TELKOMNIKA, Vol. 11, No. 8, August 2013, pp. 4399~4406
e-ISSN: 2087-278X

Determining the Minimal Software Reliability Test Effort by Stratified Sampling

Li Qiuying*[1,2], Luo Lei[3]
[1,3]School of Reliability and Systems Engineering, Beijing University of Aeronautics and Astronautics
[2]Science & Technology on Reliability & Environmental Engineering Laboratory
Beijing, China,Tel.+86-10-82339169
Corresponding author, e-mail: li_qiuying@buaa.edu.cn[1], leicherry@163.com[3]

Abstract

Software reliability testing provided the necessary guarantee for improving software reliability level and estimation. The traditional method for generating software reliability test cases could be seen as a simple random sampling, which was randomly choosen according to the operational profile defined by Musa. The stratified sampling was well known as a complex but more accurate sampling approach which could make the sampling results more accurate and less variance by dividing the population into several subsets and randomly sampling from each subset respectively. First, this paper introduced the traditional approach to determine the number of test cases in the discrete software reliability demonstration testing. Second, the background of the stratified sampling and necessary condition for the minimal test effort based on the stratified sampling were put forward. Third, the new method's principle and details were analyzed and a sample was given to show the method's effectiveness. Finally, the future work was discussed.

Keywords*: minimal test effort, software reliability, discrete software, stratified sampling*

1. Introduction

Software reliability engineering is now widely accepted and applied in lots of fields. Software reliability testing which can be seen as a key role of software reliability engineering provides the necessary guarantee and reference for improving software reliability level and estimation.

Software reliability testing is a random testing which uses the operational profile for accurately simulating the actual operational condition of software by customers. The more test cases are generated, the more easily the probability distribution is described and the more sufficiently the input space is covered, but the test cost and time may both become large as the growth of the test cases. Therefore, the approach for determining the set with the smallest test cases of software reliability testing is very significant. This is because when all test case sets satisfy the requirement of software reliability testing, the set with the smallest test cases is the best in all the test case sets.

The software reliability testing approach based on operational profile [1] and the statistical testing approach based on Markov usage chain [2] have the same characteristics that testing the software according to the actual operational condition. However, the former testing approach based on the operational profile has the problem that the adequacy of test data cannot be measured [3]. The external representation of the adequacy measurement is how to determine the minimum number of test cases (i.e. minimal test cases). To solve the problem, this paper proposes an approach for determining the minimal test cases by using the stratified sampling.

The stratified sampling has been widely studied and used since 1950s. The optimal allocation of the sampling size of the stratified random sampling was introduced in [4]. The 'proportional sampling of element' and 'the optimal allocation' of the stratified sampling were also introduced in [5]. There are many books with respect to the stratified sampling were published by the statistical experts of China, such as Feng Shiyong [6], Jin Yongjin [7] and Du Zifang [8]. However, so far as we know, the stratified sampling has not been applied into the field of software reliability engineering.

Received March 4, 2013; Revised May 13, 2013; Accepted May 22, 2013

2. Generation of Test Cases

Software reliability testing approach based on operational profile proposed by Musa [1] is a random testing process which generates test cases according to the operational profile by random sampling.

According to the operational profile [9], the input domain D of software S can be divided into m disjoint sub-domains: D_1, D_2, …, D_m. Let p_i represent the occurrence probability of the corresponding sub-domain D_i, and θ_i represent the failure probability of D_i. Then the operational profile can be denoted as OP={(D_i, p_i), i=1,2,…,m}. Let n_i represent the number of test cases sampling from D_i and n represent the total number of test cases. Then we have $n=\sum_{i=1}^{m} n_i$ and $\sum_{i=1}^{m} p_i = 1$. The sequence $\{S_i\}$ can be given by $S_j = \sum_{i=1}^{j} p_i$, where j=1,2,…,m. Let S_0=0, S_1=p_1, S_m=1 and $S_j - S_{j-1} = p_j$. Then the steps for generating test cases according to operational profile can be shown as follows:

(i) Sampling. Given a random number $\eta \in (0,1)$, if $S_{j-1} < \eta \le S_j$, then this random number η is corresponding to p_j and the corresponding input sub-domain of the sampling operation is D_j.

(ii) Determining the value of each input variable of sampling operation. Because the value style of the input variable can be discrete or continuous, the sampling approach for these two styles of input variable should be considered respectively. The continuous input variable should be sampled in its value space according to the probability density function and the discrete input variable should be sampled in its value space according to the probability distribution.

(iii) A test case can be generated by the sampling process according to the above two steps.

(iv) Repeating the above steps until generating the required number of test cases. Theoretically, the more the sampling size (i.e. the number of test cases) is, the more similar the sampling statistical characteristics of test cases is with the one of the actual usage condition and the reliability estimation results are more accurate. However, the cost and time are also improved.

3. Test Cases of Discrete Software Reliability Demonstration Testing

Lots of software systems, such as flight control system and task planning system, will be operated continuously during the mission period and thus be called as the continuous software. Generally, the reliability metrics of the continuous software are MTBF/MTTF or failure rate, because the customers will pay much attention to the performance that the systems don't fail during the continuous operational period. However, for many cases, discrete time-domain software systems should be considered in many critical-safety systems [10, 11], such as control system of missile and emergency switch system of nuclear power station. For the discrete software systems, the customers usually pay attention to the success probability of single-usage. Obviously, these discrete software systems will select the success rate (i.e. the success probability of single-usage) as their reliability metric instead of the time-dependent function. This paper will select the discrete software as the studied object.

Software reliability demonstration testing which is used for validating whether the released software achieves the requirement of the quantitative reliability level, determines that the software should be accepted or rejected according to the demonstration testing results. Software reliability demonstration testing can be classified into the fixed-duration testing and the sequential testing.

The fixed-duration reliability demonstration testing first calculates the required number of test cases according to the required quantitative reliability level, and then determines whether the software passes the demonstration testing according to the ratio between the actual failure number during the whole testing process and the allowable failure number. Now there are several fixed-duration reliability demonstration testing methods, such as the hypothesis testing [12] and the Bayesian method [13].

The sequential reliability demonstration testing first calculates the required testing duration according to the required quantitative reliability level, and then it determines whether the software passes the testing according to the operation result of each test case at any time.

Now there are several sequential reliability demonstration testing methods, such as probability ratio sequential testing [14] and single risk sequential testing [15].

According to the above, we find that the sequential testing can't determine the required number of test cases before testing, but the fixed-duration testing can determine the required number of test cases before testing according to the required reliability and confidence level. For example, if we use the hypothesis testing, the number of the required test cases is 4603 at least when the confidence level is 99% and the required failure probability is less than 0.001.

It seems that the hypothesis testing can be used to determine the minimal test cases for reliability demonstration testing. However, this method is only based on the hypothesis testing theory but neglecting the distribution of test cases. For example, if the distribution of the above 4603 test cases doesn't match the distribution of the sub-domains of operational profile well, we can't believe that the reliability testing achieves the required level (i.e. the failure probability is less than 0.001) with the 99% confidence even though the 4603 testing cases all pass the demonstration testing without any failures. In other words, the failure probability which is less than 0.001 can't reflect the actual reliability level of software. Thus, this paper will determine the minimal test cases based on the sampling theory according to the sampling characteristics.

4. Test Cases of Discrete Software Reliability Demonstration Testing

4.1. Concepts

There are several factors which have the influence on the sampling accuracy, such as the size of samples, the population size and the population variance. If the difference between each unit in the population is large, using the simple random sampling may result in the large variance. Although the variance of the population is objective and can't be changed, if the units in the population can be divided into several sub-populations, and the units in one sub-population are similar that makes the variance in each sub-population become smaller, only a handful of sample units from one sub-population are needed for describing the characteristics of this sub-population as well as the estimation accuracy of all population will be improved. The above analysis is just the principle of the stratified sampling. Before the stratified sampling, N units in the population are divided into L different and independent sub-populations which also be called as the layers. The sizes of these layers are N_1, N_2,…, N_L. That is to say, the population is composed of these layers ($N = \sum_{h=1}^{L} N_h$). Then the random sampling in each layer independently is just the stratified random sampling. In the stratified sampling, the issue that how many samples are allocated to each layer with a certain population should be determined. There are three allocation approaches, i.e. the proportion allocation, the optimal allocation and the Neymanm allocation [16]. The characteristics of the operational profile provide the required layers in the stratified sampling.

4.2. Notations

$h = 1, 2, ..., L$;

N_h: The unit number in the *h* layer;

n_h: The sample unit number in the *h* layer;

$N = \sum_{h=1}^{L} N_h$, $n = \sum_{h=1}^{L} n_h$;

Y_{hi}: The observed value of the *i* unit in the *h* layer;

y_{hi}: The observed value of the *i* sample unit in the *h* layer;

$W_h = \frac{N_h}{N}$:The weight of the *h* layer;

$\overline{Y_h} = \frac{1}{N_h} \sum_{i=1}^{N_h} Y_{hi}$:The average value of the *h* layer;

$\overline{y_h} = \frac{1}{N_h}\sum_{i=1}^{N_h} y_{hi}$: The average sample value of the *h* layer;

$Y_h = N_h \overline{Y_h} = \sum_{i=1}^{N_h} Y_{hi}$: The total value of the *h* layer;

$y_h = n_h \overline{y_h} = \sum_{i=1}^{n_h} y_{hi}$: The total sample value of the *h* layer;

$S_h^2 = \frac{1}{N_h - 1}\sum_{i=1}^{N_h} (Y_{hi} - \overline{Y_h})^2$: The variance of the *h* layer;

$s_h^2 = \frac{1}{n_h - 1}\sum_{i=1}^{n_h} (y_{hi} - \overline{y_h})^2$: The sample variance of the *h* layer.

4.3. Comparison of the Sampling Accuracy

Generally, the sampling accuracy of the stratified sampling is better than the simple random sampling. In other words, the variance of the estimated values of the stratified sampling is smaller than the simple random sampling. We will compare the sampling accuracy of the stratified sampling with the proportion allocation with the simple random sampling as follows.

The variance of the estimated values of the simple random sampling is shown as follows.

$$V_{srs} = \frac{1-f}{n} S^2 \quad (1)$$

The variance of the estimated values of the stratified sampling is shown as follows.

$$V_{prop} = \frac{1-f}{n}\sum_{h=1}^{L} W_h^2 S_h^2 \quad (2)$$

That is, we have:

$$V_{srs} \approx V_{prop} + \frac{1-f}{n}\sum_{h=1}^{L} W_h (\overline{Y_h} - \overline{Y})^2 \quad (3)$$

According to (3), we have that $V_{srs} \geq V_{prop}$. Thus it shows that the variance of the stratified sampling with the proportion allocation is smaller than that of the simple random sampling, i.e. the accuracy of the stratified sampling is better than that of the simple random sampling. The difference between the average values in each layer is larger, the stratified result is better. However, the sampling accuracy of the stratified sampling is also related to the allocation of the sample size and the variance of each layer.

5.Determining the Minimal Cases by the Stratified Sampling with Proportion Allocation

In the stratified sampling, the estimation of the average value $\overline{Y}$ of the population can be given as the average of the $\overline{Y_h}$ in each layer with the layer weight W_h and shown as follows.

$$\overline{\hat{Y}_{st}} = \sum_{h=1}^{L} W_h \overline{\hat{Y}_h} = \frac{1}{N}\sum_{h=1}^{L} N_h \overline{\hat{Y}_h} \quad (4)$$

If the stratified random samples are obtained, the sample estimated value of the average value $\overline{Y}$ of the population can be shown as follows.

$$\overline{y_{st}} = \sum_{h=1}^{L} W_h \overline{y_h} = \frac{1}{N} \sum_{h=1}^{L} N_h \overline{y_h} \tag{5}$$

In the stratified sampling, if the estimation of each layer is unbiased, the estimation of the value of the population is also unbiased. Thus, various layers can select different sampling approaches, only if the corresponding estimated value is unbiased, the estimation of the value of the population is also unbiased. If $\overline{\hat{Y}_h}$ is the unbiased estimation of $\overline{Y_h}$ (h=1,2,…,L), $\overline{\hat{Y}_{st}}$ is the unbiased estimation of $\overline{\hat{Y}}$. The variance of $\overline{\hat{Y}_{st}}$ is shown as follows.

$$V(\overline{\hat{Y}_{st}}) = \sum_{h=1}^{L} W_h^2 V(\overline{\hat{Y}_h}) \tag{6}$$

Then the variance of the sample average $\overline{y_{st}}$ is given as follows.

$$V(\overline{y_{st}}) = \sum_{h=1}^{L} W_h^2 V(\overline{y_h}) = \sum_{h=1}^{L} W_h^2 \frac{1 - n_h / N_h}{n_h} S_h^2 \tag{7}$$

Assume $n_h = nW_h$, so we can obtain the following equation based on (7) with the given variance.

$$V(\overline{y_{st}}) = \sum_{h=1}^{L} W_h^2 \frac{1 - n_h / N_h}{n_h} S_h^2 = \sum_{h=1}^{L} \frac{W_h^2}{n_h} S_h^2 - \sum_{h=1}^{L} \frac{W_h^2}{N_h} S_h^2 = \frac{1}{n} \sum_{h=1}^{L} \frac{W_h^2}{w_h} S_h^2 - \frac{1}{N} \sum_{h=1}^{L} \frac{W_h^2}{W_h} S_h^2 \tag{8}$$

Finally, we can get the general expression for determining the minimal test cases as follows.

$$n = \frac{\sum \frac{W_h^2 S_h^2}{w_h}}{V + \sum \frac{W_h S_h^2}{N}} \tag{9}$$

If the estimation accuracy is given as the form of the error limit, we have $V = (\frac{\Delta}{t})^2 = (\frac{\gamma \overline{Y}}{t})^2$, where Δ is the absolute error limit, γ is the relative error limit, t is the bilateral critical point of the standard normal distribution, $\overline{Y}$ is the average value of the population.

Then apply the stratified sampling into the actual usage condition of the software s which can be abstracted as Table 1.

Table 1. The Abstraction of the usage of Software S

D	D_1	D_2	…	D_h	…	D_m
p	p_1	p_2	…	p_h	…	p_m

Where $D_1, D_2, \ldots, D_m$ is the division of the input space of the software s according to the actual usage of customers, $p_1, p_2, \ldots p_m$ is the occurrence probability of the corresponding input sub-domain. Then the set T of test cases can be regarded as a group of samples which was

obtained by sampling *n* times according to the above distribution [7]. If the samples fail one time in the testing process, the observed value of the samples is 1, otherwise is 0. Assume that the sampling times of *T* in $D_1, D_2, \ldots, D_m$ are $n_1, n_2, \ldots, n_m$, where $n = \sum_{h=1}^{m} n_h$ and $h = 1, 2, ..., m$.

Then we can translate the problem that determining the minimal test cases for software reliability testing into the problem that determining the size of samples in the stratified sampling with the proportion allocation. According to the principle of the stratified sampling, we can regard the input space $D_1, D_2, \ldots, D_m$ as *m* layers, and the occurrence probability of the corresponding domains $p_1, p_2, \ldots p_m$ as the layer weight W_h, that is:

$$W_h = p_h \tag{10}$$

Due to the proportion allocation, the ratio w_h between the samples allocated into each layer and the samples of the population is the same with the layer weight W_h, that is:

$$w_h = p_h \tag{11}$$

Incorporating (10) and (11) into (9), then we have:

$$n = \frac{\sum p_h S_h^{\,2}}{V + \sum \frac{p_h S_h^{\,2}}{N}} \tag{12}$$

Because the actual size of software reliability test set can be infinity, i.e. $N \to \infty$, (12) can be rewritten as follows.

$$n = \frac{\sum p_i S_i^{\,2}}{V} \tag{13}$$

Although there are many parameters in the unknown variable *V*, such as the absolutely error limit Δ, the relative error limit γ and the bilateral critical point of the standard normal distribution *t*, in the calculation process, only the relative error limit γ should be considered. For example, if the confidence level is 95%, the relative error limit is less than 10%, and the corresponding *t* is 1.96, then replacing the average value by the sample average value, and we have:

$$V = (\frac{\gamma \overline{Y}}{t})^2 = (\frac{10\% \overline{y}}{1.96})^2 \tag{14}$$

Similarly, in the calculation process, only the absolute error limit Δ should be considered. For example, if the confidence level is 95%, the absolute error limit is less than 5%, and the corresponding *t* is 1.96, then we have:

$$V = (\frac{\Delta}{t})^2 = (\frac{5\%}{1.96}) = 0.000651 \tag{15}$$

6. Case Study

The operational profile of software S' is shown as the following table.

Table 2. The Operational Profile of Software S

D_k	D_1	D_2	D_3	D_4
p_k	1/2	1/3	1/10	1/15

For calculating the variance S_k^2 (k=1,2,3,4), assume that 10 samples are obtained from each sub-domain and are operated for validating whether the samples fail, then we find that the 4th and 8th sample of D_1 failed, the 5th sample of D_2 failed, all samples of D_3 didn't fail, and the 1st, 5th and 10th sample of D_4 failed. Then we have:

$$S_1^2 = \frac{1}{9}[(0.2)^2 * 8 + (0.2-1)^2 * 2] = 0.178$$

$$S_2^2 = \frac{1}{9}[(0.1)^2 * 9 + (0.1-1)^2] = 0.1$$

$$S_3^2 = 0$$

$$S_4^2 = \frac{1}{9}[(0.3-0)^2 * 7 + (0.3-1)^2 * 3] = 0.233$$

The variance discussed above is only related to the failure probability of the discrete software.

Set the absolutely error lime is less than 5%, the confidence level is 95% and the corresponding t is 2.58, then we have:

$$V = (\frac{\Delta}{t})^2 = (\frac{5\%}{1.96})^2$$

Taking the above result into (13), we have:

$$n = \frac{\sum p_i S_i^2}{V} = \frac{0.5*0.178 + 0.3333*0.1 + 0 + 0.0667*0.233}{(\frac{5\%}{1.96})^2} = 212$$

It means that 212 test cases are required at least for the reliability testing of this software when the confidence level is 95% and the absolute error limit is less than 5%. Assume the absolute error limit is less than 5%, the confidence level is 99% and the corresponding t is 2.58, then we have:

$$V = (\frac{\Delta}{t})^2 = (\frac{1\%}{2.58})^2$$

Taking the above result into (13), we have:

$$n = \frac{\sum p_i S_i^2}{V} = \frac{0.5*0.178 + 0.3333*0.1 + 0 + 0.0667*0.233}{(\frac{5\%}{2.58})^2} = 367$$

It means that 367 test cases are required at least for the reliability testing of this software when the confidence level is 99% and the absolute error limit is less than 5%.

7. Conclusion

The sampling accuracy of the stratified sampling is generally better than the simple random sampling, therefore an approach based on the stratified sampling for determining the minimal test case number was proposed for the discrete software reliability testing. This approach is significant for the theory research and engineering application because it improves the existing software reliability demonstration testing and solves the problem that the determination conclusions of the software reliability demonstration testing are not believable during any situations. This approach not only provides the guidance for the selection of the test cases in the test cases generation process, but also decreased the test cost without reducing the dependability of the result of the software reliability testing. Of course, the approach for determining the minimal test case number based on the stratified sampling is a new attempt on software reliability testing adequacy and also has some disadvantage in details and should be improved in the future work.

References

[1] JD Musa. Operational profiles in software reliability engineering. *IEEE Software*. 1993; 10(2): 14-32.
[2] JA Whittaker, G Thomason. A markov chain model for statistical software testing. *IEEE Transactions on Software Engineering*. 1994; 20(10): 812-824.
[3] QY Li, MY Lu, L Ruan. Theoretical research on software reliability testing adequacy. *Journal of Beijing University of Aeronautics and Astronautics*. 2003; 29(4): 312-316 (in Chinese).
[4] WG Cochran. *Sampling technique*. Chinese statistic press, Beijing. 1985 (in Chinese).
[5] L Kish. *Sampling technique.* Chinese statistic press, Beijing. 1997 (in Chinese).
[6] SY Feng, JX Ni, GH Zou. *The theory and approach of sampling investigation.* Chinese statistic press, Beijing. 1998 (in Chinese).
[7] YJ Jin, Y Jiang, XY Li. *Sampling technique.* Chinese People University press, Beijing. 2003 (in Chinese).
[8] ZF Du. *Sampling technique and its application.* Tsinghua University press, Beijing. 2005 (in Chinese).
[9] JD Musa. *Software reliability engineering*. McGraw-Hill Book Company, New York. 1996.
[10] K Ca. Towards a conceptual framework of software run reliability modeling. *Information Sciences*. 2000; 126: 137-163.
[11] B Littlewood, L Strigini Assessment of ultra-high dependability for software-based systems, *Communications of ACM.* 1993; 36(11): 69-80.
[12] DL Parnas, AJ van Schouwen, SP Kwan\. Evaluation of safety-critical software. *Communications of ACM.* 1990; 33(6): 636-648.
[13] B Littlewood, W David. Some conservative stopping rules for the operational testing of safety critical software. *IEEE Transactions on software Engineering*. 1997; 23(11): 673-683.
[14] MIL-HDBK2781 A, Handbook for reliability test methods, plans and environments for engineering, development, qualification, and production. 1996.
[15] O Tal, C McCollin, A Bendell. Reliability demonstration for safety-critical systems. *IEEE Transactions on Reliability*. 2001; 50(2): 194-203.
[16] JC Li. *Application of sampling technique.* Science press, Beijing. 2006 (in Chinese).