数理解析研究所講究録 1252 巻 2002 年 110-116

Analysis of Hypergeometric Distribution Software Reliability Model

土肥正, 若菜展行, 尾崎俊治, Kishor S. Trivedi
T. Dohi†, N. Wakana†, S. Osaki‡ and Kishor S. Trivedi††
†Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan
‡Department of Mathematical and Information Engineering, Nanzan University, Japan
††Department of Electrical and Computer Engineering, Duke University, USA

1 Introduction

Software reliability is an important dependability measure in supplying reliable software products to users. However, the problem of assessing the software reliability accurately is not easy, since a typical program involves a large number of logical paths, and the faults which may have remained on some logical paths are not completely detected within the limited testing period. Hence, many mathematical models called ***software reliability growth models*** (SRGMs) have been developed to describe the software debugging phenomenon and to assess the software reliability from the empirical software failure data [22, 25, 26, 27, 31, 32, 38]. Usually, two kinds of modeling approaches, ***white-box approach*** and ***black-box approach***, are applied in the actual software testing phase. In the white-box approach, the software test is executed based on ***the software architecture*** taking account of the action by each software module, and strongly depends on the internal structure of the software [2, 16, 17, 18, 19, 20]. However, such a modeling approach is not always possible, since it is very hard to identify the module structure and its transition probability law, especially for large scale real software systems. On the other hand, the black-box approach ignores the software architecture and regards the software failure occurrence phenomenon as well-defined stochastic processes [1, 14, 21, 24, 30, 41]. The advantage in the black-box approach is the ease of parameter estimation from the real software failure data. Thus, during the past three decades this modeling approach has often been applied to actual software development projects. In the recent software development process, these approaches are used in conjunction with each other.

The hypergeometric distribution software reliability model (HGDSRM) should be classified into the black-box model, but would be a distinguished SRGM from the other models based on the non-homogeneous Poisson processes (NHPPs) or the Markov/semi-Markov processes [1, 14, 21, 24, 30, 41]. The HGDSRM can model the physical and myopic debugging behavior in the probability distribution function. First, Xizi [39, 40] proposed the idea to apply the hypergeometric distribution for predicting software reliability. Independently, Tohma et al. [33, 34] developed an interesting SRGM based on the hypergeometric distribution to estimate the number of faults remaining in the software. Since the seminal contribution by Tohma et al. [33, 34], this type of SRGM has been improved from the various points of view. Jacoby and Tohma [10, 12] gave a dynamic representation for the HGDSRM in [33, 34] by solving a recursive equation, and referred to the relationship between the HGDSRM and the NHPP type of SRGMs in [1, 41]. That is, they succeeded to express the typical patterns such as the exponential growth curve and the S-shaped growth curve in the model parameters of the HGDSRM. Similar approaches were made by Hou et al. [3, 4]. They also proposed several parametric models taking account of the learning curve effect and the imperfect debugging. Jacoby and Masuzawa [13] showed that the HGDSRM can be described as a function of the test coverage for software under test. Other coverage models were developed by [28, 29, 31].

Parameter estimation methods for the HGDSRMs have also been studied. Tohma et al. [11, 35, 36, 37] focused on both the maximum likelihood method and the least square method to estimate the model parameters from the software fault data, and compared the methods on some real data sets. In fact, the problem to estimate the model parameters has not been solved consistently in the statistical theory even for a simple two-parameter hypergeometric distribution function [15]. Hence, their effort should be encouraged to apply the HGDSRM to the real software testing process. However, every method proposed in [11, 35, 36, 37] is not always acceptable, since some of their methods are based on intuitive approximation schemes. The method with the genetic algorithm in [23] may be positioned as a heuristic estimation method. In the recent years, Hou et al. [6, 8, 9] considered the optimal software release problems based on the HGDSRM and gave the optimal release schedules which minimize the total expected software costs. Also, the optimal allocation problems of testing resources for software module testing were formulated by the same authors [5, 7] using the HGDSRM. In this way, though the HGDSRM is quite a simple probability model to describe the software debugging phenomenon, it has received considerable attention in software reliability engineering over the last decade.

This article gives the detailed mathematical results on the hypergeometric software reliability model. In the earlier papers, Jacoby and Tohma [10, 12] and Tohma et al. [35, 36] derived a recursive formula on the mean cumulative number of software faults detected up to the i-th (> 0) test instance in testing phase. The derivation of the recursive formula is rather heuristic but is correct. Since their results are based on only the mean value of the cumulative number of faults, however, it is impossible to estimate not only the software reliability but also the other probabilistic dependability measures. Noting that the main purpose of the SRGMs is to estimate the software reliability, the probabilistic argument on the HGDSRMs should be developed similar to other SRGMs [22, 25, 26, 27, 31, 32, 38]. To do this, the HGDSRMs have to be treated as discrete-time stochastic processes and need to be further studied from the mathematical view point. We introduce the concept of cumulative trial processes and describe the dynamic behavior of the HGDSRM exactly. In particular, we derive explicit probability mass function (pmf) of the number of software faults detected newly at the i-th test instance and its mean as well as the software reliability defined as the probability that no faults are detected up to an arbitrary time.

The paper is organized as follows. In Section 2, we describe the basic HGDSRM. Section 3 derives some new mathematical results on the HGDSRM with the inductive argument. We give exact expressions for the software reliability and the expected number of newly detected faults at each test instance. These software reliability measures are rather complex, but are quite significant for data analysis.

2 Basic Results on the HGDSRM

Following Tohma et al. [33, 34], suppose that the test of a software constitutes a set of *test instances*, each test instance consists of input test data and observed test result. Define the software test by $D = \{t(i) | i = 1, 2, \cdots, \}$, where $t(i)$ is the i-th ($i = 0, 1, 2, \cdots$) test instance. Let $B = \{b(j) | j = 1, 2, \cdots, m\}$ denote the set of faults in the software before the initial test instance ($i = 0$), where $b(j)$ means the fault labelled by j ($= 1, 2, \cdots, m$) and m (> 0) is the initial number of faults. If a software error caused by $b(j)$ is observed at the test instance $t(i)$, the fault $b(j)$ is said to be *sensed* by the test instance $t(i)$. Suppose that a test instance $t(i)$ senses $w(i)$ software faults, where $w(i)$ is called *the sensitivity factor* and is a function of the number of test instances (or time). We make the following assumptions:

(A-1) The software faults that manifest themselves upon the application of a test instance $t(i)$ will be fixed before the next test instance $t(i+1)$ is applied.

(A-2) No new faults are introduced during software testing. This means that the software reliability is monotonically nondecreasing as the testing progresses.

(A-3) A random set of $w(i)$ software faults is sensed by test instance $t(i)$ out of the total m initial faults.

From these assumptions, it is evident that the number of faults detected by the first test instance $t(1)$ is $w(1)$. However, the number of *newly* detected faults by $t(2)$ is not necessarily $w(2)$, since some of $w(2)$ faults may have been already detected and removed by $t(1)$.

Suppose that the initial number of detected faults at $i = 0$ is 0. Let X_i be the number of newly detected faults at i-th test instance and be a positive random variable. Then, the cumulative number of detected faults until test instance $t(i)$ is described by

$$C_i = \sum_{j=1}^{i} X_j. \tag{1}$$

We make the following additional assumptions:

(B-1) The initial number of faults m is sufficiently larger than $w(i)$, i.e. $m \gg w(i)$ for all $i = 1, 2, 3, \cdots$.

(B-2) In the software test, it is impossible to detect all faults with probability one, i.e. $m > \lim_{i \to \infty} \sum_{j=1}^{i} x$ where x_i is the realization of the random variable X_i.

These assumptions are not explained in the literature [33, 34], but are needed to describe the well-defined HGDSRM. From (B-1) and (B-2), it can be seen that $\min\{w(i), m - c_{i-1}\} = w(i)$ has to be always satisfied. In fact, this assumption is plausible intuitively and is implicitly used in the earlier papers [3, 4, 10, 12, 13, 33, 34, 11, 35, 36, 37, 23]. With this notation, the probability that x faults are newly detected by the test instance $t(i)$ is given by

$$P\Big\{X_i = x \mid C_{i-1} = c_{i-1}\Big\} = P\Big\{x|m, c_{i-1}, w(i)\Big\} = \frac{\binom{m-c_{i-1}}{x}\binom{c_{i-1}}{w(i)-x}}{\binom{m}{w(i)}} \tag{2}$$

where $0 \leq x \leq \min\{w(i), m - c_{i-1}\}$ and c_i is the realization of the random variable C_i. Since the above expression is the hypergeometric pmf [15], the sequential model based on Eq.(2) is called the HGDSRM.

From Eq.(2), the *conditional* mean number of newly detected faults at the i-th test instance and its variance are

$$\mathrm{E}[X_i \mid c_{i-1}] = \Big(\frac{m - c_{i-1}}{m}\Big)w(i) \tag{3}$$

and

$$\mathrm{Var}[X_i \mid c_{i-1}] = \frac{(m - c_{i-1})c_{i-1}w(i)}{m^2}\Big(\frac{m - w(i)}{m-1}\Big), \tag{4}$$

respectively. In the literature [10, 12, 35, 36], substituting $c_{i-1} = \sum_{k=1}^{i-1} x_k \approx \sum_{k=1}^{i-1} \mathrm{E}[X_k \mid c_{k-1}] \approx \mathrm{E}[C_{i-1}]$ into Eq.(3), the following recursive formula is obtained:

$$\mathrm{E}[C_i] = \mathrm{E}[C_{i-1}]\Big(1 - \frac{w(i)}{m}\Big) + w(i). \tag{5}$$

Jacoby and Tohma [10, 12] solved the above recursive equation as follows.

Proposition: The mean cumulative number of faults up to the i-th test instance is

$$\begin{aligned} \mathrm{E}[C_i] &= m\Big\{1 - \Pi_{j=1}^{i}\Big(1 - \frac{w(j)}{m}\Big)\Big\} = m\Big\{1 - \exp\Big[\sum_{j=1}^{i}\log\Big(1 - \frac{w(j)}{m}\Big)\Big]\Big\} \\ &\approx m\Big\{1 - \exp\Big[-\frac{1}{m}\int_0^i w(x)dx\Big]\Big\}. \end{aligned} \tag{6}$$

The above result is based on the heuristic derivation, but can be shown to be correct from the independence of the Bernoulli trials, as we show later. In the following section, we obtain further detailed mathematical results on the HGDSRM.

3 Further Results

Suppose that the probability $P_1\{x_1 \mid m, w(1)\}$ that x_1 faults are detected by the test instance $t(1)$ is the hypergeometric pmf. Let $P_2\{x_2|m, w(2)\}$ denote the probability that x_2 faults are detected newly at the second test instance, i.e. $i = 2$. Then, it is straightforward to obtain

$$P_2\Big\{x_2|m, w(2)\Big\} = \sum_{x_1=0}^{w(1)} P_1\Big\{x_1|m, w(1)\Big\}P\Big\{x_2|m, c_1, w(2)\Big\}. \tag{7}$$

From similar manipulation, we have

$$P_i\Big\{x_i|m, w(i)\Big\} = \sum_{x_{i-1}=0}^{w(i-1)} P_{i-1}\Big\{x_{i-1}|m, w(i-1)\Big\}P\Big\{x_i|m, c_{i-1}, w(i)\Big\} \tag{8}$$

where the right-hand side of Eq.(8) is due to (B-1) and (B-2). Of our interest is the derivation of the pmf $P_i\{x_i|m, w(i)\}$. Then, the problem is to solve the above recursive equation with the initial conditions:

$$P_1\Big\{x_1|m,w(1)\Big\} = \frac{\binom{m}{x_1}\binom{0}{w(1)-x_1}}{\binom{m}{w(1)}} = 1 \tag{9}$$

and

$$P_2\Big\{x_2|m,w(2)\Big\} = \frac{\binom{m-x_1}{x_2}\binom{x_1}{w(2)-x_2}}{\binom{m}{w(2)}}. \tag{10}$$

The following is the main result of this paper.

Theorem 1: For the initial number of software faults m, suppose that the sensitivity factor in i-th test instance $t(i)$ is defined by $w(i)$. Then, the probability that x_i faults are detected newly at i-th test instance, $\Pr\{X_i = x_i\} = P_i\{x_i \mid m, w(i)\}$, is

$$\sum_{x_2=0}^{w(2)}\sum_{x_3=0}^{w(3)}\cdots\sum_{x_{i-1}=0}^{w(i-1)}\left[\prod_{n=2}^{i-1}\frac{\binom{m-\sum_{k=1}^{n-1}x_k}{x_n}\binom{\sum_{k=1}^{n-1}x_k}{w(n)-x_n}}{\binom{m}{w(n)}}\right]\frac{\binom{m-\sum_{k=1}^{i-1}x_k}{x_i}\binom{\sum_{k=1}^{i-1}x_k}{w(i)-x_i}}{\binom{m}{w(i)}}. \tag{11}$$

For a proof, see the appendix. From Theorem 1, the probability distribution function and the *unconditional* mean are $\Pr(X_i \le x) = \sum_{x_i=0}^{x} P_i\{x_i|m, w(i)\}$ and $\mathrm{E}[X_i] = \sum_{x_i=0}^{w(i)} x_i P_i\{x_i \mid m, w(i)\}$, respectively. The following result is a direct application of Theorem 1.

Theorem 2: The mean number of newly detected faults at i-th test instance is

$$\mathrm{E}[X_i] = w(1)\prod_{j=2}^{i}\left(\frac{1}{w(j-1)} - \frac{1}{m}\right)w(j) = w(i)\prod_{j=2}^{i}\left(1 - \frac{w(j-1)}{m}\right) \quad (i \ge 2). \tag{12}$$

A proof is given in the appendix. From the above result, it can be verified that the unconditional mean of X_i is quite different from the conditional mean in Eq.(3). This mathematical result seems to be new and is consistent with the earlier result on the HGDSRM.

Corollary 1: The mean cumulative number of detected faults up to i-th test instance is

$$\mathrm{E}[C_i] = \mathrm{E}\left[\sum_{k=1}^{i} X_i\right] = m\left[1 - \prod_{j=1}^{i}\left(1 - \frac{w(j)}{m}\right)\right]. \tag{13}$$

This is the same as the Proposition in Section 3. In Theorem 2, note that $\mathrm{E}[X_i]$ is different from the conditional expectation, provided that the observations, $x_1, \cdots, x_{i-1}$, are given, that is,

$$\begin{aligned}
&\mathrm{E}[X_i = x_i|X_1 = x_1, X_2 = x_2, \cdots, X_{i-1} = x_{i-1}]\\
&= \sum_{x_i=0}^{w(i)} x_i \frac{\binom{m-\sum_{k=1}^{i-1}x_k}{x_i}\binom{\sum_{k=1}^{i-1}x_k}{w(i)-x_i}}{\binom{m}{w(i)}}\\
&= w(i) - \frac{w(i)}{m}\sum_{k=1}^{i-1} x_k
\end{aligned} \tag{14}$$

is given by Eq.(3).

Next, consider the variance of X_i. From some algebraic manipulations, we can derive the following results.

Theorem 3: The variance of the number of newly detected faults at i-th test instance is

$$\begin{aligned}\mathrm{Var}[X_i] &= \frac{w(i)(m-w(i))}{m-1}\left[\prod_{r=1}^{i-1}\left(1-\frac{w(r)}{m}\right)\right]\left[1-\prod_{r=1}^{i-1}\left(1-\frac{w(r)}{m}\right)\right] \\ &+\frac{w(i)(w(i)-1)}{m(m-1)^2}\sum_{n=2}^{i-1} w(n)(m-w(n))\left[\prod_{r=1}^{n-1}\left(1-\frac{w(r)}{m}\right)\right] \\ &\times\left[1-\prod_{r=1}^{n-1}\left(1-\frac{w(r)}{m}\right)\right]\prod_{j=n+1}^{i-1}\left(\frac{w(j)(w(j)-1)}{m(m-1)}+1\right). \end{aligned} \tag{15}$$

Theorem 4: The variance of the cumulative number of detected faults up to i-th test instance is

$$\begin{aligned}\mathrm{Var}\,[C_i] &= \sum_{n=2}^{i}\frac{w(n)(m-w(n))}{(m-1)}\left[\prod_{r=1}^{n-1}\left(1-\frac{w(r)}{m}\right)\right] \\ &\times\left[1-\prod_{r=1}^{n-1}\left(1-\frac{w(r)}{m}\right)\right]\prod_{j=n+1}^{i}\left(\frac{w(j)(w(j)-1)}{m(m-1)}+1\right). \end{aligned} \tag{16}$$

The proofs are omitted for brevity. From Theorem 3 and Theorem 4, two random variables, X_i and C_i, are possible to be evaluated. If the NHPPs are assumed as candidates of the SRGM, the variance of C_i is not attractive as a dependability measure, since it is equivalent to the mean value. On the other hand, the variances in Theorem 3 and Theorem 4 will be usuful to assess the fluctuation of the prediction values, $\mathrm{E}[X_i]$ and $\mathrm{E}[C_i]$, to the data x_i and $c_i = \sum_{k=1}^{i} x_k$ using the HGDSRMs.

Finally, consider the software reliability for the HGDSRM. As mentioned before, the software reliability function in the HGDSRM has not been defined in the past literature. We define it as the probability that no new software faults are detected up to an arbitrary i-th test instance. The software reliability at $i = 1$ is $P_1\{0 \mid m, w(1)\} = 0$, by the assumption $X_1 = w(1)$ with probability one.

Theorem 5: Suppose that j (> 1) test instances are experienced before. Then, the software reliability at i-th test instance ($i > j$) is

$$\begin{aligned} &P\Big\{\sum_{n=j+1}^{i} X_n = 0 \mid m, w(i), \sum_{n=1}^{j} x_n\Big\} \\ &= \prod_{n=j+1}^{i}\frac{\binom{m-\sum_{n=1}^{j}x_n}{0}\binom{\sum_{n=1}^{j}x_n}{w(n)}}{\binom{m}{w(n)}} \end{aligned} \tag{17}$$

and has the product form of the non-identical hyper-geometric distributions.

The result is obvious from Theorem 1. The main reason that in the past literature the software reliability could not be defined in the consistence way is that all results on the HGDSRMs were based on only the mean value $\mathrm{E}[C_i]$. In other words, the mathematical results derived in this section are important to characterize the HGDSRMs in terms of the probabilistic argument.

References

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

[2] K. Goševa-Postojanova and K. S. Trivedi. Architecture based approach to reliability assessment of software system. *Performance Evaluation*, (in press).

[3] R. H. Hou, S. Y. Kuo and Y. P. Chang. Applying various learning curves to hyper-geometric distribution software reliability growth model. In *Proc. 5^{th} Int'l Symp. Software Reli. Eng.*, pages 7–16, 1994. IEEE CS Press.

[4] R. H. Hou, S. Y. Kuo and Y. P. Chang. Hyper-geometric distribution software reliability growth model with imperfect debugging. In *Proc. 6^{th} Int'l. Symp. Software Reli. Eng.*, pages 195–200, Toulouse, France, October 1995. IEEE CS Press.

[5] R. H. Hou, S. Y. Kuo and Y. P. Chang. Efficient allocation of testing resources for software module testing based on the hyper-geometric distribution software reliability growth model. In *Proc. 7^{th} Int'l. Symp. Software Reli. Eng.*, pages 289–298, White Plains, NY, October 1996. IEEE CS Press.

[6] R. H. Hou, I. Y. Chen, Y. P. Chang and S. Y. Kuo. Optimal release policies for hypergeometric distribution software reliability growth model with scheduled delivery time. In *Proc. Asia-Pacific Software Eng. Conf.*, pages 445–452, Tokyo, Japan, December 1996.

[7] R. H. Hou, S. Y. Kuo and Y. P. Chang. Needed resources for software module test, using the hyper-geometric software reliability growth model. *IEEE Trans. Reliab.*, 45 (4): 541–549, (December, 1996).

[8] R. H. Hou, S. Y. Kuo and Y. P. Chang. Optimal release policy for hyper-geometric distribution software-reliability growth model. *IEEE Trans. Reliab.*, 45 (4): 646–651, (December, 1996).

[9] R. H. Hou, S. Y. Kuo and Y. P. Chang. Optimal release times for software systems with scheduled delivery time based on the HGDM. *IEEE Trans. Computers*, 46 (2): 216–221, (February, 1997).

[10] R. Jacoby and Y. Tohma. The hyper-geometric distribution software reliability growth model (HGDM): precise formulation and applicability. In *Proc. COMPSAC'90*, pages 13–19, Chicago, IL, October 1990. IEEE CS Press.

[11] R. Jacoby and Y. Tohma. Parameter value computation by least square method and evaluation of software availability and reliability at service-operation by the hyper-geometric distribution software reliability growth model (HGDM). In *Proc. 13^{th} Int'l. Conf. Software Eng.*, pages 226–237, Austin, TX, May 1991. ACM/IEEE CS Press.

[12] R. Jacoby and Y. Tohma. Precise formulation and applicability of a software reliability growth model based on hyper-geometric distribution. *J. Information Processing*, 14 (2): 192–203, (February, 1991).

[13] R. Jacoby and K. Masuzawa. Test coverage dependent software reliability estimation by the HGD model. In *Proc. 3^{rd} Int'l. Symp. Software Reli. Eng.*, pages 193–204, Research Triangle Park, NC, October 1992. IEEE CS Press.

[14] Z. Jelinski and P. B. Moranda. Software reliability research. In *Statistical Computer Performance Evaluation*, W. Freiberger (ed.), pages 465–484, Academic Press, New York, NY, (1972).

[15] N. L. Johnson, S. Kotz and A. W. Kemp. *Univariate Discrete Distributions*, Second Edition. John Wiley & Sons, Pronto, Canada, (1992).

[16] J. C. Laprie. Dependability evaluation of software systems in operation. *IEEE Trans. Software Eng.*, 10(6): 701–714, (November, 1984).

[17] J. C. Laprie and K. Kanoun. X-ware reliability and availability modeling. *IEEE Trans. Software Eng.*, 18(2): 130–147, (February, 1992).

[18] J. Ledoux. Availability modeling of modular software. *IEEE Trans. Reliab.*, 48(2): 159–168, (June, 1999).

[19] B. Littlewood. A software reliability model for systems with Markov structure. *J. Royal Statist. Soc. Series C*, 24(2): 172–177, (April, 1975).

[20] B. Littlewood. Software reliability model for modular program structure. *IEEE Trans. Reliab.*, 28(3): 241–246, (August, 1979).

[21] B. Littlewood. Theories of software reliability: how good are they and how can they be improved? *IEEE Trans. Software Eng.*, 6(5): 489–500, (September, 1980).

[22] M. R. Lyu (ed). *Handbook of Software Reliability Engineering.* McGraw-Hill, New York, NY, (1996).

[23] T. Minohara and Y. Tohma. Parameter estimation of hyper-geometric distribution software reliability growth model by genetic algorithm. In *Proc. 6^{th} Int'l. Symp. Software Reli. Eng.*, pages 324–329, Toulouse, France, October 1995. IEEE CS Press.

[24] J. D. Musa. A theory of software reliability and its application. *IEEE Trans. Software Eng.*, 1(3): 312–327, (September, 1975).

[25] J. D. Musa. *Software Reliability Engineering.* McGraw-Hill, New York, NY, (1999).

[26] J. D. Musa, A. Iannino and K. Okumoto. *Software Reliability, Measurement, Prediction, Application.* McGraw-Hill, New York, NY, (1987).

[27] H. Pham. *Software Reliability.* Springer-Verlag, Singapore, (1999).

[28] P. Piwowarski, M. Ohba and J. Caruso. Coverage measurement experience during function test. In *Proc. 15^{th} Int'l. Conf. Software Eng.*, pages 287–301, Baltimore, MD, May, 1993. ACM/IEEE CS Press.

[29] A. T. Rivers and M. A. Vouk. Resource-constrained non-operational testing of software. In *Proc. 9^{th} Int'l. Symp. Software Reli. Eng.*, pages 154–163, Paderborn, Germany, November 1998. IEEE CS Press.

[30] G. J. Schick and R. Wolverton. An analysis of competing software reliability models. *IEEE Trans. Software Eng.*, 4(2): 104–120, (March, 1978).

[31] M. L. Shooman. *Software Engineering.* McGraw-Hill, New York, NY, (1983).

[32] N. D. Singpurwalla and S. P. Wilson. *Statistical Methods in Software Engineering.* Springer-Verlag, New York, NY, (1997).

[33] Y. Tohma, K. Tokunaga, S. Nagase and Y. Murata. Structural approach to the estimation of the number of residual software faults based on the hyper-geometric distribution. *IEEE Trans. Software Eng.*, 15 (3): 345–355, (March, 1989).

[34] Y. Tohma, R. Jacoby, Y. Murata and M. Yamamoto. Hyper-geometric distribution model to estimate the number of residual software faults. In *Proc. COMPSAC'89*, pages 610–617, Orlando, FL, September 1989. IEEE CS Press.

[35] Y. Tohma, H. Yamano, M. Ohba and R. Jacoby. The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model. *IEEE Trans. Software Eng.*, 17 (5): 483–489, (May, 1991).

[36] Y. Tohma, H. Yamano, M. Ohba and R. Jacoby. Parameter estimation of the hyper-geometric distribution model for real test/debug data. In *Proc. 2^{nd} Int'l. Symp. Software Reli. Eng.*, pages 28–34, Austin, TX, May 1991. IEEE CS Press.

[37] Y. Tohma and Y. Matsunaga. Application of hyper-geometric distribution model to the hardware debugging process. *Computer Syst. Sci. & Eng.*, 9 (1): 25–30, (January, 1994).

[38] M. Xie. *Software Reliability Modeling.* World Scientific, Singapore, (1991).

[39] H. Xizi. The hypergeometric distribution model for predicting the reliability of softwares. *Microelectron. Reliab.*, 24 (1): 11–20, (January, 1984).

[40] H. Xizi. The new applications of the hypergeometric distribution model on software reliability. *Microelectron. Reliab.*, 25 (4): 713–714, (April, 1985).

[41] S. Yamada, M. Ohba and S. Osaki. S-shaped reliability growth modeling for software error detection. *IEEE Trans. Reliab.*, 32(5): 475–485, (December, 1983).