SRGM with Imperfect Debugging by Genetic Algorithms

Dr.R.SatyaPrasad
Associate.Professor
Dept.of Computer science &Engineering
Acharya Nagarjuna University
Nagarjuna Nagar-522524
profrsp@gmail.com

O.NagaRaju
Asst.Professor.
Dept.of Computer science &Engineering
charya Nagarjuna University
Nagarjuna Nagar-522524
onrajunrt@gmail.com

Prof.R.R.LKantam
Professor
Dept.of Statistics
Acharya Nagarjuna University
Nagarjuna Nagar-522524
kantam_rrl@rediffmail.com

Abstract

Computer software has progressively turned out to be an essential component in modern technologies. Penalty costs resulting from software failures are often more considerable than software developing costs. Debugging decreases the error content but expands the software development costs. To improve the software quality, software reliability engineering plays an important role in many aspects throughout the software life cycle. In this paper, we incorporate both imperfect debugging and change-point problem into the software reliability growth model(SRGM) based on the well-known exponential distribution the parameter estimation is studied and the proposed model is compared with the some existing models in the literature and is find to be better.

Key words: - Software Reliability, NHPP, Mean value function, Genetic Algorithms

1. Introduction:

We are witnessing our increasing dependence on software systems, as they are becoming more and more complex, thus harder to develop and maintain. Software systems are present in many safety – critical applications such as power plants, health care systems, air-traffic, etc. they all require high quality, reliabilities and safety.

Software reliability is the probability that software will not cause the failure of a product for a specified period of time. This probability is a function of the inputs, as well as a function of the existence of faults in the software.

Various NHPP SRGMs have been studied with various assumptions. Many of the SRGMs assume that each time a failure occurs, the fault that caused it can be immediately removed and no new faults are introduced, which is usually called perfect debugging, Imperfect debugging models have been proposed, with relaxation of the above assumption (hoba, 1984, pham 1993). The other assumption of many NHPP SRGMs is that each failure occurs independently and randomly in time according to the same distribution during the fault detection process (Musa at. Al 1987). However in more realistic situations, the failure distribution can be affected by many factors, such as the running environment, testing strategy and resource allocation. Once these factors are changed during the software testing phase, this could result in a software failure intensity function that increase or decrease non-monotonically. It is identified as a change point problem (Zhao, 1993). In software reliability estimation the change point effect should be considered simultaneously, if there is a change point exists, otherwise the estimators of the model cannot express the factual software reliability behavior.

2. A General NHPP Model:

Let $\{N(t), t \geq 0\}$ be a counting process representing the cumulative number of software failures by time t. The N (t) process is shown to be a NHPP with a mean value function m(t). Mean value function represent the s-expected number of software failures by time t. Goel and Okumoto (1979) assume that the number of software failures during non-overlapped time intervals is s-independent and the software failure intensity $\lambda(t)$ is proportional to the residual fault content. Thus m(t) can be obtained by solving the following differential equation.

$$\lambda(t) = \frac{dm(t)}{dt} = b\left(a - m(t)\right) \text{ .... } (2.1)$$

where 'a' denotes the initial number of faults contained in a program and b represents the fault detection rate. The solution of

$$m(t) = a\left(1 - e^{-bt}\right) \text{........................ } (2.2)$$

In software reliability, the initial number of faults and the fault detection rate are always unknown. The maximum likelihood technique can be used to evaluate the unknown parameters.

The conditional software reliability, R(x/t), is defined as the probability that there is no failure observed in the time period (t, t+x), given that the last failure occurred at a time point $t\left(t \geq 0, x > 0\right)$. Given the mean value function m(t), the conditional software reliability can be shown as

$$R\left(x/t\right) = e^{-\left[m(t+x)-m(t)\right]} = \exp\left[1 - a\left(e^{-bt} - e^{-b(1+x)}\right)\right] \text{ .... } (2.3)$$

For a more general NHPP SRGM, we can extend and modify Eq. (2.1) as following:

$$\lambda\left(t\right) = \frac{dm\left(t\right)}{dt} = b\left(t\right)\left[a\left(t\right) - m\left(t\right)\right], \text{ .... } (2.4)$$

where a(t) is the time-dependent fault content function which includes the initial and introduced faults in the program and b(t) is the time-dependent fault detection rate. One can define a(t), b(t) to yield more complex or less complex analytic solution for m(t). Various a(t), b(t) express different assumptions of the fault detection processes (Pham et al., 1999).

3. Imperfect-software-debugging models.

Following the general NHPP model, a constant a(t) implies the perfect debugging assumption, i.e., no new faults are introduced during the debugging process. Pham (1993) introduced an NHPP SRGM that is subject to imperfect debugging. He assumed if detected faults are removed, then there is a possibility to introduce new faults with a constant rate β. Let a(t) be the number of faults to be eventually detected (denoted by "a") plus the number of new faults introduced to the program by time t, the mean value function m(t) can be given as the solution of the following system of differential equations.

$$\frac{\partial m(t)}{\partial t} = b\left[a(t) - m(t)\right], \frac{\partial a(t)}{\partial t} = \beta \frac{\partial m(t)}{\partial t} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

$$a(0) = a, m(0) = 0$$

where a is the number of faults to be eventually detected. Solving the above equations, we can obtain the mean value function and conditional software reliability, respectively, as follows:

$$m(t) = \frac{a}{1-\beta}\left[1 - e^{-(1-\beta)bt}\right], \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(3.2)$$

$$R(x/t) = \exp\left(-m(x)e^{-(1-\beta)bt}\right)$$

4. An NHPP model with change-point.

Many SRGMs suppose the fault detection rate is a constant, or a monotonically increasing function. The failure intensity is expected to be a continuous function of time. For examples, Goel and Okumoto (1979) presented a G-O model subjected to a constant fault detection rate and Yamada et al. (1983) modified the G-O model and created an increasing fault detection rate function, which represents the debugging process with the learning phenomenon. Both of the models were proposed with continuous failure intensity function. As the earlier mention, the fault detection rate can be affected by many factors such as the testing strategy and resources allocation. During a software testing process, there is a possibility that the underlying fault detection rate function is changed at some time moment τ called 'change-point'. Considering the change- point problem in software reliability models is intended to be more close to the reality.

Zhao (1993) modified the Jelinski and Moranda (1972) model to estmate the allocation of change-point and the failure intensity function. He assumed that the observed inter-failure times follow the same distribution F at the beginning. After τ failure are observed. The remaining items have the distribution G. Distribution F and G are from the same parametric family.

Chang (1997) considered the change-point problems in the NHPP SRGMs. The parameters of the NHPP with change-point models are estimated by the weighted least square method. Let the parameter τ be the change point that is considered unknown and is to be estimated from the data. The fault detection rate function is defined as

$$b(t) = \begin{cases} b_1, & \text{when } 0 \le \text{t} \le \text{t}, \\ b_2, & \text{when t.} > \text{t}, \end{cases}$$

By the assumptions, the mean value function, m(t) and the intensity function, $\lambda(t)$, can be derived as

$$m(t) = \begin{cases} a\left(1 - e^{-b_1 \tau}\right), & \text{when } 0 \le \text{t} \le \tau, \\ a\left(1 - e^{-b_1 \tau - b_2 (t-\tau)}\right), & \text{when t.} > \tau, \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (4.1)$$

$$\lambda(t) = \frac{dm(t)}{dt} \begin{cases} ab_1 e^{-b_1 \tau}, & \text{when } 0 \le \text{t} \le \tau, \\ ab_2 e^{-b_1 \tau - b_2 (t-\tau)}, & \text{when t.} > \tau, \end{cases}$$

Introducing the above mean value function, the conditional software reliability function for any times x given t can be shown as

$$R(x/t) = \exp\left\{-\left(m(t+x) - m(t)\right)\right\}$$

$$= \begin{cases} \exp\left\{-a\left(e^{-b_1 t} - e^{-b_1 (t+x)}\right)\right\}, & \text{when t} \le \text{t+x} \le \tau, \\ \exp\left\{-a\left(e^{-b_1 t} - e^{-b_1 \tau - b_2 (t+x-\tau)}\right)\right\}, & \text{when t} \le \tau\text{+t} + x, \\ \exp\left\{-a\left(e^{-b_1 \tau - b_2 (t-\tau))} - e^{-b_1 \tau - b\ (1+x-\tau)}\right)\right\}, & \text{when } \tau < t \end{cases} \quad \ldots\ldots\ldots\ldots\ (4.2)$$

5. Imperfect-software-debugging model with change- point.

To consider the NHPP SRGM that integrates imperfect debugging with change-point problem, the following assumptions are made:

(a) When detected faults are removed at time t, it is possible to introduce new faults with introduction rate $\beta(t)$

$$\beta(t)=\begin{cases}\beta_1, \text{when } 0 \le \text{t} \le \tau, \\ \beta_2, \text{ when t.} > \tau\end{cases} (\tau \text{is the change-point})$$

(b) The fault detection rate represented as following is a step function.

$$b(t)=\begin{cases}b_1, \text{ when } 0 \le \text{t} \le \tau, \\ \text{b}_2, \text{ when t} > \tau,\end{cases}$$

(c) A NHPP models with the fault detection phenomenon in the software system.

In earlier studies, the parameter, τ is considered as unknown and is to be estimated from the collected failure data (Zhao, 1993; Hinkley, 1970; Chang 1997). Because the testing strategy and resource allocation can be tracked all the time during the fault detection process, it may be more reasonable to reconsider that the change point τ is given. Therefore, we can assume but not necessary the parameter τ as allocated in a certain time point and is known in advance. According to these assumptions, one can derive the new set of differential equations to obtain the new mean value function:

$$\frac{\partial m(t)}{\partial t}=b(t)\left(a(t)-m(t)\right), \frac{\partial a(t)}{\partial t}=\beta(t)\frac{\partial m(t)}{\partial t}$$

$$a(0)=a, m(0)=0$$

Solving the differential equations under the assumptions

(a) And (b) Yields

$$m(t)=\frac{a}{1-\beta}\left[1-e^{-(1-\beta)\left(b_1\tau+b_1(t-\tau)\right)}\right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

$$R(x/t)=\exp\left\{-\left(m(t+x)-m(t)\right)\right\}$$

$$= \begin{cases} \exp\left\{-\frac{a}{1-\beta_1}\left(e^{-(1-\beta_1)b_1 t} - e^{-(1-\beta_1)b_1(t+x)}\right)\right\}, \text{ when } 0<t \le \tau \\ \exp\left\{-\frac{a}{1-\beta_2}\left(e^{-(1-\beta_2)b_1\tau-(1-\beta_2)b_2(t-\tau)}\left(1-e^{-(1-\beta_2)b_2 x}\right)\right)\right\}, \text{ when } t > \tau \end{cases}$$

6. The Present model under study:

To consider the new model of NHPP SRGM that imperfect debugging with change point problem, the following assumption are.

a) When defected faults are removed at time t, it is possible to introduce new faults with introduction rate β.
b) The faults detection rate represented as a step function is given below $b(t) = b_1$, when $0 \le t$

The unknown parameters are to be estimated from the collected failure data (Zhao, 1993, Hinkley 1970, Chang 1997) According to these assumptions; one can derive the new set of differential equation to obtain the new mean value function.

$$\frac{\partial m(t)}{\partial t} = b(t)\left[a - m(t)\right]$$

$$\frac{\partial m(t)}{\partial t} = \beta(t)\frac{\partial m(t)}{\partial t}; a(0) = a, m(0) = 0$$

$$\Rightarrow b(t) = \beta m(t) + b_1 \text{ at } t = 0 b(t) = b_1$$

where b(t) is the time dependent faults detection rate running the general NHPP model, a constant b(t) implies the perfect debugging assumption i.e. no new faults are introduced during debugging process.

$$m(t) = \left\{\left[\frac{1-\frac{a\beta}{b_1}e^{(a\beta+b)t}}{1+\frac{a\beta}{b_1}e^{(a\beta+b)t}}\left(a + \frac{b_1}{\beta}\right)\right] + a - \frac{b_1}{\beta}\right\} \qquad \text{..................(6.1)}$$

Taking this mean value function we propose to suggest a new SRGM with the help of a NHPP. It is reliability Parameter, predictive validates. Its applicability as a SRGM can also be assessed through fitness of models.

The developed function will become more complex than the other models. Let the faults introduction rate is a constant (β) during the fault detection process, the mean value function time $0 \le t$

Eq. [m(t)] assumes that the intensity function $\lambda(t)$is not a continuous function of time except when b_1 Following the same definition of Goel and Okumto (1979), the conditional reliability function of this developed model can be obtained

$$R(x/t) = \exp\left\{-\left(m(t+x) - m(t)\right)\right\}$$

$$R(x/t) = \exp\left\{ \frac{\frac{a\beta}{b_1} e^{(a\beta+b_1)t} \left(e - 1^{(a\beta+b_1)x}\right) \left[\left(\frac{a+b_1}{\beta} + a - \frac{b_1}{\beta}\right)\right]}{\left(1 + \frac{a\beta}{b_1} e^{(a\beta+b_1)(t+x)}\right)\left(1 + \frac{a\beta}{b_1} e\left(a\beta + b_1\right)t\right)} \right\} \text{...... } \mathbf{(6.2)}$$

The R(x/t) model can be used to the construct the problem with single type of fault. However based on the severity that assesses the impact of the fault on the user, software faults can be classified into varies types.

The further modified model can be applied to conduct the software reliability estimation problem not only for the imperfect debugging and change point case but also the multiple fault type's problem. The only difficulty is that more parameters need to be estimated at the same time.

7. Numerical examples and model evaluation:

To verify the proposed model that incorporates both imperfect debugging and change-point problems, four data sets are introduced. Two of them are collected from real software development project and the others are obtained from simulation.

The first set of software failure data to be analyzed in this section is taken from Misra (1983. The purpose of the first example is to illustrate the process of model creation. In the data set software faults are classified into three different types. Critical (type1), major (type2) and minor (type 3). The total testing time and number of software failure for each week are recorded. Pham (1993) used the same data to illustrate the results of his imperfect debugging model based on the following setting of parameter. $\beta = 0.05$.

According to the assumption of NHPP the remaining parameters a, b_1 can be estimated by solving the following likelihood function, for its Maximum.

$$L(a,b_1) = \prod_{i=1}^{2}\prod_{j=1}^{n} \frac{\left[m_i(t_j) - m_i(t_j - 1)\right]^{y_i(t_j) - y_i(t_{j-1})} e^{-\left[m_i(t_j) - m_i(t_{j-1})\right]}}{\left(y_i(t_j) - y_i(t_{j-1})\right)!}$$

where $y_i(t_j)$ is the cumulated number of types I faults before time $y_i(t_j)$ and n is the interval domain size It should be point out that a similar analysis was conducted using the method of maximum likelihood discussed in (Pham, 1993). The estimates for parameters $a = .200$ $b_1 = 0.000355$

The results show that the fault detection rates increase for all three types of fault after time τ table 1 present the evaluation results of the developed model. Where mean square errors (MSE) measure weighted average of the square of the distance between actual data and the model estimates and is defined as

$$\text{MSE}=\sum_{i=1}^{3}\sum_{j=1}^{n}\left[y_i(t_j) - m_i(t_j)\right]^2 / d.$$

d: the degree of freedom.

For the purpose, of comparison MSE uses the degree of freedom by assigning a large penalty to a model, with more parameters. In this proposed model larger penalty is assigned. The smaller the MSE value, the better the model fits. It is exhibited that the new model yields a little more conservative results than the other NHPP SRGM but not significant, since the change-point is created and may not exist.

The proposed model using the data set collected from the system T1 in Musa (1979) are also examined.

Table-1

Comparison of goodness-of-fit of imperfect debugging NHPP models (data from Misra)

	Imperfect NHPP SRGMs		
	Our Model	**With change-point**	**No change-point**
Maxim Mize log-likelihood	-177.53	-153.998	-154.584
MSE (critical fault)	0.0734	0.082	0.086
MSE (major fault)	1.9210	1.998	2.028
MSE (minor fault)	4.1315	4.240	4.286

Table 2 Summarizes the MSE values for investigated models. It shows that the MSE-fit and MSE-predict values for the new model are smaller than the other one which do not consider change-point problem (24.50 vs 25.11 vs. 62.91 and 1.01vs 1.52vs 1.175 respectively)

Table-2
Comparison of descriptive and predictive power of imperfect debugging NHPP models (data from Mua)

	Imperfect NHPP SRGMs			
	Our Model	**With change-point**		**No change-point**
Maxim Mize log-likelihood	-199		-215	-223
MSE (fit	24.50		25.11	62.91
MSE (predict)	1.00		1.01	1.75

The results of using simulation data to verify the new models are shown in Table.3. In the data set 2, the descriptive error for the model without considering change-point is poor and unacceptable. Compared with the model with the assumption of a change-point, the new model gives very promising results where the MSE measures for description and prediction are 12.81 and 5.36 respectively.

Table-3

Comparison of descriptive and predictive power of imperfect debugging NHPP models (simulated data)

		Imperfect NHPP SRGMs		
		Our Model	**With change-point**	**No change-point**
Data Set -1	MSE (fit)	4.91	5.01	9.88
	MSE (predict)	1.01	1.21	1.63
Data Set -2	MSE (fit)	10.77	12.81	519.18
	MSE (predict)	4.00	5.36	9.45

8. Genetic Algorithms:

We apply a genetic algorithms based optimizer to slove the formulated mathematical models of the parameter estimation for software reliability. Genetic algorithms (GAs) have been used extensively for dealing with optimization problems. GAs is based on the biological evolution process, and was firstly introduced by Holland (1975) in the 1970s. GAs is useful (Goldberg 1989) where the search space is large, nonlinear and noisy, and solutions are ill-defined a priori.

The proposed model using the data set collection from the System T1 in Musa (1979) are also examined. These data sets include 136 faults found in the test phase. Table-4 summarizes the MSE (mean squared error) values investigated models. It shows that the MSE (fit) and MSE (predict) values for the model are smaller than the other one.

Table-4

	Imperfect NHPP SRGMs		
	Our Model	With change-point	No change-point
MSE (fit)	25.01	27.734	59.471
MSE (predict)	0.821	0.838	4.688

9. Conclusion:

The developed NHPP SRGM is unique in that it allows for analysis of software failure data with change point, imperfect debugging, and various fault types. From tables 1, 2, 3 and 4 we say that our proposed model is rated as better than the other consider models with respect to all the conditions are chosen. They for genetic algorithm are more suitable to our model with minimum disabances than comparative models.

10.References:

[1]. Chang, I.P., 1997. An analysis of software reliability with change-point Models. NSC 85-2121-M031-003, National Science Council, Taiwan.

[2].Fenton, N.E., Pfleeger, S.L., 1997. Software Metrics: A Rigorous and Practical Approach. PWS Publishing Company, Boston.

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

[4]. Goldberg, D.E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, (Addison-Wesley).

[5]. Hinkley, D.V., 1970. Inference about the change-point in a sequence of Random variables. Biometrika 57, 1–16.

[6]. Huan jyh shyur, 2003. A stochastic software reliability model with imperfect-debugging and change-point, Elsevier, the journal of systems and software.135-141.

[7]. Huan jyh shyur, Mu-chen chen,Analyzing software reliability Growth model with imperfect-debugging and change-point by Genetic Algrothms.

[8]. Jelinski, Z., Moranda, P.B., 1972. In: Software Reliability Research, Statistical Computer Performance Evaluation. Academic Press, New York, p. 465.

[9]. Misra, P.N., 1983. Software reliability analysis. IBM Syst. J. 22, 262–270.

[10]. Musa, J.D., 1979. Software reliability measures applied to system Engineering. In: Proceedings CONPCON.

[11]. Musa, J.D., Iannino, A., Okumoto, K., 1987. Software Reliability: Measurement Prediction Application. McGraw-Hill, New York.

[12]. Ohba, M., 1984. Software reliability analysis model. IBM J. Res. Develop. 28, 428–443.

[13]. Pham, H., 1993. Software reliability assessment: Imperfect debugging and multiple failure types in software development. EG&GRAMM-10737, Idaho National Engineering Laboratory.

[14]. Pham, H., Nordmann, L., Zhang, X., 1999. A general imperfectsoftware-Debugging model with s-shaped fault-detection rate. IEEE Trans. Reliab. 48, 169–175.

[15].Yamada, S., Ohba, M., Osaki, S., 1983. S-shaped reliability growth Modeling for software error detection. IEEE Trans. Reliab. 12, 475–484.

[16]. Zhao, M., 1993, Change-point problems in software and hardware reliability.Commun. Statist – Theory Meth., 22, 757-768

Authors

Dr. R. Satya Prasad Received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He have a satisfactory consistent academic track of record and received gold medal from Acharya Nagarjuna University for his out standing performance in a first rank in Masters Degree. He is currently working as Associative Professor and Head of the Department, in the Department of Computer Science & Engineering, Acharya Nagarjuna University. He has occupied various academic responsibilities like practical examiner, project adjudicator, external member of board of examiners for various Universities and Colleges in and around in Andhra Pradesh. His current research is focused on Software Engineering, Image Processing & Database Management System. He has published several papers in National & International Journals.

O.NagaRaju received the Masters Degree in Computer Science & Engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He is currently pursuing Ph.D., at Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. His research interests include Software Engineering, Network Computing, and Image Processing.