

Article

Application of EM Algorithm to NHPP-Based Software Reliability Assessment with Generalized Failure Count Data

Hiroyuki Okamura *  and Tadashi Dohi

Graduate School of Advanced Science and Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 7398527, Japan; dohi@hiroshima-u.ac.jp

* Correspondence: okamu@hiroshima-u.ac.jp

Abstract: Software reliability models (SRMs) are widely used for quantitative evaluation of software reliability by estimating model parameters from failure data observed in the testing phase. In particular, non-homogeneous Poisson process (NHPP)-based SRMs are the most popular because of their mathematical tractability. In this paper, we focus on the parameter estimation algorithm for NHPP-based SRMs and discuss the EM algorithm for generalized fault count data. The presented algorithm can be applied for failure time data, failure count data, and their mixture. The paper derives the EM-step formulas for basic 12 NHPP-based SRMs and demonstrate a numerical experiment to present the convergence property of our algorithms. The developed algorithms are suitable for an automatic tool for software reliability evaluation.

Keywords: software reliability model; maximum likelihood estimation; EM algorithm; non-homogeneous Poisson process; generalized failure count data



Citation: Okamura, H.; Dohi, T. Application of EM Algorithm to NHPP-Based Software Reliability Assessment with Generalized Failure Count Data. *Mathematics* **2021**, *9*, 985. <https://doi.org/10.3390/math9090985>

Academic Editor: Frank Werner

Received: 15 March 2021

Accepted: 26 April 2021

Published: 27 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Software reliability models (SRMs) are used to assess quantitative reliability and to control the quality of software products. Since Jelinski and Moranda [1], and Goel and Okumoto [2] presented SRMs based on stochastic processes, numerous SRMs have been proposed [3–8]. In particular, non-homogeneous Poisson process (NHPP)-based SRMs have become popular in representing the dynamics of failure occurrence processes in a variety of situations [9–13]. By using an NHPP-based SRM, we predict the future behavior of software failure, i.e., the number of failures experienced in future, and estimate the quantitative measure of software reliability.

The advantage of NHPP-based SRMs is simplifying the stochastic analysis. NHPPs are generally dominated by *mean value functions*. The mean value function indicates the expected number of failures experienced at arbitrary testing time. By choosing appropriate mean value functions, NHPP-based SRMs can fit any observed failure data. The NHPP-based SRMs and the mean value functions have a one-to-one correspondence.

The Goel–Okumoto model [2]; Goel model [2]; Musa–Okumoto model [14]; Ohba [15,16]; Yamada, Ohba, and Osaki model [17]; Zhao and Xie model [18] are early NHPP-based SRMs. They were constructed with the deterministic debugging scenarios of mean value functions. Pham [19] solved a generalized differential equation by which the mean value function in the NHPP-based SRM is governed and proposed a generalized SRM with many redundant parameters.

Apart from such a deterministic modeling framework, almost all NHPP-based SRMs can be characterized as Markov processes. Shantikumar [20] discussed a modeling framework to integrate time-homogeneous Markov processes and NHPPs by using a binomial-type stochastic point process. Langberg and Singpurwalla [21] presented a unified modeling framework for almost all NHPP-based SRMs. Chen and Singpurwalla [22] also discussed the framework with a self-exciting point process. Miller [23] introduced the

concept of exponential order statistics and drastically extended Langberg and Singpurwalla's idea. In fact, the realizations of NHPP-based SRM can be described by either the general order statistics or record value statistics of the underlying software fault data, where the fault-detection times are assumed to be independent and identically distributed (i.i.d.) random variables. Specifically, the general order statistics are based on the order of all the fault detection times, and the record value statistics focus on their maximum detection time.

In this paper, we focus on the parameter estimation problem of NHPP-based SRMs. In general, there are three steps to evaluating the software reliability with NHPP-based SRM. (i) Collect the failure data such as the number of detected bugs in the testing phase, (ii) estimate the model parameters of NHPP-based SRM to fit it to the collected data, and (iii) compute reliability measures from the NHPP-based SRM with the estimated parameters. Based on quantitative measures, we control the software development process. As a typical usage of NHPP-based SRM, we estimate the number of failures that will be experienced in the future and decide whether to continue testing the software or the software can be released. In other words, the parameter estimation of NHPP-based SRM is frequently executed in the software development phase. The computation cost of the estimation should be small in practice.

Therefore, many authors have concerns about the parameter estimation problem on NHPP-based SRMs. Nevertheless, in actual software reliability assessments, a few NHPP-based SRMs and familiar maximum likelihood (ML) estimation methods are still used conventionally. The main reason for this is that the practitioners wish to use intuitively simple statistical methods, which exclude empirically based tuning parameters for a few SRMs that have survived a long history of software reliability engineering. In fact, the Bayesian estimation methods are still minor in software engineering practice, although its theoretical benefit is recognized. On the other hand, ML estimation is based on the maximization of likelihood function with software failure data and possesses several rational properties such as asymptotic efficiency. Hossain and Dahiya [24] derived necessary and sufficient conditions that the maximum likelihood estimates (MLEs), which satisfy the non-linear likelihood equations, exist in Goel and Okumoto SRM [2]. Knafl and Morgan [25] presented a method to systematically solve the likelihood equations with two model parameters. Joe [26] also discussed the confidence interval of MLEs. Zhao and Xie [18] provided the MLEs for an extended Goel and Okumoto SRM. Jeske and Pham [27] discovered empirically that the MLEs in Goel and Okumoto SRM are not statistically consistent. It should be noted, however, that ML estimation is always possible for all NHPP-based SRMs. Even if the likelihood functions are strictly concave in model parameters, it is difficult to solve the likelihood equations analytically. For instance, in the cases where the likelihood functions are not concave and where there exists no solution for the likelihood equations inside the parameter space, the conventional methods to calculate the MLEs cannot be used. Usually, the Newton method and the Nelder–Mead method are used to solve the maximization problem in the existing literature. From the recent development of computational ability, it is becoming easier to handle a large-scale complex optimization problem.

On the other hand, it is known that the local convergence property of the Newton method is a weakness for practical application. The local convergence property means that the convergence radius of algorithm is limited, and thus, it may fail to obtain a result if we set unsuitable initial guesses. For example, when we develop the application to automatically obtain parameter estimates from given data, the local convergence property becomes troublesome when choosing the initial guesses. Therefore, the Newton method is not suitable for this purpose. Additionally, the Nelder–Mead method is one of the direct search methods. The convergence property of the Nelder–Mead method is improved from the Newton method. However, some design parameters should be provided appropriately for the Nelder–Mead method. Even if we use the Nelder–Mead algorithm, the convergence of algorithm is not always guaranteed for any given data.

From the early 2000s, our research group has developed an alternative parameter estimation algorithm based on the EM (expectation maximization) principle [28,29] and applied it to the software reliability assessment based on NHPP-based SRMs [30–40]. As another examples of EM algorithms in SRMs, Kimura and Yamada [41], Leadoux [42], and Okamura and Dohi [43] attempted to use EM algorithms to estimate the imperfect debugging model [44] and architecture-based SRMs [45,46]. Their models were based on the continuous-time Markov chain and are closely related to Markov-modulated Poisson processes and/or Markovian arrival processes. Additionally, Zeephongsekul et al. [47] and Nagaraju et al. [48] proposed ECM (expectation conditional maximization) algorithms for NHPP-based SRMs to handle several specific models.

The EM algorithm is an algorithm that finds maximum likelihood estimates for a statistical model with incomplete data. The idea behind our EM algorithms is to find the incomplete data structure of NHPP-based SRMs. Concretely, in NHPP-based SRMs, we assume that the number of failures is finite due to a finite number of software bugs, but all of them cannot be observed, i.e., the number of remaining software failures can be regarded as missing data. From this insight, the EM algorithm for an individual NHPP-based SRM is developed. Although the convergence speed of EM algorithm is generally slower than other general-purpose numerical methods such as the Newton method, it has a global convergence property. This property allows us to reduce efforts in choosing good initial guesses for the model parameters and is suitable for automating the estimation procedure. In our past work [49], we summarized EM algorithms for 12 NHPP-based SRMs when the failure data were time data. The failure time data consisted of a set of exact failure times experienced. In practice, it is difficult to obtain exact failure times. Generally, we record failure count data consisting of the number of failures experienced for time intervals. For example, it is reasonable to record the number of failures per working day. From this reason, this paper presents the EM algorithms for 12 basic NHPP-based SRMs when the failure count data are given. In particular, we consider the generalized failure count data that involve both failure time and count data formats, and thus, the developed EM algorithms can be applied to either failure time data, failure count data, or their mixture.

We highlight our contributions here: (i) we derive the EM-step formula for NHPP-based SRMs with a finite number of failures under generalized fault count data, (ii) we derive concrete EM-step formulas for 12 basic NHPP-based SRMs, and (iii) we demonstrate the performance on the convergence property of the presented algorithms with real software failure data. To our best knowledge, this is the first paper that presents the EM algorithm for the generalized fault count data in 12 basic NHPP-based SRMs.

This paper is organized as follows. In Section 2, we introduce NHPP-based SRMs that are considered in this paper. In particular, NHPP-based SRMs are classified by failure time distribution and present the relationship between basic 12 NHPP-based SRMs and their failure time distributions. In Section 3, we derive the EM-step formulas for 12 basic NHPP-based SRMs. Section 4 is devoted to a numerical example to compare the convergence properties of EM algorithm, the Nelder–Mead method, and the quasi-Newton method. Finally, we conclude the paper with remarks in Section 5.

2. NHPP-Based SRMs

2.1. Model Description

Let $\{X(t), t \geq 0\}$ denote the number of software failures experienced before time t . We make the following model assumptions [21]:

- Assumption A: A software failure occurs at a random time. The probability distribution of all failure times are identical and mutually independent.
- Assumption B: The number of inherent software faults causing failures is finite.

Here, $F(t)$ and N are the cumulative distribution function of the failure time and the number of inherent faults. Then, the probability mass function of the cumulative number of failures experienced by time t is

$$P(X(t) = n) = \binom{N}{n} F(t)^n \bar{F}(t)^{N-n}, \tag{1}$$

where $\bar{F}(\cdot) = 1 - F(\cdot)$. This is often called the framework of generalized order statistics [21]. For instance, when the failure distribution is an exponential distribution, the corresponding SRM, the so-called exponential order statistics model, is the same as the Jelinski–Moranda SRM [1].

Most NHPP-based SRMs are advanced models of the generalized order statistics models. We make an additional model assumption [21]:

- Assumption C: The number of inherent faults is unknown, but prior information is given by a Poisson distribution.

When the expected number of inherent faults is ω , the cumulative number of software failures at time t has the following probability mass function:

$$P(X(t) = n) = \frac{(\omega F(t))^n}{n!} e^{-\omega F(t)}. \tag{2}$$

Equation (2) is equivalent to the probability mass function of NHPP with mean value function $\omega F(t)$. In this modeling framework, the failure time distribution $F(t)$ specifies an NHPP-based SRM.

Since the NHPP-based SRM is characterized by the failure time distribution, there have been a number of NHPP-based SRMs that change the failure time distribution. In this paper, we propose basic NHPP-based SRMs using well-known statistical distributions as the failure time distribution. Table 1 shows 11 basic NHPP-based SRMs and their failure time distributions. In the table, most of the basic NHPP-based SRMs correspond to the existing traditional NHPP-based SRMs. ‘exp’ is the so-called Goel and Okumoto model [2], ‘gamma’ is a generalized delayed S-shaped model [17,18], ‘pareto’ is a modified Duane model [50], ‘tlogis’ is an inflection S-shaped model [15], and ‘lxvmin’ is the Goel (Weibull) model [51].

Table 1. Basic NHPP-based SRMs.

Model	Failure Time Distribution
exp	Exponential distribution [2]
gamma	Gamma distribution [17,18]
pareto	Pareto type-II distribution [50]
tnorm	Truncated normal distribution [34]
lnorm	Log-normal distribution [34]
tlogis	Truncated logistic distribution [15]
llogis	Log-logistic distribution [52]
txvmax	Truncated extreme-value distribution (max) [35]
lxvmax	Log-extreme-value distribution (max) [35]
txvmin	Truncated extreme-value distribution (min) [35]
lxvmin	Log-extreme-value distribution (min) [35,51]

2.2. Parameter Estimation

As mentioned before, the model parameters of NHPP-based SRMs should be estimated from software failure data to predict the future tendency of a software failure. The most commonly used technique for parameter estimation is maximum likelihood (ML) estimation. In the context of ML estimation, we found model parameters that maximize the log-likelihood function (LLF). Since the LLF depends on the failure data experienced, the ML estimation of NHPP-based SRMs has been discussed for two types of data: failure time data and count data. The failure time data is a set of exact times in which a software failure occurs in the testing phase. The count data, equivalently called grouped data, consists of the number of failures experienced for time intervals. The estimation problems for these two data structures have been discussed separately.

This paper deals with a generalized data structure to express both failure time and count data. Our data structure is $\mathcal{D} := \{(t_1, x_1, u_1), \dots, (t_k, x_k, u_k)\}$, where x_i failures that occur at the i th time interval, (x_{i-1}, x_i) . In addition, if $u_i = 1$, an additional failure occurs at the end of the i th time interval, i.e, at time x_i . Otherwise, if $u_i = 0$, no failure occurs at the instant. If $u_i = 0$ for all time intervals, the data turns out the failure count data. If $x_i = 0$ and $u_i = 1$ for all i , \mathcal{D} is the failure time data.

Based on the generalized data, the LLF for NHPP-based SRMs is written in the following form:

$$\begin{aligned} \text{LLF}(\omega, \theta) = & \sum_{i=1}^k (x_i + u_i) \log \omega + \sum_{i=1}^k x_i \log \{F(t_i; \theta) - F(t_{i-1}; \theta)\} \\ & + \sum_{i=1}^k u_i \log f(t_i; \theta) - \log x_i! - \omega F(t_k; \theta). \end{aligned} \tag{3}$$

Then, the problem is to find the optimal (ω, θ) , so-called maximum likelihood estimates (MLEs), maximizing $\text{LLF}(\omega, \theta)$. However, it is noted that we cannot derive the closed form solution of MLEs. That is, we need to utilize numerical optimization techniques such as the Newton method, quasi-Newton method, and Nelder–Mead method.

Although conventional methods such as the Newton method and the Nelder–Mead method may be occasionally useful in computing MLEs of the NHPP-based SRMs, it is worth noting that these aim to solve unconstrained optimization problems in ML estimation. However, in many cases, we have to cope with constrained optimization problems because almost all of the model parameters of NHPP-based SRMs are implicitly constrained, such as positive constraint.

3. EM Algorithms for NHPP-Based SRMs

This paper develops numerical procedures to compute MLEs for NHPP-based SRMs with generalized data. The proposed estimation algorithms are based on the EM principle. The EM algorithm is one of the statistical approaches to compute the MLEs for incomplete data and is numerically stable because of its global convergence property. Moreover, the proposed EM algorithms for NHPP-based SRMs are based on the closed forms of MLEs for an arbitrary fault-detection time distribution and are capable of solving constrained optimization problems. Although we have already developed EM algorithms for failure time data and failure count data for several basic NHPP-based SRMs, this paper revisits their EM algorithm when generalized data are given.

3.1. EM Algorithm

The EM algorithm is an iterative method for computing ML estimates with incomplete data [28,29]. Let \mathcal{D} and \mathcal{U} be observable and unobservable data vectors, respectively, and θ be a model parameter vector θ to be estimated from only the observable data. In the ML estimation, we find a parameter vector by maximizing the following log-likelihood function (LLF) $\mathcal{L}(\theta; \mathcal{D})$:

$$\mathcal{L}(\theta; \mathcal{D}) = \log p(\mathcal{D}; \theta) = \log \int p(\mathcal{D}, \mathcal{U}; \theta) d\mathcal{U}, \tag{4}$$

where $p(\cdot)$ is any probability density or mass function and thus $p(\mathcal{D}, \mathcal{U}; \theta)$ denotes the likelihood function for complete data $(\mathcal{D}, \mathcal{U})$.

Let $Q(\theta|\theta')$ denote the conditional expected LLF with respect to the complete data vector $(\mathcal{D}, \mathcal{U})$ using the posterior distribution for unobservable data vector with a given parameter vector θ' :

$$\begin{aligned} Q(\theta|\theta') = & \text{E}[\log p(\mathcal{D}, \mathcal{U}; \theta) | \mathcal{D}; \theta'] \\ = & \int p(\mathcal{U} | \mathcal{D}; \theta') \log p(\mathcal{D}, \mathcal{U}; \theta) d\mathcal{U}. \end{aligned} \tag{5}$$

Then, the EM algorithm consists of an E-step and an M-step. The E-step computes the conditional expected LLF with respect to the complete data vector $(\mathcal{D}, \mathcal{U})$ using the posterior distribution for unobservable data vector with provisional parameter vector θ' , i.e., $Q(\theta|\theta')$. In the M-step, we find a new parameter vector θ'' that maximizes the expected LLF:

$$\theta'' := \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta'), \tag{6}$$

and θ'' becomes a provisional parameter vector at the next E- and M-steps. These steps surely increase the marginal LLF. The E- and M-steps are repeatedly executed until the parameters converge.

3.2. EM Algorithm for NHPP-Based SRMs

Consider the complete data in NHPP-based SRMs, $T_1 < T_2 < \dots < T_N$, where T_i is the i th failure time and N is the number of all the failures. It is worth noting that the number of all the failures in software is unobserved. Since N is the Poisson-distributed random variable and T_i obeys $F(\cdot; \theta)$, the complete LLF is given by

$$\text{LLF}(\omega, \theta) = N \log \omega - \omega + \sum_{i=1}^N \log f(T_i; \theta). \tag{7}$$

From the standard argument of MLEs, the MLEs of ω and θ can be derived as

$$\omega = N \tag{8}$$

and

$$\theta = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N \log f(T_i; \theta) \tag{9}$$

respectively. These imply that the estimation problem of NHPP-based SRMs under complete data can be decomposed into separate problems for two distribution functions: Poisson distribution and the failure time distribution.

Since the number of failures and the exact failure time in intervals are unobserved, the generalized data $\mathcal{D} := \{(t_1, x_1, u_1), \dots, (t_k, x_k, u_k)\}$ are incomplete data. By applying the EM algorithm, we have the following EM-step formulas for NHPP-based SRMs with the generalized data:

$$\omega \leftarrow E[N|\mathcal{D}; \omega', \theta'] \tag{10}$$

and

$$\theta \leftarrow \underset{\theta}{\operatorname{argmax}} E \left[\sum_{i=1}^N \log f(T_i; \theta) \middle| \mathcal{D}; \omega', \theta' \right] \tag{11}$$

Additionally, we obtain the following formula to compute the expected values. For any measurable function $h(\cdot)$, the expected value with the generalized data is expressed as

$$E \left[\sum_{i=1}^N h(T_i) \middle| \mathcal{D}; \omega', \theta' \right] = \sum_{i=1}^n \left\{ \frac{x_i \int_{t_{i-1}}^{t_i} h(z) f(z; \theta') dz}{\int_{t_{i-1}}^{t_i} f(z; \theta') dz} + u_i h(t_i) \right\} + \omega' \int_{t_k}^{\infty} h(z) f(z; \theta') dz, \tag{12}$$

where $f(z; \theta)$ is a probability density function (p.d.f.) of failure time provided that the parameter vector is θ . The derivation of this formula is given in Appendix A.

exp: ‘exp’ is the model where the failure time distribution is an exponential distribution. This model is exactly the same as the Goel–Okumoto model [2]. Define the c.d.f. of failure time as

$$F(t; \beta) = 1 - \exp(-\beta t), \quad \beta > 0. \tag{13}$$

Since the MLE of an ordinary exponential distribution is given by a closed form, the EM-step formulas for exp are directly derived from Equations (10) and (11);

$$\omega \leftarrow E[N|\mathcal{D}; \omega', \beta'] \tag{14}$$

$$\beta \leftarrow \frac{E[N|\mathcal{D}; \omega', \beta']}{E[\sum_{i=1}^N T_i|\mathcal{D}; \omega', \beta']}. \tag{15}$$

By applying the formula for the expected value, we have

$$\omega \leftarrow \sum_{i=1}^k (x_i + u_i) + \omega' \exp(-\beta' t_k) \tag{16}$$

$$\beta \leftarrow \frac{\sum_{i=1}^k (x_i + u_i) + \omega' \exp(-\beta' t_k)}{\sum_{i=1}^k (x_i \tau_i + u_i t_i) + \omega' (t_k + 1/\beta') \exp(-\beta' t_k)} \tag{17}$$

where

$$\tau_i = \frac{(t_{i-1} + 1/\beta') \exp(-\beta' t_{i-1}) - (t_i + 1/\beta') \exp(-\beta' t_i)}{\exp(-\beta' t_{i-1}) - \exp(-\beta' t_i)}. \tag{18}$$

gamma: The failure time distribution becomes the following gamma distribution:

$$F(t; \alpha, \beta) = \int_0^t \frac{\beta^\alpha u^{\alpha-1} \exp(-\beta u)}{\Gamma(\alpha)} du, \quad \alpha > 0, \quad \beta > 0, \tag{19}$$

where α and β are shape and scale parameters, respectively. When $\alpha = 2$ is fixed, the model reduces the delayed S-shaped SRM [17].

Similar to exp, the EM-step formulas are given using Equations (10) and (11):

$$\omega \leftarrow E[N|\mathcal{D}; \omega', \alpha', \beta'] \tag{20}$$

$$\alpha \leftarrow \left\{ \alpha \left| \log \alpha - \Psi(\alpha) = \log \left(\frac{E[\sum_{i=1}^N T_i|\mathcal{D}; \omega', \alpha', \beta']}{E[N|\mathcal{D}; \omega', \alpha', \beta']} \right) - \frac{E[\sum_{i=1}^N \log T_i|\mathcal{D}; \omega', \alpha', \beta']}{E[N|\mathcal{D}; \omega', \alpha', \beta']} \right. \right\} \tag{21}$$

$$\beta \leftarrow \frac{\alpha E[N|\mathcal{D}; \omega', \alpha', \beta']}{E[\sum_{i=1}^N T_i|\mathcal{D}; \omega', \alpha', \beta']} \tag{22}$$

where $\psi(\cdot)$ is the digamma function, i.e., $\psi(\alpha) = d \log \Gamma(\alpha) / d\alpha$. Additionally, we use the updated α to compute β . Note that Equation (21) can easily be solved with the non-linear equation solver such as a bisection method. In addition, $E[N|\mathcal{D}; \omega', \alpha', \beta']$ and $E[\sum_{i=1}^N T_i|\mathcal{D}; \omega', \alpha', \beta']$ are obtained as follows:

$$E[N|\mathcal{D}; \omega', \alpha', \beta'] = \sum_{i=1}^k (x_i + u_i) + \omega' \bar{F}(t_k; \alpha', \beta') \tag{23}$$

$$E \left[\sum_{i=1}^N T_i \middle| \mathcal{D}; \omega', \alpha', \beta' \right] = \sum_{i=1}^k (x_i \tau_i + u_i t_i) + \omega' (\alpha' / \beta') \bar{F}(t_k; \alpha' + 1, \beta'), \tag{24}$$

$$\tau_i = \frac{\alpha' \bar{F}(t_{i-1}; \alpha' + 1, \beta') - \bar{F}(t_i; \alpha' + 1, \beta')}{\beta' (\bar{F}(t_{i-1}; \alpha', \beta') - \bar{F}(t_i; \alpha', \beta'))}, \tag{25}$$

where $\bar{F}(t; \alpha, \beta)$ is the complementary c.d.f. of gamma distribution with parameters α and β . On the other hand, we need the numerical integration to obtain $E[\sum_{i=1}^N \log T_i|\mathcal{D}; \omega', \alpha', \beta']$. It should be noted that, if the shape parameter α is fixed, then the computation algorithm becomes simple because we ignore solving the nonlinear equation and computing the expected value $E[\sum_{i=1}^N \log T_i|\mathcal{D}; \omega', \alpha', \beta']$.

pareto: ‘pareto’ is the SRM where the failure time distribution is the Pareto distribution of the second kind. The Pareto distribution of the second kind is called Lomax distribution:

$$F(t) = 1 - \frac{c^a}{(x + c)^a}, \quad a > 0, \quad c > 0. \tag{26}$$

This model was proposed as the modified Duane model [50].

Since the Pareto distribution of the second kind is a mixture of exponential distribution, the EM algorithm for ‘pareto’ is constructed using this property. In general, the mixture distribution is defined as a superposition of original statistical distributions with mixture ratio. Let $G(\zeta; \theta)$ be the c.d.f. of mixture ratio distribution for the parameter ζ . Then, the mixture distribution is given by

$$F_M(x; \theta) = \int F(x; \zeta) dG(\zeta; \theta). \tag{27}$$

The Pareto distribution of the second kind is a mixture of exponential distribution when the mixture ratio distribution is a gamma distribution. That is, the failure time distribution is written in the following form:

$$F(t; a, c) = \int_0^\infty \{1 - \exp(-\zeta t)\} \frac{c^a \zeta^{a-1} \exp(-c\zeta)}{\Gamma(a)} d\zeta = 1 - \frac{c^a}{(c + t)^a}. \tag{28}$$

For the EM algorithm of mixture-type SRMs, we also define the fault detection rate for each fault as a hidden variable.

Let $(T_1, \Xi_1), \dots, (T_N, \Xi_N)$ be a set of failure time and its associated fault detection rate for all the failures. The complete LLF is given by

$$\begin{aligned} \text{LLF}(\omega, a, c) = & N \log \omega - \omega + \sum_{i=1}^N \log \Xi_i - \sum_{i=1}^N \Xi_i T_i \\ & + aN \log c + (a - 1) \sum_{i=1}^N \log \Xi_i - c \sum_{i=1}^N \Xi_i - N \log \Gamma(a). \end{aligned} \tag{29}$$

Similar to gamma, we have the following EM-step formula from the MLEs of gamma distributions:

$$\omega \leftarrow E[N | \mathcal{D}; \omega', a', c'], \tag{30}$$

$$a \leftarrow \left\{ a \left| \log a - \psi(a) = \log \left(\frac{E[\sum_{i=1}^N \Xi_i | \mathcal{D}; \omega', a', c']}{E[N | \mathcal{D}; \omega', a', c']} \right) - \frac{E[\sum_{i=1}^N \log \Xi_i | \mathcal{D}; \omega', a', c']}{E[N | \mathcal{D}; \omega', a', c']} \right\} \tag{31}$$

$$b \leftarrow \frac{a E[N | \mathcal{D}; \omega', a', c']}{E[\sum_{i=1}^N \Xi_i | \mathcal{D}; \omega', a', c']}. \tag{32}$$

On the other hand, the formula for the expected value is given by

$$E \left[\sum_{i=1}^N h(\Xi_i) \middle| \mathcal{D}; \omega', \theta' \right] = \sum_{i=1}^k \left\{ \frac{x_i \int_{t_{i-1}}^{t_i} \tilde{h}(z; \theta') dz}{\int_{t_{i-1}}^{t_i} \tilde{f}(z; \theta') dz} + \frac{u_i \tilde{h}(t_i)}{\tilde{f}(t_i)} \right\} + \omega' \int_{t_k}^\infty \tilde{h}(z; \theta') dz, \tag{33}$$

where $h(\cdot)$ is an arbitrary measurable function and

$$\tilde{h}(z; \theta') = \int h(\zeta) f(z; \zeta) dG(\zeta; \theta'), \tag{34}$$

$$\tilde{f}(z; \theta') = \int f(z; \zeta) dG(\zeta; \theta'). \tag{35}$$

tnorm, Inorm: ‘tnorm’ and ‘Inorm’ are SRMs whose failure time distributions are truncated and log normal distributions, respectively. The failure time distributions for tnorm and Inorm are

$$\text{tnorm: } F(t) = \Phi\left(\frac{t - \mu}{\sigma}\right) / \{1 - \Phi(-\mu/\sigma)\}, \tag{36}$$

$$\text{Inorm: } F(t) = \Phi\left(\frac{\log t - \mu}{\sigma}\right), \tag{37}$$

where $\Phi(\cdot)$ is the c.d.f. of the standard normal distribution. Since the EM algorithms for both models with failure time and count data were introduced in detail in the literature [34], this paper provides the EM-step formulas with the generalized data.

- EM-step formula for tnorm:

$$\tilde{\omega} \leftarrow N, \quad \mu \leftarrow T^{(1)}/N, \quad \sigma \leftarrow \sqrt{T^{(2)}/N - (T^{(1)}/N)^2} \tag{38}$$

where

$$N = \sum_{i=1}^k (x_i + u_i) + \tilde{\omega} \{ \Phi(z_0) + \bar{\Phi}(z_k) \}, \tag{39}$$

$$T^{(1)} = \sum_{i=1}^k (x_i \tau_i^{(1)} + u_i t_i) + \tilde{\omega} \{ \Phi^{(1)}(z_0) + \bar{\Phi}^{(1)}(z_k) \} \tag{40}$$

$$T^{(2)} = \sum_{i=1}^k (x_i \tau_i^{(2)} + u_i t_i^2) + \tilde{\omega} \{ \Phi^{(2)}(z_0) + \bar{\Phi}^{(2)}(z_k) \}, \tag{41}$$

$$z_0 = -\mu/\sigma, \quad z_i = (t_i - \mu)/\sigma, \tag{42}$$

$$\tau_i^{(u)} = \frac{\bar{\Phi}^{(u)}(z_{i-1}) - \bar{\Phi}^{(u)}(z_i)}{\bar{\Phi}(z_{i-1}) - \bar{\Phi}(z_i)}, \tag{43}$$

- EM-step formula for Inorm:

$$\omega \leftarrow N, \quad \mu \leftarrow T^{(1)}/N, \quad \sigma \leftarrow \sqrt{T^{(2)}/N - (T^{(1)}/N)^2} \tag{44}$$

where

$$N = \sum_{i=1}^k (x_i + u_i) + \omega \bar{\Phi}(z_k), \tag{45}$$

$$T^{(1)} = \sum_{i=1}^k (x_i \tau_i^{(1)} + u_i \log t_i) + \omega \bar{\Phi}^{(1)}(z_k), \tag{46}$$

$$T^{(2)} = \sum_{i=1}^k (x_i \tau_i^{(2)} + u_i (\log t_i)^2) + \omega \bar{\Phi}^{(2)}(z_k), \tag{47}$$

$$z_0 \rightarrow -\infty, \quad z_i = (\log t_i - \mu)/\sigma, \tag{48}$$

$$\tau_i^{(u)} = \frac{\bar{\Phi}^{(u)}(z_{i-1}) - \bar{\Phi}^{(u)}(z_i)}{\bar{\Phi}(z_{i-1}) - \bar{\Phi}(z_i)}. \tag{49}$$

In the above formulas, $\bar{\Phi}(z)$ is the complementary c.d.f. of the standard normal distribution, and $\bar{\Phi}^{(1)}(z)$ and $\bar{\Phi}^{(2)}(z)$ are expressed with the p.d.f. of the standard normal distribution $\phi(z)$:

$$\bar{\Phi}^{(1)}(z) = \sigma\phi(z) + \mu\bar{\Phi}(z), \tag{50}$$

$$\bar{\Phi}^{(2)}(z) = (\sigma^2z + 2\mu\sigma)\phi(z) + (\sigma^2 + \mu^2)\bar{\Phi}(z). \tag{51}$$

In addition, after the convergence, we take $\omega = \tilde{\omega}\bar{\Phi}(z_0)$ to obtain the ML estimate for ω in the case of tnorm.

tlogis, llogis tlogis and llogis are the SRMs with truncated and log logistic distributions, respectively. In particular, ‘tlogis’ is equivalent to the inflection S-shaped model [15]. The failure time distribution of tlogis is given by

$$F(t) = \Psi\left(\frac{t - \mu}{\psi}\right) / \{1 - \Psi(-\mu/\psi)\}, \tag{52}$$

where $\Psi(\cdot)$ is the c.d.f. of standard logistic distribution

$$\Psi(t) = \frac{1}{1 + \exp(-t)}. \tag{53}$$

By taking into account the exponential of logistic distribution, we have the following failure time distribution of llogis:

$$F(t) = \Psi\left(\frac{\log t - \mu}{\psi}\right). \tag{54}$$

Since logistic distribution does not belongs to the exponential family of distributions, neither expectation nor maximization can be expressed as simple formulas. To construct the algorithm, we consider only one assumption; the number of all failures is not observed. Then, the EM-step formulas become

- The EM-step formula for tnorm

$$\tilde{\omega} \leftarrow \sum_{i=1}^k (x_i + u_i) + \tilde{\omega}'F(0; \theta') + \tilde{\omega}'\bar{F}(t_k; \theta') \tag{55}$$

$$\theta \leftarrow \operatorname{argmax}_{\theta} \left\{ \sum_{i=1}^k (x_i \log(F(t_i; \theta) - F(t_{i-1}; \theta)) - x_i \log x_i! + u_i \log f(t_i; \theta)) + (\tilde{\omega}'F(0; \theta')) \log(F(0; \theta)) + (\tilde{\omega}'\bar{F}(t_k; \theta')) \log(\bar{F}(t_k; \theta)) \right\}. \tag{56}$$

- The EM-step formula for lnorm

$$\omega \leftarrow \sum_{i=1}^k (x_i + u_i) + \omega'\bar{F}(t_k; \theta') \tag{57}$$

$$\theta \leftarrow \operatorname{argmax}_{\theta} \left\{ \sum_{i=1}^k (x_i \log(F(t_i; \theta) - F(t_{i-1}; \theta)) - x_i \log x_i! + u_i \log f(t_i; \theta)) + (\omega'\bar{F}(t_k; \theta')) \log(\bar{F}(t_k; \theta)) \right\}. \tag{58}$$

The second equations in both formulas indicate that θ is updated by the MLEs when the number of all the failures is given by $\tilde{\omega}'$ and ω' . These algorithm are also stable if there

exists a unique solution maximizing the right-hand side of the second term. Note that, after the convergence, the model parameter ω in tlogis can be obtained as $\omega = \tilde{\omega}\bar{F}(0; \theta)$.

txvmax, lxvmax, txvmin, lxvmin Suppose that the failure time caused by each failure follows an extreme value type I distribution. The extreme value type I distribution is called Gumbel distribution, and its definition is based on the limitation of the maximum value of random variables. Here, the c.d.f. of a standard Gumbel distribution is defined as

$$\Theta(t) = \exp\{-\exp(-t)\}. \tag{59}$$

Similar to tnorm, lnorm, tlogis, and llogis, we consider the truncation and logarithm of the extreme value distribution. In addition, since the extreme value distribution is not symmetric, we also consider the case of negative samples, i.e., the minimum value of random variables.

The failure time distributions of txvmax and lxvmax are, respectively,

$$F(t) = \Theta\left(\frac{t-\mu}{\theta}\right) / \{1 - \Theta(-\mu/\theta)\}, \tag{60}$$

$$F(t) = \Theta\left(\frac{\log t - \mu}{\theta}\right). \tag{61}$$

Similarly, the failure time distributions of txvmin and lxvmin are given by

$$F(t) = \bar{\Theta}\left(\frac{t+\mu}{\theta}\right) / \{1 - \bar{\Theta}(\mu/\theta)\}, \tag{62}$$

$$F(t) = \bar{\Theta}\left(\frac{\log t + \mu}{\theta}\right), \tag{63}$$

where $\bar{\Theta}(t) = 1 - \Theta(-t)$ corresponds to the c.d.f. of a standard extreme value type I distribution of the minimum. From Equation (63), we find that lxvmin is equivalent to the Weibull distribution.

Since the extreme value distribution is not an exponential family, we consider only one assumption; the number of all the failures is not observed. Then, the EM-step formulas are given by

- The EM-step formula for txvmax and txvmin

$$\tilde{\omega} \leftarrow \sum_{i=1}^k (x_i + u_i) + \tilde{\omega}'F(0; \theta') + \tilde{\omega}'\bar{F}(t_k; \theta') \tag{64}$$

$$\theta \leftarrow \operatorname{argmax}_{\theta} \left\{ \sum_{i=1}^k (x_i \log(F(t_i; \theta) - F(t_{i-1}; \theta)) - x_i \log x_i! + u_i \log f(t_i; \theta)) + (\tilde{\omega}'F(0; \theta')) \log(F(0; \theta)) + (\tilde{\omega}'\bar{F}(t_k; \theta')) \log(\bar{F}(t_k; \theta)) \right\}. \tag{65}$$

- The EM-step formula for lxvmax and lxvmin

$$\omega \leftarrow \sum_{i=1}^k (x_i + u_i) + \omega'\bar{F}(t_k; \theta') \tag{66}$$

$$\theta \leftarrow \operatorname{argmax}_{\theta} \left\{ \sum_{i=1}^k (x_i \log(F(t_i; \theta) - F(t_{i-1}; \theta)) - x_i \log x_i! + u_i \log f(t_i; \theta)) + (\omega'\bar{F}(t_k; \theta')) \log(\bar{F}(t_k; \theta)) \right\}. \tag{67}$$

4. Numerical Example

We investigated the numerical characteristics of the presented EM algorithms. Here, we compare the convergence property with the Nelder–Mead method and the quasi-Newton method (BFGS method). First, we check the trace of model parameters until they converge to MLE for the proposed method (EM algorithm), the Nelder–Mead method, and the BFGS method. In this experiment, we used the fault count data, which were collected from real software projects [53]. The statistics of fault count data are given as follow.

- Data label: SS1A
- Working days: 151
- The number of failures: 112
- LOC: Hundreds of thousands
- Software type: Operations

For the above failure count data, we estimated the parameters of ‘exp’. The MLEs of exp are $\hat{\omega} = 354.75$ and $\hat{\beta} = 0.00251$, and the maximum LLF is $\text{LLF}(\hat{\omega}, \hat{\beta}) = -180.79$.

Figures 1–3 show the trace of model parameters for EM algorithm, the Nelder–Mead method, and the BFGS method when the initial guesses are $\omega = 100$ and $\beta = 0.1$. We use the ‘optim’ function in R for the Nelder–Mead and BFGS methods. From these figures, we find that the EM algorithm stably updates the model parameters and converges to close parameters to the MLEs. However, the convergence speed is not fast, since the update becomes smaller as the parameters are close to the MLEs. The Nelder–Mead method provides the MLEs, but the trace of the algorithm is not stable. In particular, this algorithm sometimes takes invalid values that violate the parameter constraints, i.e., $\omega < 0$ or $\beta < 0$, while searching for the parameters. Figure 3 depicts the trace of parameters for the BFGS method. The convergence property is the worst among the three methods. Additionally, the algorithm fails to obtain the MLE.

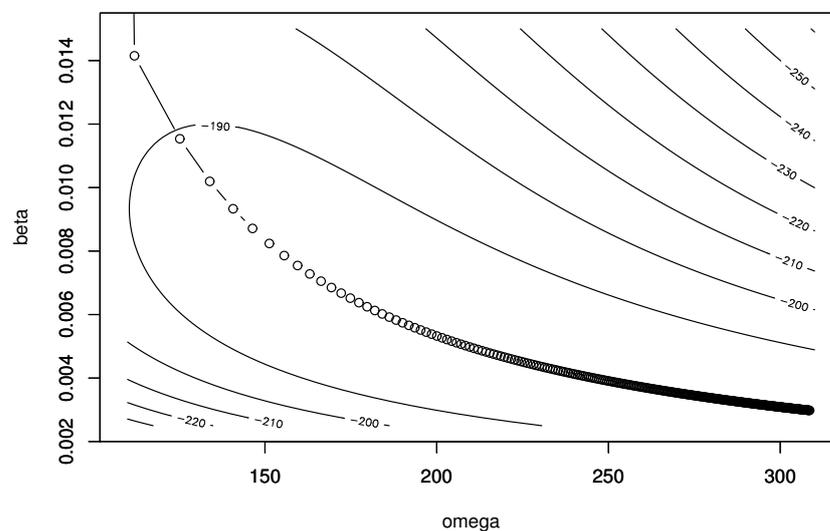


Figure 1. Trace of parameters in the EM algorithm.

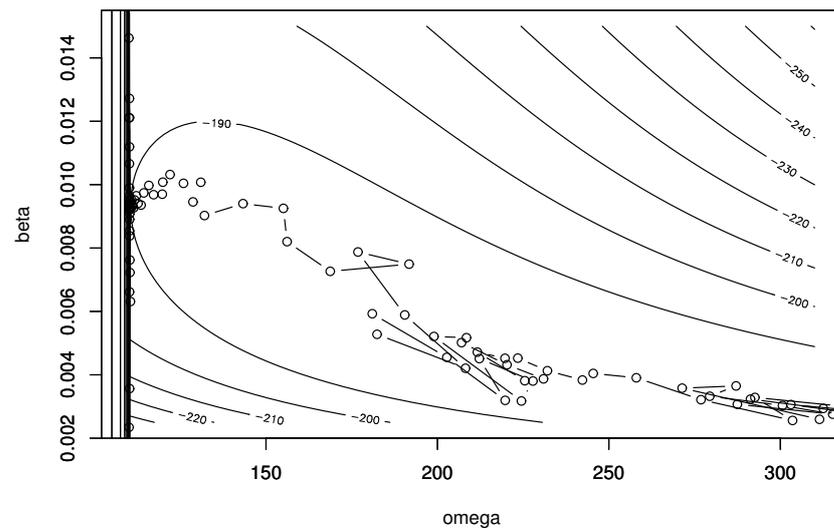


Figure 2. Trace of parameters in the Nelder–Mead method.

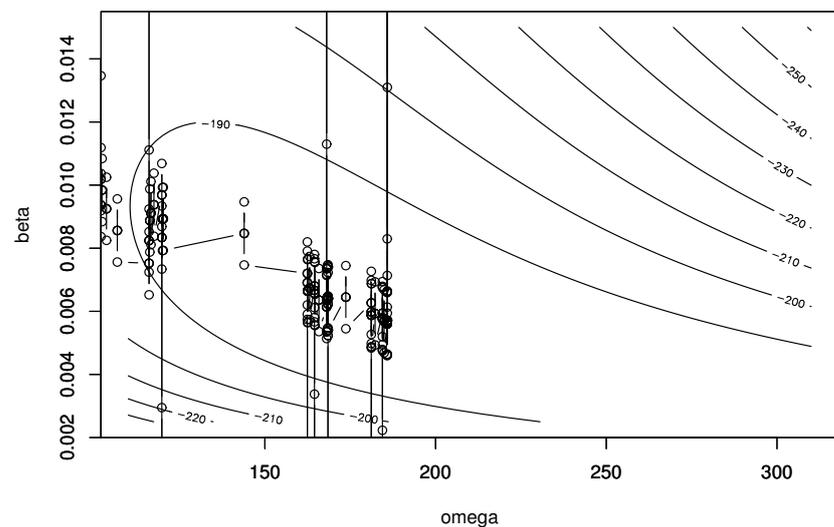


Figure 3. Trace of parameters in the BFGS method.

Next, we present the convergence properties for the proposed EM algorithm, the Nelder–Mead method, and the BFGS method quantitatively. Here, we use two additional fault count data that were collected from real software projects [53] as well as SS1A.

- Data label: SS1B
- Working days: 663
- The number of failures: 375
- LOC: Hundreds of thousands
- Software type: Operations
- Data label: SS1C
- Working days: 472
- The number of failures: 277
- LOC: Hundreds of thousands
- Software type: Operations

For three data sets—SS1A, SS1B, and SS1C—we applied the proposed EM algorithm, the Nelder–Mead method, and the BFGS method for 12 basic NHPP-based SRMs with 100 different initial parameters. In the experiment, the initial parameters were selected by random numbers. Tables 2–4 present the number of converged estimations, i.e., the number

of times that the model parameters are successfully estimated for each NHPP-based SRMs and methods. If this value is 100, the method succeeded in obtaining the MLE for all of the initial parameters. On the other hand, if this value is 0, the estimation method fails to obtain the MLE for all of the initial parameters due to numerical computation errors such as overflow and underflow.

Table 2. The number of converged estimations (SS1A).

Model	EM	Nelder-Mead	BFGS
exp	100	19	19
gamma	100	34	34
pareto	100	57	57
tnorm	100	88	88
lnorm	100	98	98
tlogis	100	100	100
llogis	100	100	100
txvmax	100	100	100
lxvmax	99	99	99
txvmin	64	65	65
lxvmin	92	92	92

Table 3. The number of converged estimations (SS1B).

Model	EM	Nelder-Mead	BFGS
exp	100	7	7
gamma	100	7	6
pareto	100	18	18
tnorm	100	98	98
lnorm	87	87	87
tlogis	100	100	100
llogis	87	87	87
txvmax	99	99	99
lxvmax	0	0	0
txvmin	89	89	89
lxvmin	100	41	41

Table 4. The number of converged estimations (SS1C).

Model	EM	Nelder-Mead	BFGS
exp	100	9	8
gamma	100	13	12
pareto	100	47	47
tnorm	100	98	98
lnorm	96	97	97
tlogis	100	100	100
llogis	96	96	96
txvmax	99	99	99
lxvmax	0	0	0
txvmin	89	89	89
lxvmin	100	43	43

From these results, we find that the convergence rates of the proposed EM algorithms are 100% in the cases of exp, gamma, pareto, tnorm, and tlogis. Since the number of converged estimations of the Nelder–Mead is almost the same as that of BFGS for all cases, the convergence properties of their methods are the same if we use the ‘optim’ function in R. Additionally, since lxvmax did not fit SS1B and SS1C, all of the estimation methods failed to obtain the MLE. Furthermore, it was found that the numbers of converged estimates in the Nelder–Mead and BFGS methods are worse than that of the EM algorithm, specifically

in the cases of exp, gamma, and pareto. Additionally, in the cases of tnorm and lnorm, the convergence property of EM is slightly superior to the other two methods. In exp, gamma, pareto, tnorm, and lnorm, the failure time distributions belong to the exponential family, and thus, their EM-step formulas do not include the numerical optimization step. That is, these EM-step formulas are ‘pure’ EM-step formulas. Therefore, the convergence properties outperform those of the Nelder–Mead and BFGS methods. On the other hand, in the cases of flogis, llogis, txvmax, lxvmax, txvmin, and lvxmin, the failure time distributions are not in the exponential family, and we should use the numerical optimization step in their EM-step formulas. This is the reason why the convergence property of the presented EM algorithm is same as that in the Nelder–Mead and BFGS methods.

5. Conclusions

This paper derived EM-step formulas for 12 basic NHPP-based SRMs when the generalized failure count data are given. Since the generalized fault count data involve both time and count data formats, the presented EM algorithms can be applied to failure data experienced in practice. In addition, the convergence property of EM algorithm is better than or equivalent to other ordinary methods such as the Nelder–Mead and BFGS methods for practical software fault data. Thus, the presented algorithms are suitable for implementation in the automatic tool for software reliability evaluation. In fact, our research group has developed an AddIn of Microsoft Excel to estimate software reliability [54].

In the future, we will develop a reliability assessment tool by integrating a software repository such as GitHub, a bug tracking system, and a continuous integration system, and the tool will continuously monitor the reliability of software.

Author Contributions: Conceptualization, H.O. and T.D.; methodology, H.O.; software, H.O.; supervision, T.D.; validation, T.D.; writing—original draft, H.O. and T.D.; writing—review and editing, H.O. and T.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of Equation (12)

For convenient, ω' and θ' are written as ω and θ , respectively, and $E[\cdot|\mathcal{D}; \omega', \theta']$ is simplified as $E[\cdot|\mathcal{D}]$. Here we have

$$E \left[\sum_{i=1}^N h(T_i) \middle| \mathcal{D} \right] = \sum_{i=1}^n \left\{ E \left[\sum_{j=s_{i-1}+1}^{s_i+x_i} h(T_j) \middle| \mathcal{D} \right] + b_i E[h(T_{s_i})|\mathcal{D}] \right\} + E \left[\sum_{i=s_n+1}^N h(T_i) \middle| \mathcal{D} \right], \tag{A1}$$

where $s_i = \sum_{j=1}^i (x_j + b_j)$.

According to the order statistics of failure times, the first term of the right-hand side of the above can be rewritten as follows.

$$E \left[\sum_{j=s_{i-1}+1}^{s_i+x_i} h(T_j) \middle| \mathcal{D} \right] = \frac{\int_{t_{i-1}}^{t_i} \int_{z_1}^{t_i} \dots \int_{z_{x_i}}^{\sum_{j=1}^{x_i} h(z_j)} \prod_{j=1}^{x_i} f(z_j) dz_{x_i} \dots dz_1}{\int_{t_{i-1}}^{t_i} \int_{z_1}^{t_i} \dots \int_{z_{x_i}}^{\sum_{j=1}^{x_i} f(z_j)} dz_{x_i} \dots dz_1}. \tag{A2}$$

Since $T_{s_{i-1}+1}, \dots, T_{s_{i-1}+x_i}$ are i.i.d. random variables, the multiple integrals of denominator in Equation (A2) is given by

$$\int_{t_{i-1}}^{t_i} \int_{z_1}^{t_i} \dots \int_{z_{x_i}} \prod_{j=1}^{x_i} f(z_j) dz_{x_i} \dots dz_1 = \frac{1}{x_i!} \left(\int_{t_{i-1}}^{t_i} f(z) dz \right)^{x_i}. \tag{A3}$$

Similarly, the numerator becomes

$$\int_{t_{i-1}}^{t_i} \int_{z_1}^{t_i} \dots \int_{z_{x_i}} \sum_{j=1}^{x_i} h(z_j) \prod_{j=1}^{x_i} f(z_j) dz_{x_i} \dots dz_1 = \frac{x_i}{x_i!} \int_{t_{i-1}}^{t_i} h(z) f(z) dz \left(\int_{t_{i-1}}^{t_i} f(z) dz \right)^{x_i-1}. \tag{A4}$$

Henceforth we have

$$E \left[\sum_{j=s_{i-1}+1}^{s_{i-1}+x_i} h(T_j) \middle| \mathcal{D} \right] = \frac{x_i \int_{t_{i-1}}^{t_i} h(z) f(z) dz}{\int_{t_{i-1}}^{t_i} f(z) dz}. \tag{A5}$$

The second term of the right-hand side of Equation (A1) is straightforwardly given by $h(t_i)$. The third term can be derived by a similar way to the first term. Taking account of the condition $N = v$, we have

$$\begin{aligned} E \left[\sum_{i=s_n+1}^N h(T_i) \middle| \mathcal{D} \right] &= \frac{\sum_{v=s_k}^{\infty} e^{-\omega \frac{\omega^v}{v!} \frac{v!(v-s_k)}{(v-s_k)!} \int_{t_k}^{\infty} h(z) f(z) dz \bar{F}(t_k)^{v-s_k-1}}}{\sum_{v=s_k}^{\infty} e^{-\omega \frac{\omega^v}{v!} \frac{v!(v-s_k)}{(v-s_k)!} \bar{F}(t_k)^{v-s_k}}} \\ &= \omega \int_{t_k}^{\infty} h(z) f(z) dz, \end{aligned} \tag{A6}$$

where $\bar{F}(t) = 1 - F(t)$.

References

1. Jelinski, Z.; Moranda, P.B. Software Reliability Research. In *Statistical Computer Performance Evaluation*; Freiberger, W., Ed.; Academic Press: New York, NY, USA, 1972; pp. 465–484.
2. Goel, A.L.; Okumoto, K. Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Trans. Reliab.* **1979**, R-28, 206–211. [\[CrossRef\]](#)
3. Lyu, M.R. (Ed.) *Handbook of Software Reliability Engineering*; McGraw-Hill: New York, NY, USA, 1996.
4. Musa, J.D.; Iannino, A.; Okumoto, K. *Software Reliability, Measurement, Prediction, Application*; McGraw-Hill: New York, NY, USA, 1987.
5. Pham, H. *Software Reliability*; Springer: Singapore, 2000.
6. Xie, M. *Software Reliability Modelling*; World Scientific: Singapore, 1991.
7. Rook, P. (Ed.) *Software Reliability Handbook*; Elsevier Science: London, UK, 1990.
8. Singpurwalla, N.D.; Wilson, S.P. *Statistical Methods in Software Engineering*; Springer: New York, NY, USA, 1997.
9. Li, Q.; Pham, H. A Generalized Software Reliability Growth Model With Consideration of the Uncertainty of Operating Environments. *IEEE Access* **2019**, 7, 84253–84267. [\[CrossRef\]](#)
10. Li, Q.; Pham, H. NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage. *Appl. Math. Model.* **2017**, 51, 68–85. [\[CrossRef\]](#)
11. He, Y. NHPP software reliability growth model incorporating fault detection and debugging. In *Proceedings of the 2013 IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013*; pp. 225–228.
12. Song, K.Y.; Chang, I.H.; Pham, H. NHPP Software Reliability Model with Inflection Factor of the Fault Detection Rate Considering the Uncertainty of Software Operating Environments and Predictive Analysis. *Symmetry* **2019**, 11, 521. [\[CrossRef\]](#)
13. Sun, H.; Zhang, L.; Wu, J.; Wu, J.; Yang, H. A New Method of Model Combination Based on the NHPP Software Reliability Models. In *ICMSS 2018: Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*; ACM: New York, NY, USA, 2018; pp. 153–158. [\[CrossRef\]](#)
14. Musa, J.D.; Okumoto, K. A Logarithmic Poisson Execution Time Model for Software Reliability Measurement. In *Proceedings of the 7th International Conference Software Engineering (ICSE-1084)*; IEEE CS Press: Los Alamitos, CA, USA; ACM: New York, NY, USA, 1984; pp. 230–238.
15. Ohba, M. Software Reliability Analysis. *IBM J. Res. Dev.* **1984**, 28, 428–443. [\[CrossRef\]](#)
16. Ohba, M. Inflection S-Shaped Software Reliability Growth Model. In *Stochastic Models in Reliability Theory*; Osaki, S., Hatoyama, Y., Eds.; Springer: Berlin, Germany, 1984; pp. 144–165.

17. Yamada, S.; Ohba, M.; Osaki, S. S-Shaped Reliability Growth Modeling for Software Error Detection. *IEEE Trans. Reliab.* **1983**, R-32, 475–478. [[CrossRef](#)]
18. Zhao, M.; Xie, M. On Maximum Likelihood Estimation for a General non-Homogeneous Poisson Process. *Scand. J. Stat.* **1996**, *23*, 597–607.
19. Pham, H.; Zhang, X. An NHPP Software Reliability Models and its Comparison. *Int. J. Reliab. Qual. Safe. Eng.* **1997**, *4*, 269–282. [[CrossRef](#)]
20. Shanthikumar, J.G. A General Software Reliability Model for Performance Prediction. *Microelectron. Reliab.* **1981**, *21*, 671–682. [[CrossRef](#)]
21. Langberg, N.; Singpurwalla, N.D. Unification of Some Software Reliability Models. *SIAM J. Sci. Comput.* **1985**, *6*, 781–790. [[CrossRef](#)]
22. Chen, Y.; Singpurwalla, N.D. Unification of Software Reliability Models by Self-Exciting Point Processes. *Adv. Appl. Probab.* **1997**, *29*, 337–352. [[CrossRef](#)]
23. Miller, D.R. Exponential Order Statistic Models of Software Reliability Growth. *IEEE Trans. Softw. Eng.* **1986**, SE-12, 12–24. [[CrossRef](#)]
24. Hossain, S.A.; Dahiya, R.C. Estimating the Parameters of a non-Homogeneous Poisson-Process Model for Software Reliability. *IEEE Trans. Reliab.* **1993**, *42*, 604–612. [[CrossRef](#)]
25. Knafel, G.; Morgan, J. Solving ML Equations for 2-parameter Poisson-Process Model for Ungrouped Software Failure Data. *IEEE Trans. Reliab.* **1996**, *45*, 42–53. [[CrossRef](#)]
26. Joe, H. Statistical Inference for General-Order-Statistics and Nonhomogeneous-Poisson-Process Software Reliability Models. *IEEE Trans. Softw. Eng.* **1989**, *15*, 1485–1490. [[CrossRef](#)]
27. Jeske, D.R.; Pham, H. On the maximum likelihood estimates for the Goel-Okumoto software reliability model. *Am. Stat.* **2001**, *55*, 219–222. [[CrossRef](#)]
28. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* **1977**, B-39, 1–38.
29. Wu, C.F.J. On the convergence properties of the EM algorithm. *Ann. Stat.* **1983**, *11*, 95–103. [[CrossRef](#)]
30. Okamura, H.; Dohi, T.; Osaki, S. EM algorithms for logistic software reliability models. In Proceedings of the 22nd IASTED International Conference on Software Engineering, Innsbruck, Austria, 17–19 February 2004; ACTA Press: Crete, Greece, 2004; pp. 263–268.
31. Okamura, H.; Murayama, A.; Dohi, T. EM Algorithm for Discrete Software Reliability Models: A Unified Parameter Estimation Method. In Proceedings of the 8th IEEE International Symposium High Assurance Systems Engineering, Tampa, FL, USA, 25–26 March 2004; pp. 219–228.
32. Okamura, H.; Watanabe, Y.; Dohi, T. Estimating a mixed software reliability models based on the EM algorithm. In Proceedings of the International Symposium on Empirical Software Engineering (ISESE2002), Nara, Japan, 3–4 October 2002; IEEE Computer Society Press: Los Alamitos, CA, USA, 2002; pp. 69–78.
33. Okamura, H.; Watanabe, Y.; Dohi, T. An iterative scheme for maximum likelihood estimation in software reliability modeling. In Proceedings of the 14th International Symposium on Software Reliability Engineering, Denver, CO, USA, 17–20 November 2003; pp. 246–256.
34. Okamura, H.; Dohi, T.; Osaki, S. Software reliability growth models with normal failure time distributions. *Reliab. Eng. Syst. Saf.* **2013**, *116*, 135–141. [[CrossRef](#)]
35. Ohishi, K.; Okamura, H.; Dohi, T. Gompertz software reliability model: Estimation algorithm and empirical validation. *J. Syst. Softw.* **2009**, *82*, 535–543. [[CrossRef](#)]
36. Okamura, H.; Dohi, T. Phase-type software reliability model: Parameter estimation algorithms with grouped data. *Ann. Oper. Res.* **2016**, *244*, 177–208. [[CrossRef](#)]
37. Okamura, H.; Dohi, T. Hyper-Erlang software reliability model. In Proceedings of the 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing, Taipei, Taiwan, 15–17 December 2008.
38. Okamura, H.; Dohi, T. Building Phase-Type Software Reliability Model. In Proceedings of the 2006 17th International Symposium on Software Reliability Engineering, Raleigh, NC, USA, 7–10 November 2006; pp. 289–298.
39. Xiao, X.; Okamura, H.; Dohi, T. NHPP-based software reliability models using equilibrium distribution. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci. (A)* **2012**, E95-A, 894–902. [[CrossRef](#)]
40. Kawasaki, M.; Okamura, H.; Dohi, T. A Comprehensive Evaluation of Software Reliability Modeling Based on Marshall-Olkin Type Fault-Detection Time Distribution. In Proceedings of the 2017 24th Asia-Pacific Software Engineering Conference (APSEC), Nanjing, China, 4–8 December 2017; pp. 486–494. [[CrossRef](#)]
41. Kimura, M.; Yamada, S. Software reliability management: Techniques and applications. In *Handbook of Reliability Engineering*; Pham, H., Ed.; Springer: London, UK, 2003; pp. 265–284.
42. Ledoux, J.; Rubino, G. A counting model for software reliability analysis. *Int. J. Model. Simul.* **1997**, *17*, 289–297. [[CrossRef](#)]
43. Okamura, H.; Dohi, T. Unification of software reliability models using Markovian arrival processes. In Proceedings of the 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC-2011), Pasadena, CA, USA, 12–14 December 2011; IEEE CS Press: Los Alamitos, CA, USA, 2011; pp. 20–27.

44. Goel, A.L.; Okumoto, K. *An Imperfect Debugging Model for Reliability and Other Quantitative Measures of Software Systems*; Technical Report 78-1; Department of IE & OR, Syracuse University: Syracuse, NY, USA, 1978.
45. Littlewood, B. A Reliability Model for Systems with Markov Structure. *Appl. Stat.* **1975**, *24*, 172–177. [[CrossRef](#)]
46. Ledoux, J. Availability modeling of modular software. *IEEE Trans. Reliab.* **1999**, *48*, 159–168. [[CrossRef](#)]
47. Zeephongsekul, P.; Jayasinghe, C.L.; Fiondella, L.; Nagaraju, V. Maximum-Likelihood Estimation of Parameters of NHPP Software Reliability Models Using Expectation Conditional Maximization Algorithm. *IEEE Trans. Reliab.* **2016**, *65*, 1571–1583. [[CrossRef](#)]
48. Nagaraju, V.; Fiondella, L.; Zeephongsekul, P.; Jayasinghe, C.L.; Wandji, T. Performance Optimized Expectation Conditional Maximization Algorithms for Nonhomogeneous Poisson Process Software Reliability Models. *IEEE Trans. Reliab.* **2017**, *66*, 722–734. [[CrossRef](#)]
49. Okamura, H.; Dohi, T. Application of EM Algorithm to NHPP-Based Software Reliability Assessment with Ungrouped Failure Time Data. In *Stochastic Reliability and Maintenance Modeling: Essays in Honor of Professor Shunji Osaki on his 70th Birthday*; Dohi, T., Nakagawa, T., Eds.; Springer: London, UK, 2013; pp. 285–313. [[CrossRef](#)]
50. Littlewood, B. Rationale for a Modified Duane Model. *IEEE Trans. Reliab.* **1984**, *R-33*, 157–159. [[CrossRef](#)]
51. Goel, A.L. Software Reliability Models: Assumptions, Limitations and Applicability. *IEEE Trans. Softw. Eng.* **1985**, *SE-11*, 1411–1423. [[CrossRef](#)]
52. Gokhale, S.S.; Trivedi, K.S. Log-Logistic Software Reliability Growth Model. In Proceedings of the Third IEEE International High-Assurance Systems Engineering Symposium (HASE-1998), Washington, DC, USA, 13–14 November 1998; IEEE CS Press: Turku, Finland, 1998; pp. 34–41.
53. Musa, J. *Software Reliability Data*; Technical Report; Rome Air Development Center, Griffiss AFB: Rome, NY, USA, 1979.
54. Okamura, H.; Dohi, T. SRATS: Software reliability assessment tool on spreadsheet (Experience report). In Proceedings of the 24th International Symposium on Software Reliability Engineering (ISSRE 2013), Pasadena, CA, USA, 4–7 November 2013; IEEE CS Press: Los Alamitos, CA, USA, 2013; pp. 100–117.