quad \text{for } \nu = 1, \ldots, n.$$

Define

$$B_i = \sum_{\nu=1}^{n} \mathbf{1}_{\{I_0^{[\nu]}=i\}}, \quad \text{for } i = 1, \ldots, m, \tag{25}$$

$$Z_i = \sum_{\nu=1}^{n} \prod_{p=0}^{M-1} \mathbf{1}_{\{I_0^{[\nu]}=i\}} S_p^{[i]}, \quad \text{for } i = 1, \ldots, m \tag{26}$$

and

$$N_{ij} = \sum_{\nu=1}^{n} \sum_{p=0}^{M-1} \mathbf{1}_{\{I_p^{[\nu]}=i, I_{p+1}^{[\nu]}=j\}}, \quad \text{for } i \neq j,\ i = 1, \ldots, m,\ j = 0, \ldots, m. \tag{27}$$

In Eqs. (25)–(27), B_i, Z_i and N_{ij} are the number of processes starting from the phase i, the total sojourn time in the phase i and the total number of transitions from the phase i to the phase j ($j = 0$ represents an absorption of process), respectively, and $\mathbf{1}_{\{A\}}$ denotes the indicator function for an event A:

$$\mathbf{1}_{\{A\}} = \begin{cases} 1, & \text{event } A \text{ occurs}, \\ 0, & \text{otherwise}. \end{cases} \tag{28}$$

The likelihood function of the complete data set is then given by

$$L(\boldsymbol{\alpha}, \boldsymbol{Q}) = \omega^n \exp\{-\omega\} \prod_{i=1}^{m} \alpha_i^{B_i} \prod_{i=1}^{m} \exp\{t_{ii} Z_i\} \prod_{i=1}^{m} \prod_{\substack{j=0 \\ j \neq i}}^{m} t_{ij}^{N_{ij}}. \tag{29}$$

From Eq. (29), the maximum likelihood estimates can be calculated as follows. For all $i, j = 1, \ldots, m$, we have

$$\hat{\omega} = n, \quad \hat{\alpha}_i = \frac{B_i}{n}, \quad \hat{t}_{ij} = \frac{N_{ij}}{Z_i}, \quad \hat{\xi}_i = \frac{N_{i0}}{Z_i} \tag{30}$$

and

$$\hat{t}_{ii} = -\left(\hat{t}_i + \sum_{j=1,\ j\neq i}^{m} \hat{t_{ij}} \right). \tag{31}$$

Let $\boldsymbol{y} = (x_1, \ldots, x_k)$, $k < N$ be the software fault detection time data observed before time t, which is incomplete data in both senses of the censored data and the macro data. The censored data means that the software fault detection time data is censored at time x_k, and the macro data means that trajectory of the phase in the fault detection process cannot be observed. From Eqs. (22) and (23), we can obtain an iteration scheme at the $(n+1)$-st step in the phase–type SRMs for $i, j = 1, \ldots, m$,

$$\hat{\omega}^{(n+1)} = \mathrm{E}[N|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}], \tag{32}$$

$$\hat{\alpha}_i^{(n+1)} = \frac{\mathrm{E}[B_i|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]}{\mathrm{E}[N|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]}, \tag{33}$$

$$\hat{t}_{ij}^{(n+1)} = \frac{\mathrm{E}[N_{ij}|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]}{\mathrm{E}[Z_i|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]}, \tag{34}$$

$$\hat{\xi}_i^{(n+1)} = \frac{\mathrm{E}[N_{i0}|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]}{\mathrm{E}[Z_i|X = \boldsymbol{y}; \hat{\omega}^{(n)}, \hat{\boldsymbol{\alpha}}^{(n)}, \hat{\boldsymbol{Q}}^{(n)}]} \tag{35}$$

and

$$\hat{t}_{ii}^{(n+1)} = -\left(\hat{\xi}_i^{(n+1)} + \sum_{j=1,\ j\neq i}^{m} \hat{t}_{ij}^{(n+1)} \right). \tag{36}$$

From the EM algorithm for NHPP-I models and the independence of X_i, it is straightforward to see that

$$\mathrm{E}[N|X = \boldsymbol{y}; \omega, \boldsymbol{\alpha}, \boldsymbol{Q}] = k + \omega \boldsymbol{\alpha} \boldsymbol{h}(x_k|\boldsymbol{Q}), \tag{37}$$

$$\begin{aligned} \mathrm{E}[B_i|X = \boldsymbol{y}; \omega, \boldsymbol{\alpha}, \boldsymbol{Q}] &= \sum_{\nu=1}^{k} \mathrm{E}[B_i^{[\nu]}|X = x_\nu; \boldsymbol{\alpha}, \boldsymbol{Q}] \\ &\quad + \omega \boldsymbol{\alpha} \boldsymbol{h}(x_k|\boldsymbol{Q}) \mathrm{E}[B_i|X > x_k; \boldsymbol{\alpha}, \boldsymbol{Q}], \end{aligned} \tag{38}$$

$$\begin{aligned} \mathrm{E}[Z_i|X = \boldsymbol{y}; \omega, \boldsymbol{\alpha}, \boldsymbol{Q}] &= \sum_{\nu=1}^{k} \mathrm{E}[Z_{ij}^{[\nu]}|X = x_\nu; \boldsymbol{\alpha}, \boldsymbol{Q}] \\ &\quad + \omega \boldsymbol{\alpha} \boldsymbol{h}(x_k|\boldsymbol{Q}) \mathrm{E}[Z_{ij}|X > x_k; \boldsymbol{\alpha}, \boldsymbol{Q}] \end{aligned} \tag{39}$$

and

$$\begin{aligned} \mathrm{E}[N_{ij}|X = \boldsymbol{y}; \omega, \boldsymbol{\alpha}, \boldsymbol{Q}] &= \sum_{\nu=1}^{k} \mathrm{E}[N_{ij}^{[\nu]}|X = x_\nu; \boldsymbol{\alpha}, \boldsymbol{Q}] \\ &\quad + \omega \boldsymbol{\alpha} \boldsymbol{h}(x_k|\boldsymbol{Q}) \mathrm{E}[N_{ij}|X > x_k; \boldsymbol{\alpha}, \boldsymbol{Q}], \end{aligned} \tag{40}$$

where

$$\boldsymbol{h}(x|\boldsymbol{Q}) = \exp\{\boldsymbol{Q}x\}\boldsymbol{e}^t. \tag{41}$$

The expected values in Eqs. (38)–(40) appear in the EM algorithms for the phase–type distribution using the right-censored data. Olsson [11] introduces the computational method using the homogeneous linear differential equations. We will use this method for our problem in this paper.

The estimation method in the phase–type SRMs is now described by Eqs. (32)–(41). From these results, we will carry out the EM algorithms in the phase–type SRMs when the fault detection time data is given.

5 Conclusions

This paper has proposed the phase–type SRMs as the most general software reliability model. By modeling the fault detection process as a Markov process, we have shown that the phase–type SRMs were proved to be a comprehensive class of some existing SRMs. In addition, we have developed the EM algorithms for the maximum likelihood estimation in the phase–type SRMs. The algorithms proposed here are simple iteration algorithms but are effective in terms of computation effort.

References

[1] S. Asmussen, O. Nerman and M. Olsson, Fitting phase-type distributions via the EM algorithm, Scand. J. Statist., 23, 419–441, 1996

[2] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Statist. Soc., B-39, 1–38, 1977.

[3] A. Goel and K. Okumoto, Time-dependent error-detection rate model for software reliability and other performance measures, IEEE Trans. Reliab., R-28, 206–211, 1979.

[4] Z. Jelinski and P. B. Moranda, Software reliability research, Statistical Computer Performance Evaluation, W. Freiberger (ed.), 465–484, Academic Press, New York, 1972.

[5] L. Kuo and T. Y. Yang, Bayesian computation for nonhomogeneous Poisson processes in software reliability, J. Amer. Statist. Assoc, 91, 763–773, 1996.

[6] Y. Kutoyants, Multidimensional parameter estimation of the intensity of inhomogeneous Poisson processes, Control Inform. Theory, 11, 325–334, 1982.

[7] N. Langberg and N. D. Singpurwalla, Unification of some software reliability models, SIAM J. Sci. Comput., 6, 781–790, 1985.

[8] D. R. Miller, Exponential order statistic models of software reliability growth, IEEE Trans. Software Eng., SE-12, 12–24, 1986.

[9] J. D. Musa and K. Okumoto, A logarithmic Poisson execution time model for software reliability measurement, Proc. 7th Int'l Conf. Software Eng., 230–238, 1984.

[10] M. F. Neuts, Matrix-Geometric Solutions in Stochastic Models, Johns Hopkins University Press, Baltimore, MD, 1981.

[11] M. Olsson, Estimation of phase-type distributions from censord data, Scand. J. Statist., 23, 443–460, 1996.

[12] A. E. Raftery, Inference and prediction for a general order statistic model with unknown population size, J. Amer. Statist. Assoc., 82, 1163–1168, 1987.

[13] G. J. Schick and R. W. Wolverton, An analysis of competing software reliability models, IEEE Trans. Software Eng., SE-4, 104–120, 1978.

[14] C. F. J. Wu, On the convergence properties of the EM algorithm, Ann. Statist., 11, 95–103, 1983.

[15] S. Yamada, M. Ohba and S. Osaki, S-shaped reliability growth modeling for software error detection, IEEE Trans. Reliab., R-32, 475–478, 1983.