International Journal of Reliability and Applications
Vol. 6, No. 2, pp. 87-99, 2005

A Coverage Function for Arbitrary Testing Profile and Its Performance

Joong-Yang Park*
Department of Information Statistics, Gyeongsang National University
Jinju, 660-701, Republic of Korea

Takaji Fujiwara
Development Dept. 3, Development Division, Fujitsu Periperals Limited
35 Saho, Yashiro-cho, Katoh-gun Hyogo, 673-1447, Japan

Jae-Heung Park
Department of Computer Science, Gyeongsang National University
Jinju, 660-701, Republic of Korea

Abstract. Coverage-based software reliability growth models (SRGMs) have been developed and successfully applied in practice. Performance of a coverage-based SRGM depends on the coverage function employed by the SRGM. When the coverage function represents the coverage growth behavior well irrespective of type of the testing profile, the corresponding coverage-based SRGM is expected to be widely applicable. This paper first conducts a study of selecting the most representative coverage functions among the available coverage functions. Then their performances are empirically evaluated and compared. The result provides a foundation for developing widely applicable coverage-based SRGMs and monitoring the progress of a testing process.

Key Words : *construct, coverage function, coverage growth rate, software reliability growth model, testing profile.*

1. INTRODUCTION

Recently software is becoming an integral part of computer system. 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

*Corresponding Author.
E-mail address: joongyang@nongae.gsnu.ac.kr

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. Fault detection and removal data collected during testing are used for estimation of reliability measures. Developers usually determine when to stop testing and release the software based on the estimates of reliability measures. In this context, many SRGMs have been proposed and applied in practice to estimate software reliability measures.

A new trend for modeling software reliability is to incorporate coverage information into SRGMs. Such SRGMs, e.g., Gokhale et al. (1996), Gokhale and Trivedi (1999), Grottke (2002), Malaiya et al. (2002), Park et al. (2004), Pham and Zhang (2003), Piwowarski et al. (1993) and Rivers and Vouk (1995), are referred to as the coverage-based SRGMs. This trend is based on the observation that the more a software system is covered, the more likely reliable is the software system. The coverage information is important for both software developers and software consumers. It helps software developers assess how thoroughly the software has been tested. On the other hand, it provides software consumers with the confidence of using the software. As the testing proceeds and the coverage grows, the number of detected faults tends to increase since the faults are distributed over and located at the constructs. This implies that the coverage growth behavior during the testing directly influences the fault detection process and consequently software reliability. Most time domain coverage-based SRGMs such as Gokhale et al. (1996), Gokhale and Trivedi (1999), Grottke (2002), Malaiya et al. (2002), Pham and Zhang (2003) and Piwowarski et al. (1993) adopt a coverage function, which represents the coverage growth behavior during the testing. Performance of a time domain coverage-based SRGM depends on how closely its coverage function represents the coverage growth behavior. It is worthy of note that non-time domain models such as Park et al. (2004) and Rivers and Vouk (1995) can be transformed to time domain models by substituting coverage in the model with the coverage function.

The coverage growth is mainly determined by the strategy of selecting test cases, i.e., the testing profile. However, most coverage-based SRGMs do not explicitly specify the testing profile. In order for a coverage-based SRGM to be widely applicable, its coverage function should be able to represent the coverage growth produced by arbitrary testing profile. Otherwise, its application is limited. It is therefore necessary to examine the ability of available coverage functions to represent the coverage growth behavior.

This paper considers the coverage growth in the testing environment where the specified testing profile remains unchanged during the testing and test cases are randomly chosen according to the testing profile. The primary aim is to suggest a coverage function that is able to represent the coverage growth behavior irrespective of type of the testing profile. First we attempt to capture the typical coverage growth behavior by means of simulation for some selected testing profiles in Section 2. Section 3 presents currently available coverage functions and chooses the plausible coverage functions from theoretical and empirical aspects. Then a practical modification is applied to the chosen coverage functions. Performance of the chosen coverage functions is evaluated empirically by applying them to real data sets in Section 4. Conclusions are given in Section 5.

2. TYPICAL COVERAGE GROWTH BEHAVIOR IN PROFILE-BASED TESTING

We begin by defining the coverage function. Denote by M the set of all the constructs of a software under testing. Constructs may be statements, blocks, branches, p-uses or c-uses depending on the coverage metric under consideration. Let $|\cdot|$ be the cardinality of a set of constructs. As the testing progresses, the number of covered constructs increases. We represent the set of constructs covered up to testing time t by $M_c(t)$. Then the coverage at testing time t is defined as $C(t) = |M_c(t)||M|^{-1}$. Since the number of constructs executed by a test case is usually modeled as a random variable, $|M_c(t)|$ and $C(t)$ are also considered as random variables. Thus the coverage function is defined as the expected value of $C(t)$, i.e., $c(t) = E[C(t)]$.

Suppose that a testing profile is given and that test cases are randomly chosen from the input domain according to the testing profile. In general, some of the constructs executed by a test case are in $M_c(t)$ and others are in $M - M_c(t)$. That is, some constructs are executed repeatedly. This is referred to as the redundant execution of constructs. The redundant execution of constructs indicates that some constructs may be executed more often than others and that all constructs are not equally likely to be executed. Thus the testing profile can be regarded as a distribution of selection probability defined over M. In order to appreciate typical coverage growth behavior, we consider 4 testing profiles depicted in Figure 1. They are respectively the uniform, linear, concave and convex profiles. Rearranging the constructs in M in ascending order of selection probability, the shape of a testing profile will be similar to one of the 4 profiles. Denoting the selection probability of ith construct by $p_i, i = 0, 1, \ldots, |M| - 1$, these testing profiles are given as follows:

(i) uniform profile: $p_i = \frac{1}{|M|}$

(ii) linear profile: $p_i = \frac{0.2}{|M|} + 0.8\frac{i}{\sum_{j=0}^{|M|-1} j}$

(iii) convex profile: $p_i = \frac{0.2}{|M|} + 0.8\frac{i^2}{\sum_{j=0}^{|M|-1} j^2}$

(iv) concave profile: $p_i = \frac{0.2}{|M|} + 0.8\frac{p^*_{|M|-1} - p^*_{|M|-1-i}}{\sum_{j=0}^{|M|-1}\left[p^*_{|M|-1} - p^*_{|M|-1-j}\right]}$, where p^*_i is p_i given in (iii).

As explained later in Section 3, the coverage growth has been analytically studied only for the uniform profile. The coverage growth for non-uniform profiles has not been derived yet mainly due to its complexity. We thus attempt to capture typical coverage growth behavior via simulation based on the 4 profiles. Figure 2 shows how the coverage grows as the testing is performed according to the 4 profiles.

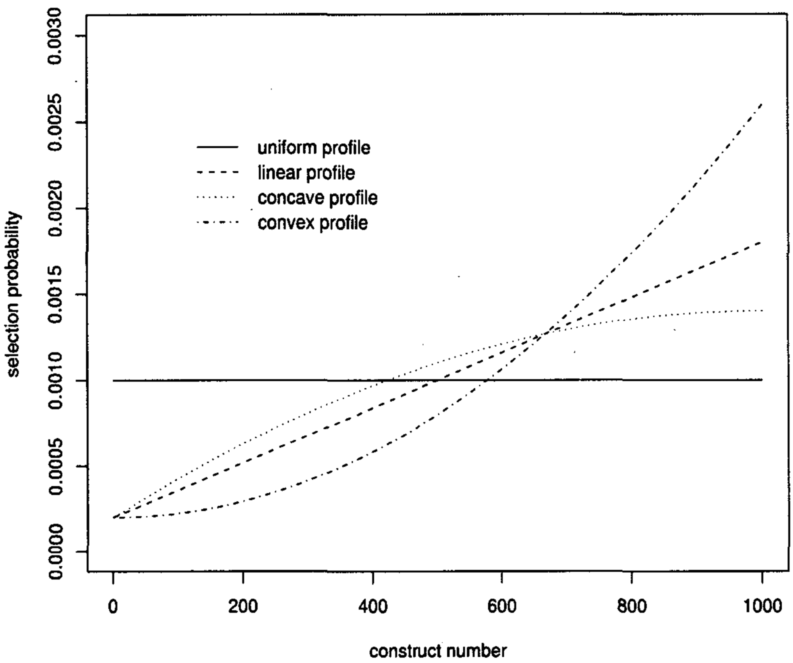

Figure 1. Testing profiles for simulating coverage growth behavior.

It is assumed that $|M| = 1000$ and the number of constructs executed per test case is a constant 10. The expected number of constructs executed by a test case is determined by the characteristics of the software. It usually remains constant during the testing. However, the expected number of constructs newly covered by a test case tends to decrease, because execution of test cases expands $M_c(t)$. That is, the coverage growth rate, $dc(t)/dt$, is nonincreasing. Figure 2 complies with this argument. Figure 2 also shows that the coverage growth rate of the uniform profile is greater than non-uniform profiles. It should be noted that the above simulation result and discussion are valid for the testing environment where test cases are randomly selected according to the specified testing profile and the testing profile remains unchanged during the testing. If test cases are generated so as to execute as many uncovered constructs as possible and learning takes place in test case generation, the coverage growth rate may increase especially in the early phase of testing. Then s-shaped coverage growth can be observed. Such testing strategies are not in the scope of this paper.

3. COVERAGE FUNCTIONS

In this section we first present coverage functions appeared in the literature and then choose a coverage function among them which approximates well the coverage growth behavior. Finally a practical modification is applied to the chosen coverage function.

Table 1. Currently available coverage functions.

coverage function	parameter constraints and shape	references
$c_1(t) = 1 - \exp(-\beta t)$	$0 < \beta$ exponential	Piwowarski et al. (1993) Gokhale et al. (1996)
$c_2(t) = \beta_0 \ln(1 + \beta_1 t)$	$0 < \beta_0, 0 < \beta_1$ exponential	Malaiya et al. (2002)
$c_3(t) = 1 - \exp(-\beta t^{\gamma})$	$0 < \beta, 0 < \gamma$ exponential	Gokhale et al. (1996)
$c_4(t) = 1 - (1 + \beta t)\exp(-\beta t)$	$0 < \beta$ s-shape	Gokhale et al. (1996)
$c_5(5) = \frac{(\beta t)^{\gamma}}{1+(\beta t)^{\gamma}}$	$0 < \beta, 0 < \gamma$ s-shape	Gokhale and Trivedi (1999)
$c_6(t) = 1 - (1 - \beta\delta t)^{1/\delta}$	$0 < \beta, 0 < \delta \leq 1$ exponential	Grottke (2002)
$c_7(t) = b - c\exp(-\beta t)$	$0 \leq c \leq b \leq 1, 0 < \beta$ exponential	Sedigh-Ali et al. (2002)
$c_8(t) = b - \frac{c\exp(\beta t)}{1+\exp(-2\beta t)}$	$0 \leq c \leq 2b, 0 < \beta$ s-shape	Sedigh-Ali et al. (2002)
$c_9(t) = \frac{1-\exp(-\beta t)}{1+\gamma\exp(-\beta t)}$	$0 < \beta, 0 < \gamma$ exponential for $0 < \gamma \leq 1$ s-shape for $\gamma \geq 1$	Yamamoto et al. (2004)

The currently available coverage functions are summarized in Table 1. The coverage functions $c_1(t)$, $c_6(t)$ and $c_9(t)$ are analytical models based on certain assumptions. Especially, $c_1(t)$ is the coverage function for the uniform profile. On the contrary, the others are empirical coverage functions. In this section we select from the coverage functions listed in Table 1 a coverage function that is able to describe the coverage growth behavior well from theoretical and empirical aspects. As explained in the previous section and shown in Figure 2, the coverage growth rate is to be nonincreasing. Therefore, s-shape coverage functions $c_4(t)$, $c_5(t)$ and $c_8(t)$ are not appropriate for the testing environment under our consideration. Theoretically coverage functions should meet the conditions $c(0) = 0$, $c(\infty) = 1$ and $0 \leq c(t) \leq 1$ for all $t \geq 0$. Since $c_7(t)$ is proposed for multi-phase testing, it is not forced to meet the conditions $c(0) = 0$ and $c(\infty) = 1$. If we subject $c_7(t)$ to the conditions, $c_7(t)$ becomes identical to $c_1(t)$. However, $c_2(t)$ and $c_6(t)$ do not satisfy these conditions. Specifically, $c_2(t) \leq 1$ for $t \leq \left[\exp\left(\beta_0^{-1}\right) - 1\right] \beta_1^{-1}$ and $c_6(t) \leq 1$ for $t \leq 1/(\beta\delta)$. It should also be noted that $c_1(t)$ is a special case of $c_3(t)$ and $c_9(t)$. Judging from the above discussion, we can select $c_3(t)$ and $c_9(t)$ as the plausible coverage functions for the profile-based testing.

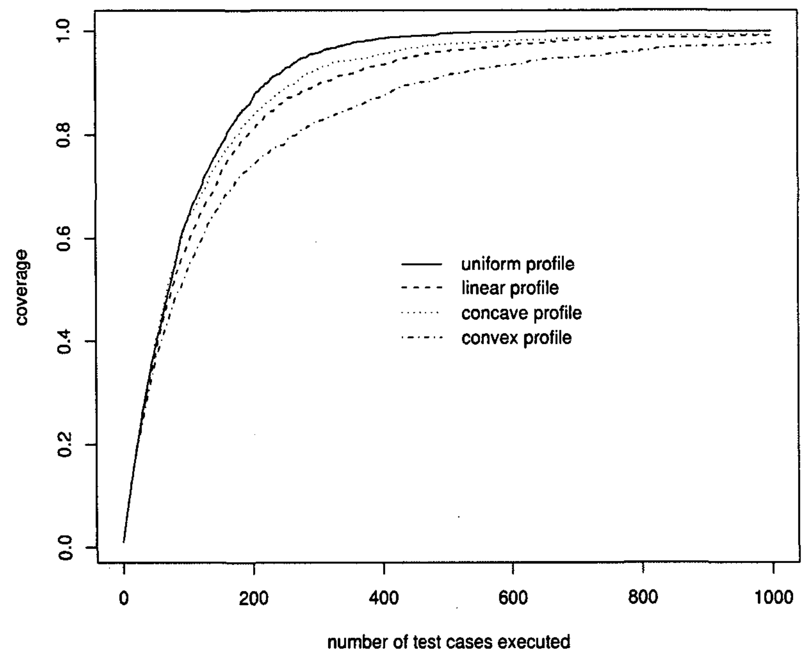

Figure 2. Simulated coverage growth behavior of the testing profiles in Figure 1.

There is a practical aspect to be integrated into a coverage function. It is generally accepted that 100% coverage is rarely achieved because of the presence of infeasible constructs and constructs with extremely small selection probability. The coverage functions $c_3(t)$ and $c_9(t)$ are thus modified to meet this practical observation. That is,

$$c_3^*(t) = c_{max} c_3(t) \text{ and } c_9^*(t) = c_{max} c_9(t) \tag{3.1}$$

where c_{max} is the maximum achievable coverage for the given testing profile. Actually $c_9^*(t)$ is the coverage function proposed in Yamamoto et al. (2004).

4. EMPIRICAL PERFORMANCE EVALUATION

The coverage functions $c_3^*(t)$ and $c_9^*(t)$ are empirically evaluated and compared by fitting them to 4 real data sets, DS1-DS4. DS1, reported by Pasquini et al. (1996), is collected from a configuration software for an array of antenas developed by the European Space Agency. It consist of 29 observations of the testing time in the number of executed test cases and 4 coverage values. The 4 coverages are respectively block, branch, c-use and p-use coverages. The first 28 observations are used for fitting and the last observation is used for validation of the paramater estimates of the proposed model. DS2-DS4, reported by Vouk (1992), are from a NASA supported project implementing sensor management in an inertial navigation system. They also contain values of the testing time and the same 4 coverages. By fitting both $c_3^*(t)$ and $c_9^*(t)$ to DS1-DS4, we found that $c_3^*(t)$ performs significantly better than $c_9^*(t)$ for these data sets. For the sake of briefness, the results of fitting to the block coverage of DS2 and the p-use coverage of DS3 are presented in Table 2 and Figures 3 and 4. We thus choose $c_3^*(t)$ as the best among the available coverage functions.

Table 2. Least squares estimates of parameters in $c_3^*(t)$ and $c_9^*(t)$ for the block coverage of DS2 and the p-use coverage of DS3.

	block coverage of DS2		p-use coverage of DS3	
parameter	$c_3^*(t)$	$c_9^*(t)$	$c_3^*(t)$	$c_9^*(t)$
c_{max}	0.9597	0.9597	0.8986	0.8691
β	0.6577	0.2996	0.6367	0.1027
γ	0.2479	4.1320×10^{-6}	0.2720	1.7370×10^{-6}
MSE	3.9863×10^{-4}	3.0277×10^{-2}	2.1182×10^{-4}	8.3484×10^{-2}

We further evaluate performance of $c_3^*(t)$ by applying it to the above mentioned data sets. The $c_3^*(t)$'s fitted to 4 coverages of DS1 are plotted in Figure 5 and the parameter estimates and MSE are presented in Table 3. The testing time and coverage values of the last observation of DS1 are respectively 20000, 0.82, 0.70, 0.74, and 0.67. The estimates of c_{max} are very close to the coverage values of the last observation. The fitting results for DS2-DS4 are given in Figures 6-8 and Tables 4-6. It is evident that $c_3^*(t)$ works well for all the data sets.

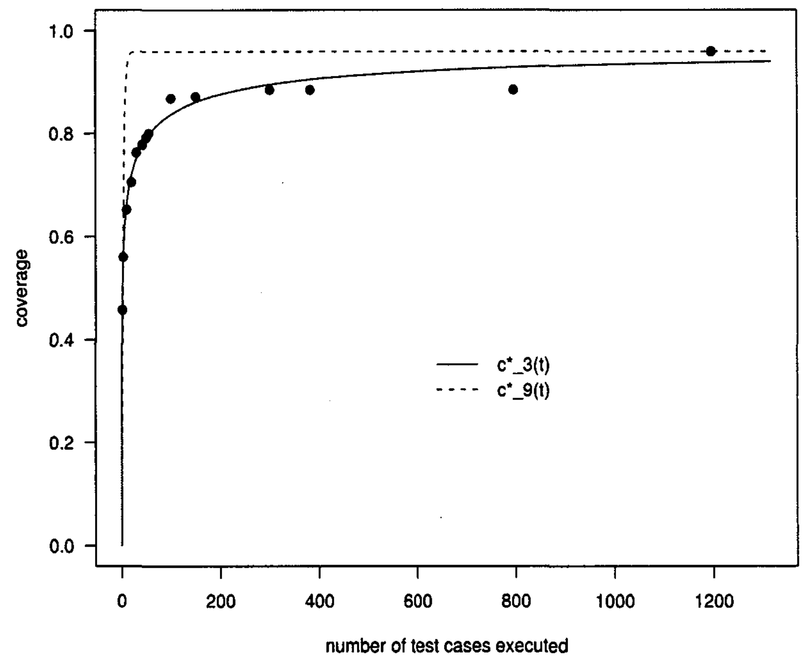

Figure 3. $c_3^*(t)$ and $c_9^*(t)$ fitted to the block coverage of DS2.

Table 3. Least squares estimates of parameters in $c_3^*(t)$. (DS1)

parameter	coverage			
	block	branch	c-use	p-use
c_{max}	0.8000	0.6820	0.7301	0.6699
β	0.6032	0.3781	0.4834	0.4374
γ	0.3561	0.4263	0.4124	0.3620
MSE	4.6229×10^{-4}	2.0274×10^{-4}	4.6694×10^{-4}	2.2705×10^{-4}

Table 4. Least squares estimates of parameters in $c_3^*(t)$. (DS2)

parameter	coverage			
	block	branch	c-use	p-use
c_{max}	0.9597	0.9376	0.9107	0.6829
β	0.6577	0.5235	0.8632	0.4155
γ	0.2479	0.2681	0.1296	0.3152
MSE	3.9863×10^{-4}	5.8429×10^{-4}	3.2583×10^{-4}	5.0281×10^{-4}

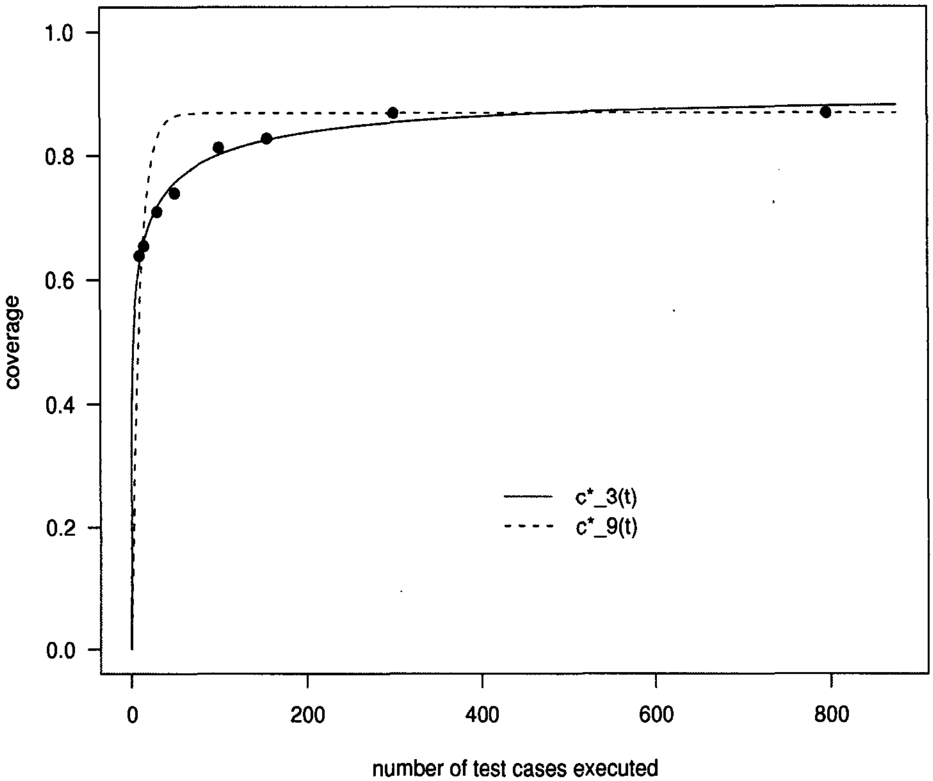

Figure 4. $c_3^*(t)$ and $c_9^*(t)$ fitted to the p-use coverage of DS3.

Table 5. Least squares estimates of parameters in $c_3^*(t)$. (DS3)

	coverage			
parameter	block	branch	c-use	p-use
c_{max}	0.9357	0.9397	0.9438	0.8986
β	0.5759	0.5663	0.6549	0.6367
γ	0.4175	0.3827	0.4223	0.2720
MSE	4.2061×10^{-5}	9.3382×10^{-5}	9.1593×10^{-5}	2.1182×10^{-4}

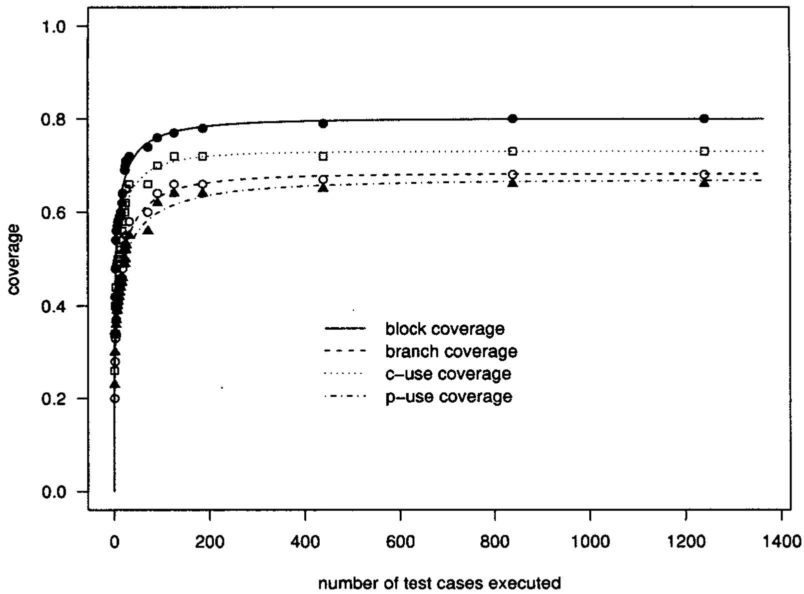

Figure 5. $c_3^*(t)$'s fitted to DS1.

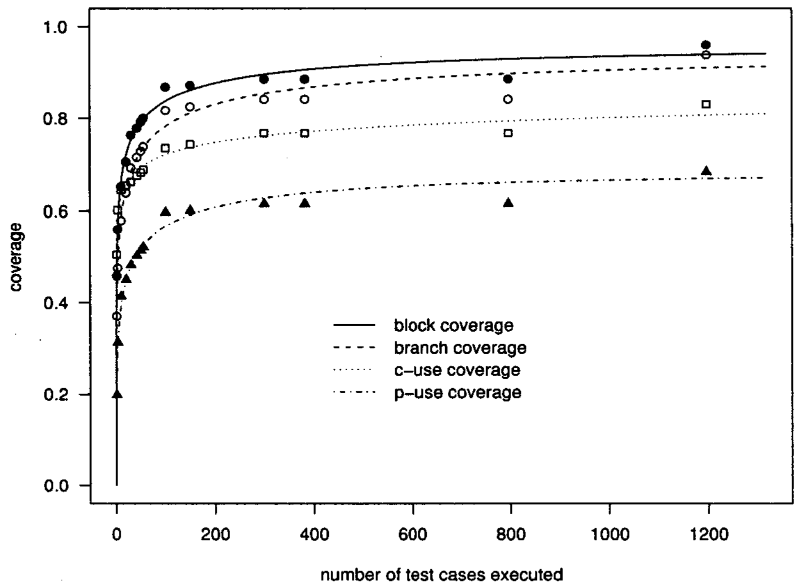

Figure 6. $c_3^*(t)$'s fitted to DS2.

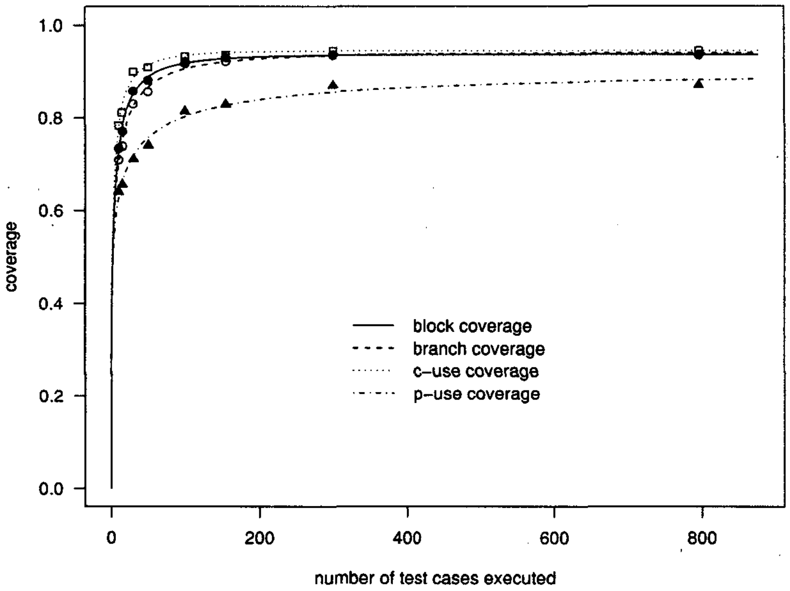

Figure 7. $c_3^*(t)$'s fitted to DS3.

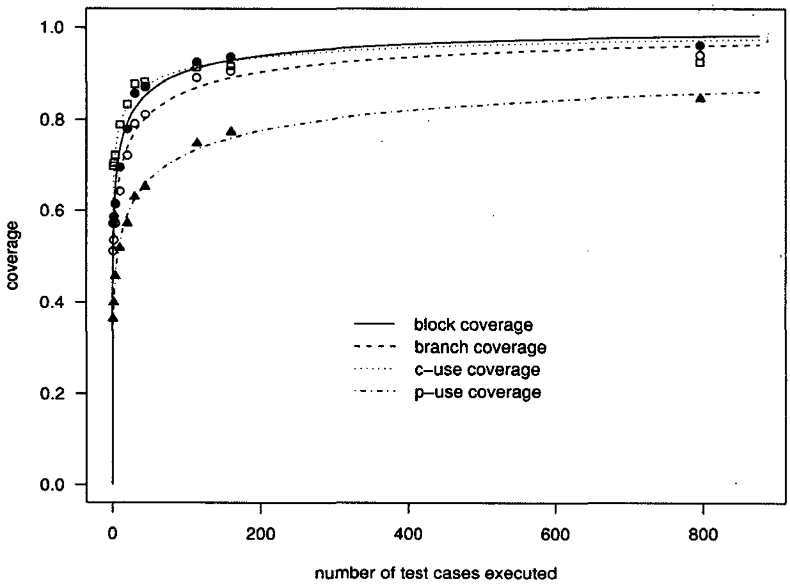

Figure 8. $c_3^*(t)$'s fitted to DS4.

Table 6. Least squares estimates of parameters in $c_3^*(t)$. (DS4)

	coverage			
parameter	block	branch	c-use	p-use
c_{max}	0.9990	0.9925	0.9999	0.9281
β	0.7487	0.6499	0.9999	0.4719
γ	0.2479	0.2460	0.1887	0.2525
MSE	1.0229×10^{-3}	6.9015×10^{-4}	1.1251×10^{-3}	1.0653×10^{-4}

5. CONCLUSIONS

In this paper we have proposed a coverage function which is able to represent the coverage growth behavior of the profile-based testing. It has been shown empirically that the proposed coverage function performs well irrespective of type of the testing profile and test coverage metric. It provides us with a motivation of developing new SRGMs based on it. Since the coverage function shows the progress of the testing under the given testing profile, the proposed coverage function can be also used for determining when to stop testing and release the software. Furthermore, it is useful for deciding whether we need to change the current testing profile for accelerating software reliability improvement and testing progress.

REFERENCES

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. *Proceedings of 7th IEEE International Symposium on Software Reliability Engineering*, 299-307.

Gokhale, S. S. and Trivedi, K. S. (1999). A Time/Structure Based Software Reliability Model. *Annals of Software Engineering*, 85-121.

Grottke, M. (2002). A Vector Morkov Model for Structural Coverage Growth and the Number of Failure Occurrences. *Proceedings of 13th International Symposium on Software Reliability Engineering.*

Malaiya, Y. K., Li, M. N., Bieman, J. M. and Karcich, R. (2002). Software Reliability Growth and Test Coverage. *IEEE Transactions on Reliability*, **51**, 420-426.

Park, J-Y, Park, J-H and Kim, Y-S (2004). A Coverage-Based Software Reliability Growth Model for Imperfect Fault Detection and Repeated Construct Execution, *Transactions of Korean Information Processing Seciety*, 11-D, 1287-1294.

Pasquini, A., Crespo, A. N. and Matrella P. (1996). Sensitivity of Reliability-Growth Models to Operational Profile Errors vs Testing Accuracy. *IEEE Transactions on Reliability*, **45**, 531-540.

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. *Proceedings of 15th IEEE International Conference on Software Engineering*, 287-301.

Rivers, A. T. and Vouk, M. A. (1995). An Empirical Evaluation of Testing Efficiency During Non-Operational Testing. *Proceedings of 4th Software Engineering Research Forum*, 111-120.

Sedigh-Ali, S., Ghafoor, A. and Paul, R. A. (2002). Temporal Modeling of Software Test Coverage. *Proceedings of 26th Annual International Computer Software and Application Conference.*

Vouk, M. (1992). Using Reliability Models During Testing with Non-Operational 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, *Proceedings of 10th ISSAT International Conference.*