Vietnam J Comput Sci (2014) 1:39–45
DOI 10.1007/s40595-013-0003-4

REGULAR PAPER

Loglog fault-detection rate and testing coverage software reliability models subject to random environments

Hoang Pham

Received: 1 October 2013 / Accepted: 2 October 2013 / Published online: 9 November 2013

Abstract Many software reliability growth models (SRGMs) have developed in the past three decades to quantify several reliability measures including the expected number of remaining faults and software reliability. The underlying common assumption of many existing models is that the operating environment and the developing environment are the same. In reality, this is often not the case because the operating environments are unknown due to the uncertainty of environments in the field. In this paper, we present two new software reliability models with considerations of the fault-detection rate based on a Loglog distribution and the testing coverage subject to the uncertainty of operating environments. Examples are included to illustrate the goodness-of-fit test of proposed models and several existing non-homogeneous Poisson process (NHPP) models based on a set of failure data collected from software applications. Three goodness-of-fit test criteria, such as, mean square error, predictive-ratio risk, and predictive power, are used as an example to illustrate the model comparisons. The results show that the proposed models fit significantly better than other existing NHPP models based on the studied criteria. As we know different criteria have different impacts in measuring the software reliability and that no software reliability model is optimal for all contributing criteria. In this paper, we also discuss a method, called normalized criteria distance, to show ways to rank and select the best model from among SRGMs based on a set of criteria taken all together. Examples show that the proposed method offers a promising technique for selecting the best model based on a set of contributing criteria.

H. Pham (✉)
Department of Industrial and Systems Engineering,
Rutgers University, Piscataway, NJ 08854, USA
e-mail: hopham@rci.rutgers.edu

Keywords Non-homogeneous Poisson process (NHPP) · Software reliability growth models · Loglog distribution · Normalized criteria distance · Operating environments

Abbreviations

$m(t)$	Expected number of software failures detected by time t, also known as mean value function
N	Number of faults that exist in the software before testing
$h(t)$	Time-dependent fault detection rate per unit of time

1 Introduction

Among all software reliability growth models (SRGMs), a large family of stochastic reliability models based on a non-homogeneous Poisson process (NHPP), known as NHPP reliability models, has been widely used to track reliability improvement during software testing. Many existing NHPP software reliability models [1–26] have been carried out through the fault intensity rate function and the mean value functions $m(t)$ within a controlled testing environment to estimate reliability metrics such as the number of residual faults, failure rate, and reliability of software. Generally, these models are applied to the software testing data and then used to make predictions on the software failures and reliability in the field. In other words, the underlying common assumption of such models is that the operating environments and the developing environment are about the same. The operating environments in the field for the software, in reality, are quite different. The randomness of the operating environments will affect the software failure and software reliability in an unpredictable way.

Estimating software reliability in the field is important, yet a difficult task. Usually, software reliability models are applied to system test data with the hope of estimating the failure rate of the software in user environments. Teng and Pham [3] have discussed a generalized model that captures the uncertainty of the environments and its effects upon the software failure rate. Other researchers [8,19–21,24,27] have also developed reliability and cost models incorporating both testing phase and operating phase in the software development cycle for estimating the reliability of software systems in the field. Software development is a very complex process and there are still issues that have not yet been addressed. Testing coverage is one of these issues. Testing coverage [27] is a measure that enables software developers to evaluate the quality of the tested software and determine how much additional effort is needed to improve the reliability of the software. Testing coverage can provide customers with a quantitative confidence criterion when they plan to buy or use the software products.

In this paper, we present two new software reliability models. The first model is, called Loglog fault-detection rate, an NHPP model where the fault-detection rate is based on a loglog distribution function. The second is, called testing coverage model with uncertainty environments, also an NHPP with considerations of the uncertainty of operating environments where the testing coverage function follows the Loglog distribution. The explicit solution of the mean value functions for these new models are derived in Sect. 2. Criteria for model comparisons and a new method called normalized criteria distance (NCD), for selecting the best model is discussed in Sect. 3. Model analysis and results are discussed in Sect. 4 to illustrate the goodness-of-fit criteria of proposed models and compare them with several existing NHPP models based on three common criteria such as mean square error, predictive-ratio risk, and predictive power from a set of software failure data. Section 5 concludes the paper with remarks.

2 Software reliability modeling

2.1 An NHPP loglog fault-detection rate model

Many existing NHPP models assume that failure intensity is proportional to the residual fault content. A general NHPP mean value function $m(t)$ with time-dependent fault detection rate is given by [2]:

$$m(t) = N\left[1 - \mathrm{e}^{-\int_0^t h(x)\mathrm{d}x}\right] \tag{1}$$

In this paper, we consider that the software fault-detection rate per unit of time, $h(t)$, has a Vtub-shaped based on a loglog distribution function and is given by [2]:

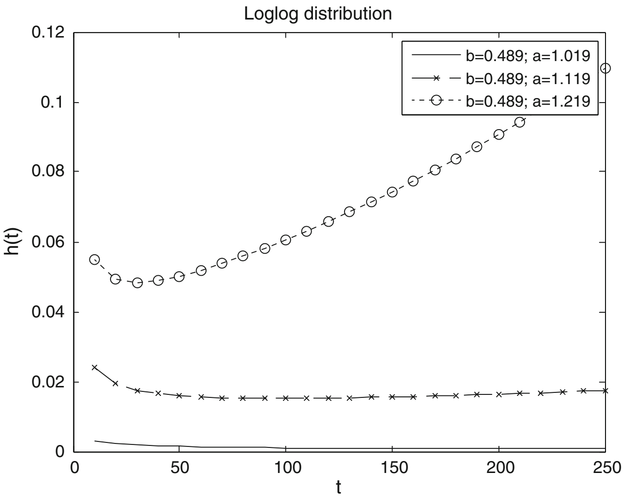

Fig. 1 Fault-detection rate function $h(t)$ for various values of a and $b = 0.489$

$$h(t) = b\ln(a)t^{b-1}a^{t^b} \qquad \text{for } a > 1,\ b > 0 \tag{2}$$

It should be noted that the loglog distribution has a unique Vtub-shaped curve while the Weibull distribution has a bathtub-shaped curve. They, however, are not the same. As for the Vtub-shaped from the Loglog distribution, after the infant mortality period, the system starts to experience at a relatively low increasing rate, but not at a constant rate, and then increases with failures due to aging. For the bathtub-shaped, after the infant mortality period, the useful life of the system begins. During its useful life, the system fails as a constant rate. This period is then followed by a wear out period during which the system starts slowly and increases with the onset of wear out. Figure 1 describes the Vtub-shaped function $h(t)$ for various values of parameter a where $b = 0.489$.

From Eq. (2), we can obtain the expected number of software failures detected by time t using Eq. (1):

$$m(t) = N\left(1 - \mathrm{e}^{-\left(a^{t^b}-1\right)}\right) \tag{3}$$

2.2 An NHPP testing coverage model with random environments

Testing coverage is important information for both software developers and customers of software products. Such information can be used by managers in order to determine how much additional effort is needed to improve the quality of the software products.

A generalized mean value function $m(t)$ based on the testing coverage function subject to the uncertainty of operating environments can be obtained by solving the following defined differential equation:

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t} = \eta\frac{\frac{\partial c(t)}{\partial t}}{(1-c(t))}[N - m(t)] \tag{4}$$

where $c(t)$ represents the testing coverage and η is a random variable that represents the uncertainty of system detection rate in the operating environments with a probability density function g. The closed-form solution for function $m(t)$ in term of random variable η with an initial condition $m(0) = 0$ is given by:

$$m(t) = N\left[1 - \mathrm{e}^{-\eta\int_0^t \frac{c'(x)}{(1-c(x))}\mathrm{d}x}\right] \tag{5}$$

If we assume that the random variable η has a gamma distribution with parameters α and β where the pdf of η is given by

$$g(x) = \frac{\beta^\alpha x^{\alpha-1}\mathrm{e}^{-\beta x}}{\Gamma(\alpha)} \qquad \text{for } \alpha,\ \beta > 0;\ x \geq 0 \tag{6}$$

then from Eq. (5), we can obtain [21]:

$$m(t) = N\left[1 - \left(\frac{\beta}{\beta + \int_0^t \frac{c'(s)}{1-c(s)}\mathrm{d}s}\right)^\alpha\right] \tag{7}$$

In this paper, we assume that the testing coverage function has a loglog distribution [2] as follows:

$$c(t) = 1 - \mathrm{e}^{1-a^{t^b}} \qquad \text{for } a > 1,\ b > 0 \tag{8}$$

Figures 2 and 3 describe the testing coverage function $c(t)$ and testing coverage rate $c'(t)$ for various values of parameter a where $b = 0.196$. We observe that for a given value b, as parameter a increases the testing coverage function increases but the testing coverage rate decreases.

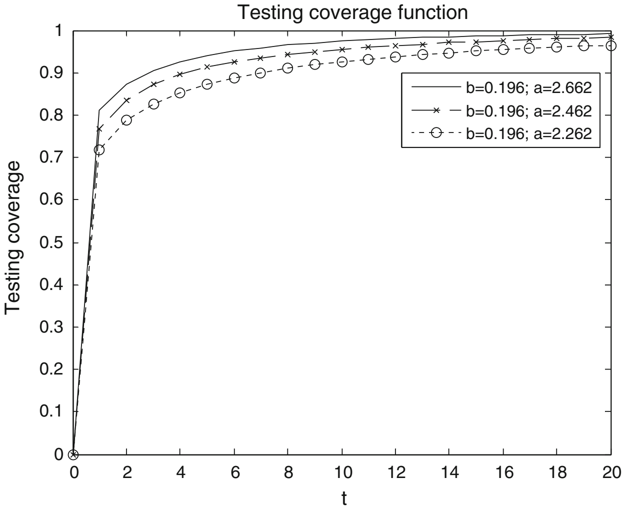

Fig. 2 Testing coverage function $c(t)$ for various values of a and $b = 0.196$

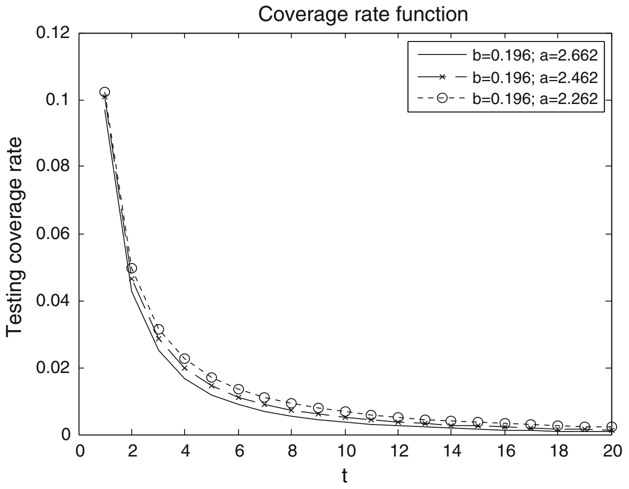

Fig. 3 Testing coverage rate function $c'(t)$ for various values of a and $b = 0.196$

Substitute the function $c(t)$ into Eq. (7), we can easily obtain the expected number of software failures detected by time t with random environments:

$$m(t) = N\left(1 - \left(\frac{\beta}{\beta + a^{t^b} - 1}\right)^\alpha\right) \tag{9}$$

Table 1 summarizes the two proposed models and several existing well-known NHPP models with different mean value functions.

3 Normalized criteria distance method

Once the analytical expression for the mean value function $m(t)$ is derived, the model parameters to be estimated in the mean value function can be obtained with a help of developed Matlab programs that based on the least square estimate (LSE) method.

There are more than a dozen of existing goodness-of-fit test criteria. Obviously different criteria have different impact in measuring the software reliability due to the selection among the existing models and, however, that no software reliability model is optimal for all contributing criteria. This makes the job of developers and practitioners much more difficult when they need to select an appropriate model, if not the best, to use from among existing SRGMs for any given application based on a set of criteria.

In this section, we discuss a new method called, NCD, for ranking and selecting the best model from among SRGMs based on a set of criteria taken all together with considerations of criteria weight w_1, w_2,..., w_d. Let s denotes the number of software reliability models with d criteria, and C_{ij} represents the criteria value of ith model of jth criteria where $i = 1, 2, \ldots, s$ and $j = 1, 2, \ldots, d$.

The NCD value, D_k, measures the distance of the normalized criteria from the origin for kth model and can be defined as follows [21]:

$$D_k = \sqrt{\left(\sum_{j=1}^{d}\left(\left(\frac{C_{kj}}{\sum_{i=1}^{s} C_{ij}}\right)^2 w_j\right)\right)} \qquad k = 1, 2, \ldots, s \tag{10}$$

where s and d are the total number of models and total number of criteria, respectively, and w_j denotes the weight of the criterion j for $j = 1, 2, \ldots, d$.

Thus, the smaller the NCD value, D_k, it represents the better rank as compare to higher NCD value. In Sect. 4, we use three common criteria such as the mean square error, the predictive-ratio risk, and the predictive power, to illustrate the proposed NCD method.

4 Model analysis and results

4.1 Some existing criteria

As mentioned in Sect. 3, there are more than a dozen of existing goodness-of-fit criteria. In this study, we discuss briefly three common criteria in this section and use them to compare those models as listed in Table 1. They are: the mean square error, the predictive-ratio risk, and the predictive power.

The mean square error (MSE) measures the deviation between the predicted values with the actual observation and is defined as:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (\hat{m}(t_i) - y_i)^2}{n - k} \tag{11}$$

where n and k are the number of observations and number of parameters in the model, respectively.

The predictive-ratio risk (PRR) measures the distance of model estimates from the actual data against the model estimate, and is defined as [17]:

$$\mathrm{PRR} = \sum_{i=1}^{n} \left(\frac{\hat{m}(t_i) - y_i}{\hat{m}(t_i)}\right)^2 \tag{12}$$

where y_i is total number of failures observed at time t_i according to the actual data and $\hat{m}(t_i)$ is the estimated cumulative number of failures at time t_i for $i = 1, 2, \ldots, n$.

The predictive power (PP) measures the distance of model estimates from the actual data against the actual data, is as follows:

$$\mathrm{PP} = \sum_{i=1}^{n} \left(\frac{\hat{m}(t_i) - y_i}{y_i}\right)^2 \tag{13}$$

For all these three criteria—MSE, PRR, and PP—the smaller the value, the better the model fits, relative to other models run on the same data set.

4.2 Software failure data

A set of system test data was provided in [2, p. 149] which is referred to as Phase 2 data set and is given in Table 2. In this data set the number of faults detected in each week of testing is found and the cumulative number of faults since the start of testing is recorded for each week. This data set provides the cumulative number of faults by each week up to 21 weeks. We perform the calculations for LSE estimates and other measures using Matlab programs.

Table 1 Software reliability models

Model	$m(t)$
Goel–Okumoto (G–O)	$m(t) = a(1 - \mathrm{e}^{-bt})$
Delayed S-shaped	$m(t) = a(1 - (1 + bt)\mathrm{e}^{-bt})$
Inflection S-shaped	$m(t) = \frac{a(1-\mathrm{e}^{-bt})}{1+\beta \mathrm{e}^{-bt}}$
Yamada imperfect debugging 1	$m(t) = a[1 - \mathrm{e}^{-bt}]\left[1 - \frac{\alpha}{b}\right] + \alpha a t$
PNZ model	$m(t) = \frac{a}{1+\beta \mathrm{e}^{-bt}}\left([1 - \mathrm{e}^{-bt}]\left[1 - \frac{\alpha}{b}\right] + \alpha t\right)$
Pham–Zhang model	$m(t) = \frac{1}{1+\beta \mathrm{e}^{-bt}}\left((c + a)(1 - \mathrm{e}^{-bt}) - \frac{a}{b-\alpha}(\mathrm{e}^{-\alpha t} - \mathrm{e}^{-bt})\right)$
Dependent-parameter model	$m(t) = \alpha(1 + \gamma t)(\gamma t + \mathrm{e}^{-\gamma t} - 1)$
Dependent-parameter model with $m(t_0) \neq 0$	$m(t) = m_0\left(\frac{\gamma t+1}{\gamma t_0+1}\right)\mathrm{e}^{-\gamma(t-t_0)}$ $+\alpha(\gamma t + 1)[\gamma t - 1 + (1 - \gamma t_0)\mathrm{e}^{-\gamma(t-t_0)}]$
Loglog fault-detection rate model	$m(t) = N\left(1 - \mathrm{e}^{-(a^{t^b}-1)}\right)$
Testing coverage model with uncertainty	$c(t) = 1 - \mathrm{e}^{1-a^{t^b}}$ $m(t) = N\left(1 - \left(\frac{\beta}{\beta+a^{t^b}-1}\right)^{\alpha}\right)$

Table 2 Phase 2 system test data [2]

Week index	Exposure time (cumulative system test hours)	Fault	Cumulative fault
1	416	3	3
2	832	1	4
3	1,248	0	4
4	1,664	3	7
5	2,080	2	9
6	2,496	0	9
7	2,912	1	10
8	3,328	3	13
9	3,744	4	17
10	4,160	2	19
11	4,576	4	23
12	4,992	2	25
13	5,408	5	30
14	5,824	2	32
15	6,240	4	36
16	6,656	1	37
17	7,072	2	39
18	7,488	0	39
19	7,904	0	39
20	8,320	3	42
21	8,736	1	43

4.3 Model results and comparison

Table 3 summarizes the results of the estimated parameters for all ten models as shown in Table 1 using the least square estimation (LSE) method and its criteria (MSE, PRR, and PP) values. The coordinates X, Y and Z in Fig. 4 illustrate the MSE, PRR, and PP criteria values, respectively, of the models. From Table 3, we observe that model 10 has the smallest MSE value, while model 9 has the smallest PRR value, and model 8 has the smallest PP value.

It is worthwhile noting that although both the PRR and PP values for the proposed testing coverage model with uncertainty (model 10) are slightly larger than the dependent parameter model (model 8), the MSE value for model 10 is significantly smaller than the dependent parameter model 8. Similarly, to compare all the models based on the PRR criterion, we find that the proposed loglog fault-detection rate (model 9) provides the best fit with the smallest PRR value.

As we can see from Table 3, the selection of the best model will then depend upon the modeling criteria. We now illustrate the proposed NCD method (in Sect. 3) to obtain the ranking results of all the ten models from Table 3 based on all three goodness-of-fit criteria such as MSE, PRR, and PP.

Table 3 Model parameter estimation and comparison criteria

Model name	LSEs	MSE	PRR	PP
G–O model (model 1)	$\hat{a} = 98{,}295$ $\hat{b} = 5.2 \times 10^{-8}$	6.61	0.69	1.10
Delayed S-shaped (model 2)	$\hat{a} = 62.3$ $\hat{b} = 2.85 \times 10^{-4}$	3.27	44.27	1.43
Inflection S-shaped (model 3)	$\hat{a} = 46.6$ $\hat{b} = 5.78 \times 10^{-4}$ $\hat{\beta} = 12.20$	1.87	5.94	0.90
Yamada imperfect debugging (model 4)	$\hat{a} = 1.5$ $\hat{b} = 1.1 \times 10^{-3}$ $\hat{\alpha} = 3.8 \times 10^{-3}$	4.98	4.30	0.81
PNZ model (model 5)	$\hat{a} = 45.99$ $\hat{b} = 6.0 \times 10^{-4}$ $\hat{\alpha} = 0$ $\hat{\beta} = 13.24$	1.99	6.83	0.96
Pham–Zhang model (model 6)	$\hat{a} = 0.06$ $\hat{b} = 6.0 \times 10^{-4}$ $\hat{\alpha} = 1.0 \times 10^{-4}$ $\hat{\beta} = 13.2$ $\hat{c} = 45.9$	2.12	6.79	0.95
Dependent parameter model (model 7)	$\hat{\alpha} = 3.0 \times 10^{-6}$ $\hat{\gamma} = 0.49$	43.69	601.34	4.53
Dependent parameter model with $m(t_0) \neq 0,\ t_0 \neq 0$ (model 8)	$\hat{\alpha} = 890{,}996$ $\hat{\gamma} = 1.2{\times}10^{-6}$ $t_0 = 832$ $m_0 = 4$	24.79	1.14	0.73
Loglog fault-detection rate model (model 9)	$\hat{N} = 231.92$ $\hat{a} = 1.019$ $\hat{b} = 0.489$	7.03	0.05	1.21
Testing coverage model with uncertainty (model 10)	$\hat{N} = 43.25$ $\hat{a} = 2.662$ $\hat{b} = 0.196$ $\hat{\alpha} = 4.040$ $\hat{\beta} = 35.090$	1.80	2.06	0.77

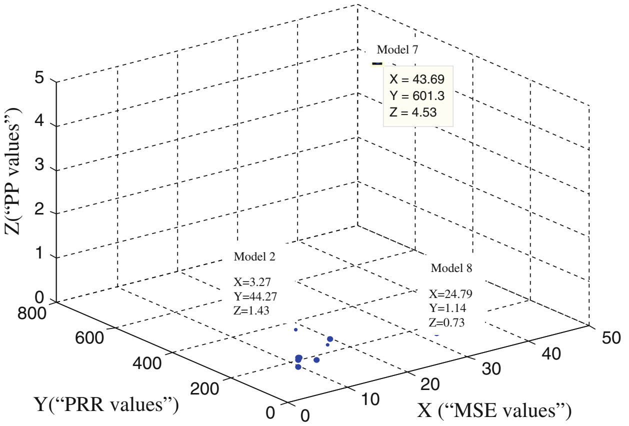

Fig. 4 A three-dimension plot (X, Y, Z) represents (MSE, PRR, PP) values when $w_1 = 0.3$, $w_2 = 100$, $w_3 = 0.1$

The modeling comparison and results for the case when all the criteria weight are the same (i.e., $w_1 = w_2 = w_3 = 1$) and when all are not the same ($w_1 = 0.3$, $w_2 = 100$, $w_3 = 0.1$) are presented in Tables 4 and 5, respectively. In other words, using Eq. (10) and the criteria values and results given in

Table 3, we obtain the NCD values and their corresponding ranking as shown in Table 4 for all $w_j = 1$ for $j = 1, 2,$ and 3. Table 5 shows the NCDs and their corresponding ranking when $w_1 = 0.3$, $w_2 = 100$, and $w_3 = 0.1$. In Fig. 4, the coordinates *X, Y* and *Z* represent the corresponding of the MSE, PRR, and PP values of each model for criteria weight $w_1 = 0.3$, $w_2 = 100$, and $w_3 = 0.1$. The delayed s-shaped (model 2) for example, $X = 3.27$, $Y = 44.27$, and $Z = 1.43$, indicates the MSE, PRR, and PP values of model 2. Figure 5 illustrates the model ranking based on the NCD values given in Table 5 for criteria weight $w_1 = 0.3$, $w_2 = 100$, and $w_3 = 0.1$. For example, a set of coordinates ($X = 10$, $Y = 1$, and $Z = 0.03698$) indicates that (shown in Table 5) model 10 is ranked the best (1st) where the NCD value is 0.03698.

Based on this study we can draw a conclusion that the proposed testing coverage model (model 10) and the loglog fault detection rate (model 9) can provide the best fit based on the MSE and PRR criteria, respectively. The NCD method in general is a simple and useful tool for modeling selection. Obviously, further work in broader validation of this conclusion is needed using other data sets as well as other comparison criteria.

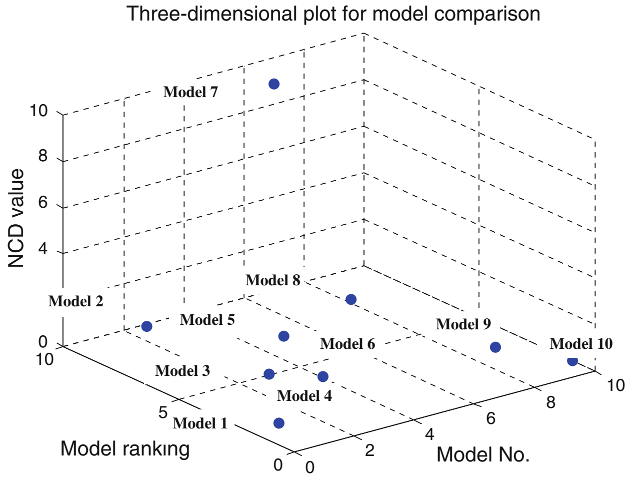

Fig. 5 A three-dimension plot with the model ranking and NCD values for $w_1 = 0.3$, $w_2 = 100$, $w_3 = 0.1$

5 Conclusion

We present two new software reliability models by considering a loglog fault-detection rate function and the testing

Table 4 Parameter estimation and model comparison when $w_j = 1$ for $j = 1, 2, 3$

Model/criteria	MSE (rank)	PRR (rank)	PP (rank)	NCD value (D_k)	Model rank
1. G–O model	6.61 (7)	0.69 (2)	1.10 (7)	0.106232216	6
2. Delayed S-shaped	3.27 (5)	44.27 (9)	1.43 (9)	0.129758016	8
3. Inflection S-shaped	1.87 (2)	5.94 (6)	0.90 (4)	0.070417115	2
4. Yamada imperfect debugging model	4.98 (6)	4.30 (5)	0.81 (3)	0.079212229	5
5. PNZ model	1.99 (3)	6.83 (8)	0.96 (6)	0.075194169	4
6. Pham–Zhang model	2.12 (4)	6.79 (7)	0.95 (5)	0.074845806	3
7. Dependent-parameter model	43.69 (10)	601.34 (10)	4.53 (10)	1.053569725	10
8. Dependent-parameter model with $m(t_0) \neq 0,\ t_0 \neq 0$	24.79 (9)	1.14 (3)	0.73 (1)	0.258395095	9
9. Loglog fault-detection rate model	7.03 (8)	0.05 (1)	1.21 (8)	0.115308973	7
10. Testing coverage model with uncertainty	1.80 (1)	2.06 (4)	0.77 (2)	0.060436587	1

Table 5 Parameter estimation and model comparison when $w_1 = 0.3$, $w_2 = 100$, $w_3 = 0.1$

Model/criteria	MSE (rank)	PRR (rank)	PP (rank)	NCD value (D_k)	Model rank
1. G–O model	6.61 (7)	0.69 (2)	1.10 (7)	0.046265595	2
2. Delayed S-shaped	3.27 (5)	44.27 (9)	1.43 (9)	0.65852017	9
3. Inflection S-shaped	1.87 (2)	5.94 (6)	0.90 (4)	0.091330651	5
4. Yamada imperfect debugging model	4.98 (6)	4.30 (5)	0.81 (3)	0.072219186	4
5. PNZ model	1.99 (3)	6.83 (8)	0.96 (6)	0.104518861	7
6. Pham–Zhang model	2.12 (4)	6.79 (7)	0.95 (5)	0.103971347	6
7. Dependent-parameter model	43.69 (10)	601.34 (10)	4.53 (10)	8.933743693	10
8. Dependent-parameter model with $m(t_0) \neq 0,\ t_0 \neq 0$	24.79 (9)	1.14 (3)	0.73 (1)	0.140433906	8
9. Loglog fault-detection rate model	7.03 (8)	0.05 (1)	1.21 (8)	0.04854067	3
10. Testing coverage model with uncertainty	1.80 (1)	2.06 (4)	0.77 (2)	0.036977987	1

coverage subject to the uncertainty of the operating environments. The explicit mean value function solutions for the proposed models are presented. The results of the estimated parameters of proposed models and other NHPP models and their MSE, PRR, and PP are also discussed. We also discuss an NCD method for obtaining the model ranking and selecting the best model from among SRGMs based on a set of criteria taken all together. Example results show that the presented new models can provide the best fit based on the NCD method as well as some studied criteria. Obviously, further work in broader validation of this conclusion is needed using other data sets as well as considering other comparison criteria.

Acknowledgments This paper was made possible by the support of NPRP 4-631-2-233 grant from Qatar National Research Fund (QNRF). The statements made herein are solely the responsibility of the author.

References

1. Goel, A.L., Okumoto, K.: Time-dependent fault-detection rate model for software and other performance measures. IEEE Trans. Reliab. **28**, 206–211 (1979)
2. Pham, H.: System Software Reliability. Springer, London (2006)
3. Teng, X., Pham, H.: A new methodology for predicting software reliability in the random field environments. IEEE Trans. Reliab. **55**(3), 458–468 (2006)
4. Ohba, M.: Inflexion S-shaped software reliability growth models. In: Osaki, s., Hatoyama, Y. (eds.) Stochastic Models in Reliability Theory, pp. 144–162. Springer-Verlag, Berlin, Germany (1984)
5. Pham, H.: Software reliability assessment: imperfect debugging and multiple failure types in software development. In: EG&G-RAAM-10737. Idaho National Engineering Laboratory (1993)
6. Pham, H.: A software cost model with imperfect debugging, random life cycle and penalty cost. Int. J. Syst. Sci. **27**(5), 455–463 (1996)
7. Ohba, M., Yamada, S.: S-shaped software reliability growth models. In: Proceeding of the 4th international conference on reliability and maintainability, pp. 430–436 (1984)
8. Teng, X., Pham, H.: A software cost model for quantifying the gain with considerations of random field environments. IEEE Trans. Comput. **53**(3) (2004)
9. Zhang, X., Teng, X., Pham, H.: Considering fault removal efficiency in software reliability assessment. IEEE Trans. Syst. Man Cybern. Part A **33**(1), 114–120 (2003)
10. Pham, H., Zhang, X.: NHPP software reliability and cost models with testing coverage. Eur. J. Oper. Res. **145**, 443–454 (2003)
11. Pham, H., Nordmann, L., Zhang, X.: A general imperfect software debugging model with s-shaped fault detection rate. IEEE Trans. Reliab. **48**(2), 169–175 (1999)
12. Pham, H., Zhang, X.: An NHPP software reliability model and its comparison. Int. J. Reliab. Qual. Saf. Eng. **4**(3), 269–282 (1997)
13. Pham, L., Pham, H.: Software reliability models with time-dependent hazard function based on Bayesian approach. IEEE Trans. Syst. Man Cybern. Part A **30**(1), 25–35 (2000)
14. Yamada, S., Ohba, M., Osaki, S.: S-shaped reliability growth modeling for software fault detection. IEEE Trans. Reliab. **12**, 475–484 (1983)
15. Yamada, S., Osaki, S.: Software reliability growth modeling: models and applications. IEEE Trans. Softw. Eng. **11**, 1431–1437 (1985)
16. Yamada, S., Tokuno, K., Osaki, S.: Imperfect debugging models with fault introduction rate for software reliability assessment. Int. J. Syst. Sci. **23**(12) (1992)
17. Pham, H., Deng, C.: Predictive-ratio risk criterion for selecting software reliability models. In: Proceeding of the 9th international conference on reliability and quality in design (2003)
18. Pham, H.: An imperfect-debugging fault-detection dependent-parameter software. Int. J. Autom. Comput. **4**(4), 325–328 (2007)
19. Zhang, X., Pham, H.: Software field failure rate prediction before software deployment. J. Syst. Softw. **79**, 291–300 (2006)
20. Sgarbossa, F., Pham, H.: A cost analysis of systems subject to random field environments and reliability. IEEE Trans. Syst. Man Cybern. Part C **40**(4), 429–437 (2010)
21. Pham, H.: A software reliability model with vtub-shaped fault-detection rate subject to operating environments. In: Proceeding of the 19th ISSAT international conference on reliability and quality in design, Hawaii (2013)
22. Kapur, P.K., Pham, H., Aggarwal, A.G., Kaur, G.: Two dimensional multi-release software reliability modeling and optimal release planning. IEEE Trans. Reliab. **61**(3), 758–768 (2012)
23. Kapur, P.K., Pham, H., Anand, S., Yadav, K.: A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. IEEE Trans. Reliab. **60**(1), 331–340 (2011)
24. Persona, A., Pham, H., Sgarbossa, F.: Age replacement policy in random environment using systemability. Int. J. Syst. Sci. **41**(11), 1383–1397 (2010)
25. Xiao, X., Dohi, T.: Wavelet shrinkage estimation for non-homogeneous Poisson process based software reliability models. IEEE Trans. Reliab. **60**(1), 211–225 (2011)
26. Kapur, P.K., Pham, H., Chanda, U., Kumar, V.: Optimal allocation of testing effort during testing and debugging phases: a control theoretic approach. Int. J. Syst. Sci. **44**(9), 1639–1650 (2013)
27. Pham, H., Zhang, X.: NHPP software reliability and cost models with testing coverage. Eur. J. Oper. Res. **145**, 443–454 (2003)