The Korean Communications in Statistics **Vol. 14** No. 3, 2007, pp. 497–506

A Class of Discrete Time Coverage Growth Functions for Software Reliability Engineering

Joong-Yang Park[1], Gyemin Lee[2] and Jae Heung Park[3]

Abstract

Coverage-based NHPP SRGMs have been introduced in order to incorporate the coverage growth behavior into the NHPP SRGMs. The coverage growth function representing the coverage growth behavior during testing is thus an essential factor of the coverage-based NHPP SRGMs. This paper proposes a class of discrete time coverage growth functions and illustrates its application to real data sets.

Keywords: Software reliability growth model; non-homogeneous Poisson process; coverage growth function.

1. Introduction

Recently software is becoming an integral part of computer systems. Since failures of a software system can cause severe consequences, reliability of a software system is a primary concern for both software developers and software users. Testing is a key activity for detecting and removing faults and improving reliability of a software system. In theory, it is impossible to detect and remove all the faults in the software system within a reasonable amount of testing time. Therefore developers usually determine when to stop testing and release the software based on the estimates of reliability measures. Many

1) Professor, Department of Information Statistics, RINS and RICI, Gyeongsang University, 900 Gazwa-dong, Jinju 660-701, Korea.
E-mail : joongyang@gnu.ac.kr

2) Associate Professor, Department of Information Statistics, RINS and RICI, Gyeongsang University, 900 Gazwa-dong, Jinju 660-701, Korea.
Correspondence : gyemin@gnu.ac.kr

3) Professor, Department of Computer Science, RINS and RICI, Gyeongsang University, 900 Gazwa-dong, Jinju 660-701, Korea.
E-mail : jhp@gnu.ac.kr

software reliability growth models (SRGMs) have been proposed and applied in practice to estimate software reliability measures.

One of the most popular SRGMs is the class of non-homogeneous Poisson process (NHPP) SRGMs. NHPP SRGMs postulate that the number of faults detected up to t testing time follows a Poisson distribution with mean value function (MVF) $m(t)$. An NHPP SRGM is therefore characterized by its MVF. Recently the coverage growth behavior during the testing has been integrated into the NHPP SRGMs. This approach is reasonable because the more is covered the software system, the more faults are likely to be detected. Such NHPP SRGMs are referred to as the coverage-based NHPP SRGMs. The coverage growth behavior is usually represented by the coverage growth function (CGF) $c(t)$. MVFs of well-known coverage-based NHPP SRGMs are as follows:

$$\frac{dm(t)}{dt} = a\,\frac{dc(t)}{dt}, \tag{1.1}$$

$$\frac{dm(t)}{dt} = a\,b(t)\,\frac{dc(t)}{dt}, \tag{1.2}$$

$$\frac{dm(t)}{dt} = [a(t) - m(t)]\,\frac{\frac{dc(t)}{dt}}{1 - c(t)}, \tag{1.3}$$

$$\frac{dm(t)}{dt} = b\,[a - m(t)]\,\frac{dc(t)}{dt}, \tag{1.4}$$

where a is the initial fault content, $a(t)$ is the fault content function, b is the fault detection rate, and $b(t)$ is the fault detection rate function. Eqs. (1.1)–(1.4) were respectively proposed by Piwowarski *et al.* (1993), Gokhale *et al.* (1996), Pham and Zhang(2003), and Yamamoto *et al.* (2004). If we insert different CGFs into one of Eqs. (1.1)–(1.4), different MVFs and consequently different NHPP SRGMs will be obtained. Eqs. (1.1)–(1.4) can be thus regarded as frameworks for generating NHPP SRGMs. Other types of coverage-based NHPP SRGMs can be found in Yamada and Fujiwara(2001), Fujiwara and Yamada(2002) and Park *et al.* (2005a, 2005b).

Eqs. (1.1)–(1.4) indicate that important software reliability metrics such as the number of detected faults, the number of remaining faults and the probability that the software operates without failing for a specified time interval are directly related to the coverage. Performance of coverage-based NHPP SRGMs derived from Eqs. (1.1)–(1.4) depends on how closely the corresponding CGFs represent the actual coverage growth behavior. In order for a coverage-based NHPP SRGM to be widely applicable, its CGF should be able to represent the coverage growth produced by arbitrary testing profile. Otherwise, its application will be limited. It is therefore necessary to develop CGFs with good representational ability.

This paper first reviews the previous studies on the CGF in Section 2. A class of discrete time CGFs is proposed in Section 3. Section 4 deals with maximum likelihood estimation of the parameters of the CGFs in the proposed class. A specific CGF should be selected from the class for practical application. Section 5 suggests the CGF based on the beta distribution and examines its usability by applying to the real data sets. Finally, concluding remarks are presented in Section 6.

2. Review on the CGFs

We begin this section by giving the formal definition of CGF. Let M be the set of constructs, which are basic building elements of a software system under testing. That is, M itself is the software system under testing. Testing is performed by executing test cases randomly selected from the input space according to the specified testing profile. The set of constructs executed up to t testing time is denoted by $M(t)$. One plausible metric for measuring the thoroughness of the testing and/or the progress of the testing is the coverage defined as $C(t) = |M(t)||M|^{-1}$, where $|\cdot|$ is the cardinality of a set. The CGF is defined as the expected value of $C(t)$, *i.e.*, $c(t) = E[C(t)]$.

The currently available CGFs are summarized in Table 2.1. Park *et al.* (2005a, 2005b) and Park and Fujiwara(2006) empirically compare the CGFs in Table 2.1 and remark that $c_3(t)$ and $c_{10}(t)$ are significantly better than others and compete with each other. However, these comparative studies ignore the stochastic attributes of $C(t)$. This is mainly because all the $c_i(t)$'s in Table 2.1 except $c_6(t)$ and $c_{10}(t)$ have been proposed without studying the underlying stochastic process $C(t)$ or $|M(t)|$. Grottke(2002) derived $c_6(t)$ by considering the coverage growth process as a Markov model, whereas Gokhale and Mullen(2004, 2005) derived $c_{10}(t)$ under the assumption that the execution rate of a construct follows a lognormal distribution. Even though $c_3(t)$ and $c_{10}(t)$ were shown to be well-performing CGFs, $c_3(t)$ lacks the theoretical foundation and $c_{10}(t)$ is not supported by some real data sets. Therefore, it is necessary to further investigate the coverage

Table 2.1: Available CGFs

CGF	references
$c_1(t) = 1 - \exp(-\beta t)$	Gokhale *et al.*(1996), Piwowarski *et al.* (1993)
$c_2(t) = \gamma \ln(1 + \beta t)$	Malaiya *et al.* (2002)
$c_3(t) = 1 - \exp(-\beta t^\gamma)$	Gokhale *et al.* (1996)
$c_4(t) = 1 - (1 + \beta t)\exp(-\beta t)$	Gokhale *et al.* (1996)
$c_5(5) = \frac{(\beta t)^\gamma}{1+(\beta t)^\gamma}$	Gohale and Trivedi(1999)
$c_6(t) = 1 - (1 - \beta\gamma t)^{1/\gamma}$	Grottke(2002)
$c_7(t) = \alpha_0 - \alpha_1 \exp(-\beta t)$	Sedigh-Ali *et al.* (2002)
$c_8(t) = \alpha_0 - \frac{\alpha_1 \exp(\beta t)}{1+\exp(-2\beta t)}$	Sedigh-Ali *et al.* (2002)
$c_9(t) = \frac{1-\exp(-\beta t)}{1+\gamma \exp(-\beta t)}$	Yamamoto *et al.* (2004)
$c_{10}(t) = 1 - \int_0^\infty e^{-\beta t} \frac{e^{-(\ln \beta - \mu)^2/2\sigma^2}}{\beta\sigma\sqrt{2\pi}} d\beta$	Gokhale and Mullen(2004), Gokhale and Mullen(2005)

growth phenomenon. In this respect, we develop a class of CGFs in the next section.

3. A Class of Coverage Growth Functions

The CGFs listed in Table 2.1 implicitly assumes that the testing time is continuous. Such CGFs are applicable when the testing time is measured as the execution time or the testing effort. The remaining of this paper assumes that the testing time is the number of executed test cases, *i.e.*, the testing time is discrete. Then we will develop a class of CGFs under the following assumptions:

(i) Constructs in M are executed independently;

(ii) Execution probability p of a construct is a random variable with distribution $F(p)$.

Assumption (i) implies that whether a construct is executed does not depend on other constructs. In general, this does not hold. For example, if some constructs belong to the same execution path, their executions will not be independent. However, if the system is fairly large, the constructs unrelated to a certain construct will be much more than the constructs related to the construct. Then overall dependency among constructs would be negligible. Assumption (i) can be employed for such circumstances. Assumption (ii) is concerned with the probability that a construct is executed by a test case. The execution probability, p, of a construct mainly depends on the testing profile which expresses how to select test cases from the input space. Once the testing profile is given, the execution probability of a specific construct is determined. However, the execution probabilities of the constructs in M are not equal. The distribution $F(p)$ represents the distribution of p over the constructs in M.

Let $\pi(t\,|\,p)$ denote the probability that a construct with execution probability p is executed up to t^{th} test case. Since test cases are randomly selected from the input space, $\pi(t\,|\,p)$ is computed as

$$\pi(t\,|\,p) = 1 - (1-p)^t . \tag{3.1}$$

That is, the time to execution of a construct has the geometric distribution with parameter p. The probability that a construct is executed up to t^{th} test case is therefore obtained as

$$\pi(t) = \int_0^1 \pi(t\,|\,p)\, dF(p). \tag{3.2}$$

Due to Assumption (i), $|M(t)|$ follows a binomial distribution with parameters $|M|$ and $\pi(t)$. It is worthy of note that $\pi(t)$ is the mean of $\pi(t\,|\,p)$ averaged over all the constructs in M. Thus $\pi(t)$ can be interpreted as the proportion of executed constructs in M, *i.e.*, $c(t) = \pi(t)$ holds. In the remaining of this paper, $c(t)$ and $\pi(t)$ will be used interchangeably. Eq. (3.2) generates different $\pi(t)$'s for different $F(p)$'s. Thus we suggest the class of $\pi(t)$'s as a collection of CGFs. Each $\pi(t)$ in the class is identified by the corresponding $F(p)$.

4. Maximum Likelihood Estimation

Suppose that a coverage growth phenomenon has been observed at t_i for $i=1,2,\ldots,n$. Let m_{t_i} and c_{t_i} be the values of $|M(t)|$ and $C(t)$ measured at t_i. The observed increments of $|M(t)|$ and $C(t)$ during $(t_{i-1},t_i]$ are then expressed as $x_{t_i} = m_{t_i} - m_{t_{i-1}}$ and $c_{t_i} - c_{t_{i-1}} = x_{t_i}|M|^{-1}$, where $m_{t_0} = 0$ and $t_0 = 0$. Since a coverage growth phenomenon is repeatedly observed at different times, the observed values m_{t_i}'s and c_{t_i}'s are not independent. That is, the coverage growth behavior during $(t_{i-1},t_i]$ depends on $m_{t_{i-1}}$ and the conditional probability that a construct not executed up to t_{i-1} is executed up to t_i. Since the conditional probability is computed as

$$\frac{\pi(t_i) - \pi(t_{i-1})}{1 - \pi(t_{i-1})}, \tag{4.1}$$

the increment of $|M(t)|$ during $(t_{i-1},t_i]$ has the binomial distribution with parameters $|M| - m_{t_{i-1}}$ and $\left[\pi(t_i) - \pi(t_{i-1})\right]\left[1 - \pi(t_{i-1})\right]^{-1}$. Therefore, for given m_{t_i}'s, c_{t_i}'s and t_i's, the likelihood function is obtained as

$$\begin{aligned} L &= \prod_{i=1}^{n} \binom{|M| - m_{t_{i-1}}}{x_{t_i}} \left[\frac{\pi(t_i) - \pi(t_{i-1})}{1 - \pi(t_{i-1})}\right]^{x_{t_i}} \left[\frac{1 - \pi(t_i)}{1 - \pi(t_{i-1})}\right]^{|M| - m_{t_i}} \\ &= \frac{|M|!}{\prod_{i=1}^{n} x_{t_i}!(|M| - m_{t_n})!} \prod_{i=1}^{n} \left[\pi(t_i) - \pi(t_{i-1})\right]^{x_{t_i}} \left[1 - \pi(t_n)\right]^{|M| - m_{t_n}} \end{aligned}$$

The above likelihood function indicates that the joint distribution of x_{t_i}'s is the multinomial distribution with parameters $|M|$ and $\pi(t_i) - \pi(t_{i-1})$, $i = 1,2,\ldots,n$. The corresponding log likelihood function is

$$\begin{aligned} &\ln L \\ &= K + \sum_{i=1}^{n} x_{t_i} \ln\left(\pi(t_i) - \pi(t_{i-1})\right) + \left(|M| - m_{t_n}\right)\ln\left(1 - \pi(t_n)\right) \\ &= K + |M|\Big[\sum_{i=1}^{n}(c_{t_i} - c_{t_{i-1}})\ln\left(\pi(t_i) - \pi(t_{i-1})\right) + (1 - c_{t_n})\ln\left(1 - \pi(t_n)\right)\Big], \\ &= K + |M| \cdot L^* \end{aligned} \tag{4.2}$$

where K is a constant independent of parameters to be estimated. Maximum likelihood estimates can be obtained by maximizing L^*.

5. Numerical Illustration

In this section we illustrate how to use the proposed class of CGFs in practice. First, we have to select a specific $\pi(t)$ from the class. Selection of $\pi(t)$ is equivalent to specification of $F(p)$. We suggest the beta distribution as one of the plausible $F(p)$'s. If $F(p)$ is the beta distribution with parameters α and β, then

$$\pi(t) = \int_0^1 \left[1-(1-p)^t\right] \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)} \tag{5.1}$$

$$= 1 - \frac{B(\alpha,\beta+t)}{B(\alpha,\beta)}, \tag{5.2}$$

where $B(\alpha,\beta)$ is the beta function. The CGFs given by Eq. (5.2) is now applied to three real data sets reported by Vouk(1992). The three data sets are referred to as DS1, DS2 and DS3, respectively. Vouk(1992) collected the three data sets from a NASA supported project implementing sensor management in an inertial navigating system. Each data set consists of values of the number of executed test cases, the number of detected faults and four coverage metrics. The four coverage metrics are respectively block, branch, c-use and p-use coverages. The proposed CGF are applied to each coverage metric of each data set. The CGF given by Eq. (5.2) was fitted to each coverage metric of these data sets. The maximum likelihood estimates are obtained by maximizing L^* and presented in Tables 5.1–5.3. The observed coverage values c_{t_i}'s and the fitted CGFs are plotted in Figures 5.1–5.3. The CGF given by Eq. (5.2) works well for DS1-DS3.

Table 5.1: Maximum likelihood estimates for DS1

	coverage			
parameter	block	branch	c-use	p-use
α	0.2820	0.2637	0.1347	0.1323
β	0.3506	0.4804	0.1314	0.5206

Table 5.2: Maximum likelihood estimates for DS2

	coverage			
parameter	block	branch	c-use	p-use
α	0.4004	0.4036	0.3911	0.2398
β	0.6060	0.7336	0.4085	0.3929

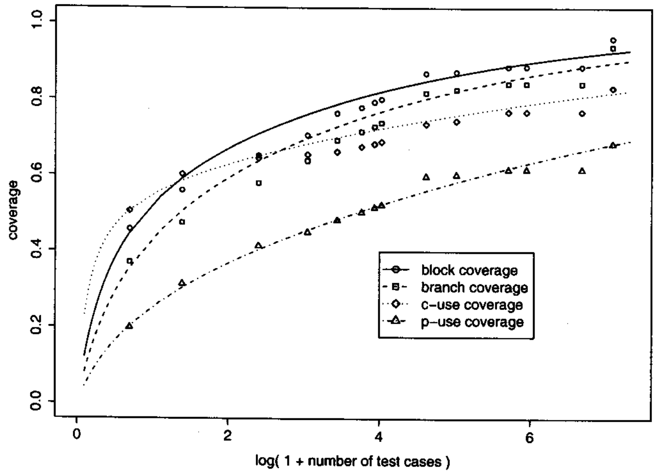

Figure 5.1: Coverage growth functions for DS1

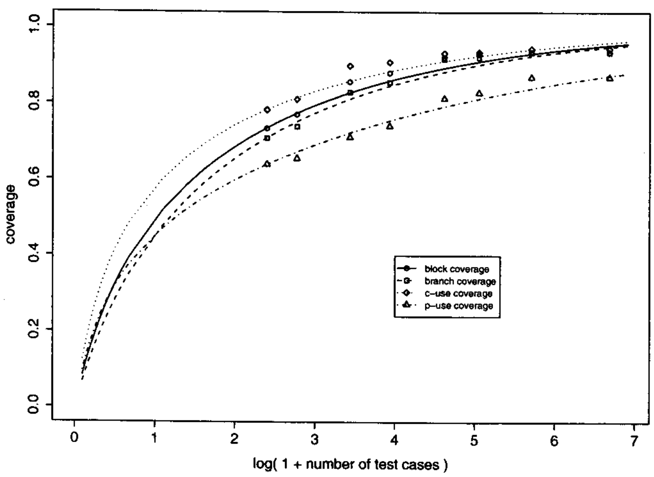

Figure 5.2: Coverage growth functions for DS2

Table 5.3: Maximum likelihood estimates for DS3

	coverage			
parameter	block	branch	c-use	p-use
α	0.2915	0.2642	0.1931	0.1952
β	0.2355	0.2731	0.0857	0.3683

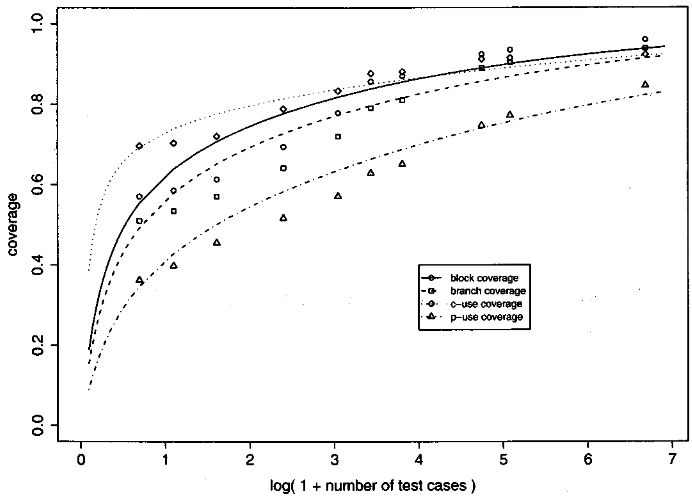

Figure 5.3: Coverage growth functions for DS3

6. Concluding Remarks

A recent trend in developing NHPP SRGMs is to incorporate some additional information other than the testing time into the NHPP SRGMs. As the testing progresses, only the faults located in the covered portion of the software system, not all the faults in the software system, are exposed to the fault detection activity. Thus the coverage-based NHPP SRGMs have been developed. Since one of the most important factors of the coverage-based NHPP SRGMs is the CGF, we modeled the stochastic process underlying the coverage growth phenomenon. The resulting model provides us with a class of discrete time CGFs. Each CGF of the class is identified with the distribution of execution probability p. It was also shown that the CGF based on the beta distribution performs well for some real data sets. However, it is necessary to investigate performance of NHPP

SRGMs, which are combinations of the coverage-based NHPP SRGM frameworks given by Eqs. (1.1)–(1.4) and the CGF of Eq. (5.2). Plausible distributions other than the beta distribution are to be searched. In addition, the continuous time version of the proposed class is to be developed and compared with the currently available CGFs.

References

Fujiwara, T. and Yamada, S. (2002). Co coverage-measure and testing-domain metrics based on a software reliability growth model. *International Journal of Reliability, Quality and Safety Engineering*, **9**, 329–340.

Gokhale, S. S. and Mullen, R. E. (2004). From test count to code coverage using the lognormal failure rate. In *Proceedings 15th ISSRE*, St. Malo, Bretagne.

Gokhale, S. S. and Mullen, R. E. (2005). Dynamic code coverage metrics: a lognormal perspective. In *Proceedings of the 11th IEEE International Software Metrics Symposium.*

Gokhale, S. S., Philip, T., Marinos, P. N. and Trivedi, K. S. (1996). Unification of finite failure non-homogeneous Poisson process models through test coverage. In *Proceedings of the 7th International Symposium on Software Reliability Engineering*, 299–307, White Plains, New York.

Gokhale, S. S. and TrivediK, K. S. (1999). A time/structure based software reliability model. *Annals of Software Engineering* **8**, 85–121.

Grottke, M. (2002). A vector Markov model for structural coverage growth and the number of failure occurrences. In *Proceedings of the 13th International Symposium on Software Reliability Engineering*, 304–315, Annapolis, Maryland.

Malaiya, Y. K., Li, M. N., Bieman, J. M. and Karcich, R. (2002). Software reliability growth with test coverage. *IEEE Transactions on Reliability,* **51**, 420–426.

Park, J.-Y., Hwang, Y.-S. and Fujiwara,,T. (2005a). Integration of imperfect debugging in general testing-domain dependent NHPP SRGM. *International Journal of Reliability, Quality and Safety Engineering* **12**, 493–505.

Park, J.-Y., Fujiwara, T. and Park, J. H. (2005b). A coverage function for arbitrary testing profile and its performance. *International Journal of Reliability and Applications,* **6**, 87–99.

Park, J.-Y. and Fujiwara, T. (2006). Coverage growth functions for software reliability modeling. In *Proceedings of 2nd AIWARM*, 435–442.

Pham, H. and Zhang, X. (2003). NHPP software reliability and cost models with testing coverage. *European Journal of Operational Research* **145**, 443–454.

Piwowarski, P., Ohba, M. and Caruso, J. (1993). Coverage measure experience during function test. In *Proceedings of the 15th international conference on Software Engineering*, 287–301, Baltimore, MD.

Sedigh-Ali, S., Ghafoor, A. and Paul, R. A. (2002). Temporal modeling of software test coverage. In *Proceedings of the 26th Annual International Computer Software and Applications Conference*, 823–828, Orlando, Florida.

Vouk, M. A. (1992). Using reliability models during testing with no-noperational profiles. *Proceedings of 2nd Bellcore /Purdue Workshop Issues in Software Reliability Estimation,* 103–111.

Yamamoto, T., Inoue, S. and Yamada, S. (2004). A software reliability growth model with testing-coverage maturity process. In *Proceedings of the Tenth ISSAT International Conference on Reliability and Quality in Design*, 299-303, Las Vegas, NV.

Yamada, S. and Fujiwara, T. (2001). Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit. In *Proceedings of the 7th ISSAT International Conference on Reliability and Quality in Design*, **8**, 205–218.

[Received July 2007, Accepted July 2007]