ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 9, Issue, 05, pp.51113-51116, May, 2017

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

IMPROVEMENT OF SOFTWARE RELIABILITY ASSESSMENT USING ADAPTIVE TESTING STRATEGY

Dr. P. Maragathavalli and *Saranya. A.

Department of Information Technology, Pondicherry Engineering College, Puducherry, India

ARTICLE INFO

Article History:
Received 24th February, 2017
Received in revised form
21st March, 2017
Accepted 04th April, 2017
Published online 31st May, 2017

Key words:

Adaptive testing, Software testing, Software reliability, Adaptive Testing with Gradient Descent method.

ABSTRACT

Software testing has ever remained a brazen out particularly when testing is done with objective in enhancing the reliability. Adaptive Testing (AT) is an online testing strategy, which can be adopted to decrease the variance of software reliability estimator. In order to reduce the computational overhead of decision-making, the implemented AT strategy in practice deviates from its theoretical design that guarantees AT's local optimality. Adaptive testing is increasing the testing in a predictable way by reducing the number of faults. There is a need to enhance the reliability by conveying probabilistic priorities to testing mechanism, which is done through software under testing with testing profile. AT strategy named Adaptive Testing with Gradient Descent method (AT-GD) is proposed. AT-GD, a locally optimal testing strategy, converges to the globally optimal solution as the assessment process proceeds. Simulation and experiments are set up to validate AT-GD's effectiveness and efficiency.

Citation: Dr. P. Maragathavalli and Saranya. A. 2017. "Improvement of software reliability assessment using adaptive testing strategy", *International Journal of Current Research*, 9, (05), 51113-51116.

INTRODUCTION

Mission-Critical systems play an important role in daily life, such as avionics and flight safety, financial services, and so on. In such systems, more and more jobs are proficient by computers and software. One difficulty is that, some software systems might not be as trustworthy as hardware systems, which can probably lead to business or property loss. Usually, high reliability is required in mission-critical systems, e.g., less than $1x10^9$ probability of failure per hour in a flight controls system. Thus, software reliability assessment is a critical and hard problem in such systems, since an estimate that significantly deviate from the true value of reliability might lead to overconfidence in the product, and unexpected loss after deploying the system. In the theory of software reliability assessment, estimating the reliability value as well as the corresponding confidence interval is usually conducted based on the testing data collected during the assessment process. Software reliability testing is a statistical measures for the operation of software system without failure occurring. Reliability can also be measured by probability of software failure occurrence. Two terms related to SRT i) Fault: A defect in the software ii) Failure: A derivation of programs observed behavior from the required behavior. Software reliability improvement by detecting and removing defects from software under testing, or for software assessment by freezing the code of the SUT without removing defects during testing. The technique such as mutation used to assess the defect the capability of test cases of software under test. The technique used for test cases generation is random testing, control Markov chain. Adaptive testing strategy (Junpeng *et al*., 2014) is a new software testing technique which result of feedback and adaptive control used at software under test. Adaptive control is treated as the software testing counterpart there are two feedback loops in the adaptive testing strategy i) feedback loop constitutes the software under test, the database and the testing strategy in which the history is matinees the test cases and also used to generate the next cases by a given testing policy or test data adequacy criterion. ii) Feedback loop constitutes the software under testing, the parameter estimation scheme and the testing strategy, in which the history of test data is also used to improve or modify the underlying test cases policy. For software reliability assessment, there are mainly two branches: the continuous-time base and the discrete-time base. The earlier focuses on the reliability behavior measured in terms of CPU execution time and so on. The continuous-time assumption is suitable for a extensive scope of systems; there are many systems that do not essentially convince this assumption. For example, reliability performance of the software system for rocket control should be deliberate in terms of how many rockets are successfully instigate, rather than of how long a rocket flies without failures. For such systems, the time base of reliability

Corresponding author: **Saranya. A.**
Department of Information Technology, Pondicherry Engineering College, Puducherry, India.

measurement is essentially discrete rather than continuous. In fact, the reliability assessment for many mission-critical systems can be viewed as a discrete-time problem, such as, GPS system on an aircraft. GPS system can provide the position of an aircraft for landing and departure procedures.

In many mission-critical systems, the reliability is required to be as high as probable. On the other hand, the reliability estimation process can be costly, and thus, only limited testing resources are presented. This will guide to few failures being experimental after the assessment, which will increase the difficulty of delivering a highly accurate estimate. Therefore, both high effectiveness and efficiency are required in the reliability assessment. Under such condition, a reliability estimator with minimum variance is preferred. Thus, the main target of this study is to minimize the variation of reliability estimator for such structure in discrete time domain.Test case prioritization techniques schedule test cases so that those with the highest priority, according to some standard, are accomplish earlier in the testing process than lower priority test cases. An advantage of prioritization techniques is that, unlike many other techniques for assisting deterioration testing, they do not discard test cases are effective with respect to the time required for the two extensive components specifically, the test case selection and code coverage. On applying Adaptive Random Testing it contributes towards the improvement of techniques and that is made possible via the combination of two or more techniques. Number of faults, testing time, number of test cases, subject program etc. these factors affect the cost-effectiveness of decrease and prioritization techniques. We can also use the clustering (Paterlini and Krink, 2014) of test cases to improve the fault detection rate of our test cases suits

Proposed work

The software under test or reliability assessment is frozen. The assessment aims to find out the recent status of system reliability. Thus, during the assessment process, the system will not be adapted even though it can be modified after the assessment. The input domain of the software under test can be divided into n number of sub domains. The test cases in the same subdomain should have some common properties. These properties are related to the software under test, and thus the choice of division can vary a lot of different in software under testing systems. The output of the software under test is independent of the testing record. There are some cases where one test case is deemed failure-free but it actually leads to some fault status that cannot be observed due to limited knowledge of test cases in java. In such situation, this test case is not considered to reveal a failure. However, this imperfect status may cause a failure to be observed when some following test cases are executed even though no failure should be observed after executing these test cases. Then, the prior test cases are mistakenly considered to reveal a failure. This will lead to some error in reliability estimation, but this problem should be considered as a test cases problem rather than a testing strategy one since it can be encountered by every testing strategy without a proper test cases. Software under testing includes selecting a test case from the input domain/subdomain (Junpeng *et al*., 2014), executing the test case, collecting the testing data, and updating the estimates of necessary parameters. Moreover, a total of t test cases are allowed in the assessment Where n_i denotes the number of test cases from subdomain. Each test case will lead the software under test to failure or success.

Pr {observing failures by test cases from Di}= θi

Literature survey

Table.1 Literature survey

S.No.	Author	Title	Approach	Application	Disadvantage
1	Salfner, F. and Malek	Using hidden semi-Markov models for effective online failure prediction.	Failures modeled as non-stationary Bernoulli process	Software reliability prediction	Adapts to changing system Dynamics
2	Liang, Y., Zhang, Y., Siva subramaniam, A., Jette, M., and Sahoo	Failure analysis and prediction models.	Temporal / spatial compression of failures	Extreme-scale parallel systems	Focus on long-running Applications
3	Hoffmann, G. A. and Malek, M.	Call availability prediction in a telecommunication System: A data driven empirical approach.	Approximation of interval call availability by universal basis functions	Performance failures of a telecommunication system	Also applied to response time and memory prediction in Apache web server
4	Fu, S. and Xu, C.-Z	Quantifying temporal and spatial fault event correlation for proactive Failure management.	Estimation of number of failures from CPU utilization and temporal and spatial correlation by use of neural networks	Failures of Wayne computing grid	Focus on number of Failures
5	Abraham and Grosan	Genetic programming approach for fault modeling. .	Genetically generating code to approximate failure probability as a function of external stressors (e.g. temperature)	Power circuit failures of an electronic device	Applicable to systems where the root cause of failures can be assigned to a small set of stressors
6	Meng, H., Di Hou, Y., and Chen, Y.	A rough wavelet network model with genetic algorithm and its application to aging forecasting of application server	Rough set wavelet network to predict next monitoring value	Memory consumption	One step ahead prediction lead-time equal to monitoring interval
7	Salfner and Malek	Prediction-based software availability Enhancement	Model error report sequences using hidden semi-Markov models (HSMM)	Performance failures of a telecommunication system	Includes both type and time of error reports, can handle permutations in event sequences

Adaptive solution with gradient descent method

The solution of operational profile might not be directly utilized in observe, since it needs the true failure rates to allocate the tests. Thus, an accepted idea is to estimate the failure rates and substitute the true failure rates with the estimates to Implement the operational profile solution, e.g., two-stage design. However, the deviate on between the estimates and the true values in the first stage will affect the optimality of testing resource distribution scheme in the second stage. Moreover, the estimation accuracy might be unpredictable since the length of the first stage is set based on experience. In fact, according to the strong law of large number, the estimate will tend to be closer to the true value when more tests are conducted. Therefore, enthusiastically adjusting the testing strategy during the assessment process might be reasonable. If such a testing strategy is properly constructed, e.g., Adaptive Testing for reliability assessment, the estimator variance can be lowered down.

Adaptive Testing Strategy

Adaptive Testing (AT) is a software testing technique that results from the application of feedback and adaptive control ethics in software testing, there are two feedback loops in the AT strategy. The first feedback loop constitutes in fig. 1 the software under test, the database (history of testing data) and the testing strategy, where the history of testing data is used to engender the next test case by a given testing policy or test sufficiency criterion. Feedback loop constitutes the software under test, the database, the parameter estimation scheme and the testing strategy, where the history of testing data is used to improve or change the fundamental testing policy or test sufficiency criterion. The improvement may lead the random testing to switch from one test distribution (e.g., homogeneous distribution) to another test distribution (e.g., non-homogeneous distribution). It may lead partition testing to improve the partitioning of the contribution area of the software under test. It may also lead data flow testing to perform boundary value testing. At the beginning of software testing, the software tester has limited knowledge of the software under test and the ability of the test suite. As testing proceeds, the understanding of the software under test as well as that of the test suite is improved, and thus the testing strategy should be improved. The improvement leads to better test case selection scheme due to better failure rate estimation. The AT strategy for reliability assessment is constructed based on the optimal solution. In order to obtain the optimal solution, software states are defined in terms of the numbers of applied tests and experimental failures, whereas the ordering of applied tests can be ignored. Based on the states, the software under test for reliability assessment can be modeled as a Controlled Markov Chain (CMC) with a special cost constitution. In this way, the problem of determining the optimal testing strategy is of CMC approach. Based on the theories of CMC, it can be accomplished that there exists an optimal solution to this problem. The optimal testing strategy is designed for the testing process, and it is just locally optimal compared with the operational profile solution. Since the parameters, i.e., true failure rates, required by the optimal testing strategy are infrequently known in practice, an AT strategy (Chen *et al.*, 2010) is proposed by substituting the true failure rates with the corresponding estimates that are updated during the assessment process. Furthermore, the optimal testing strategy is determined by recursive evaluation (Tian and Noore, 2013) from the current state to the states where all available testing resources are fatigued, which highly increases the computational complexity. Thus, in practice, the number of following states that are taken into account in the process of recursive evaluation is limited to lower down the computational overhead off AT, and this cooperation might lead AT to deviate from its locally optimal design.

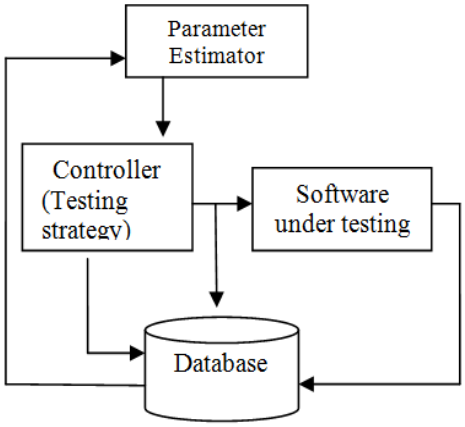

Fig. 1.System Architecture

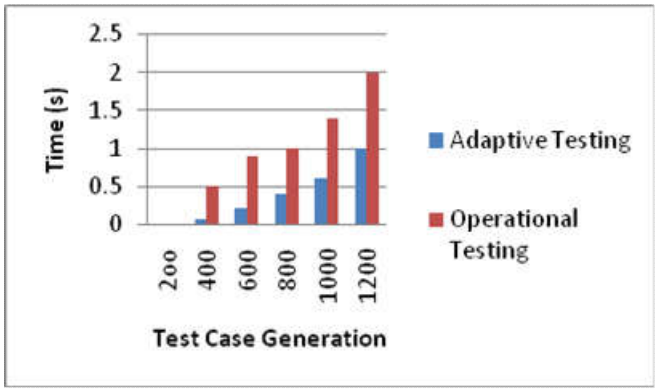

Fig.2. Test Case Generation

Algorithm

Randomly produced a reference point R from the search space Find the nearest individual X ∑ P, which means that compared with other test cases in P,X is chosen randomly

Combine M-1 individuals of the population P, which is nearest to x to form subpopulation

Eliminate the M individual from P

Execute step 2 and 4 repeating until the population P is individual into subpopulation.

EXPERIMENTAL RESULTS

As the number of tests cases is limited, which leads few failures to be observed during the assessment, the mean values of reliability estimates are not so close to the true values. However, when more tests are conducted, the means get closer to the true values. According to the experimental results, the means of AT-GD are closer to the true values than those of Adaptive Testing. Both AT-GD and AT provide more accurate means than operational profile results. Operational profile Fig.

2 provide lower sample variance, which indicates that the AT strategies provide more stable estimates when few testing resources are available. Since the biased estimator is adopted when no failure is observed for one subdomain, MSE is utilized to compare the effectiveness of the testing strategies. The MSE values for the examined testing strategies. It can be seen that AT-GD has lower MSEs than AT, which means that AT-GD provides more accurate and stable estimates than AT. As the four testing strategies are all randomized algorithms, the differences in sample means and variances should be confirmed by statistical test. Figure 3 shows the reliability model for classification integrates various levels of software fault tolerance. This model can be used to illustrate different levels of RCC engagement in an objective system regarding its reliability. Different levels of RCC impact the system reliability through the vary of parameters in these models, which will be discussed in the following segment. The reliability analysis, using the growth of reliability model, to study a theoretical RCC application. The recovery growth model depicted in Figure 3 to predict the system reliability under RCC application. The modeling parameters are classified in three types: failure rates, recovery rates, and coverage factors. The failure rates are estimated from similar projects. The recovery rates are estimated from the underlying recovery mechanism provided by RCC, and the coverage factors are implicit in different test case generation.

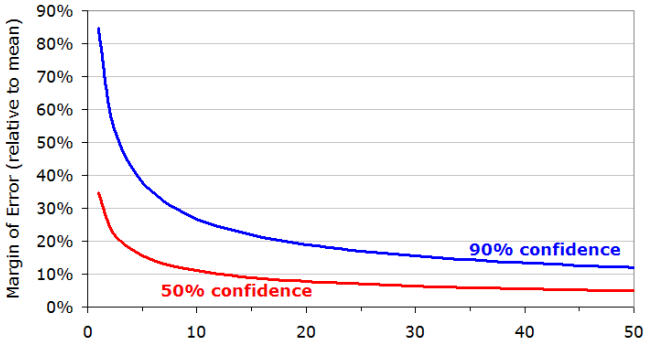

Fig.3. Fault Detection Rate

Conclusion

This new technique AT-GD strategy enhances the Adaptive testing strategy such that both local and global optimality are guaranteed. Specifically, a local optimization scheme is adopted in AT-GD as the test case selection scheme and this leads to a desirable result that AT-GD can converge to globally optimal solution as the assessment profits. Compared with the AT strategy, the computational overhead of AT-GD is also reduced to an acceptable level. The experiments results of real-life software systems are set up to validate the effectiveness and efficiency of AT-GD. The experimental results confirm the global optimality of AT-GD. In those experiments with limited testing resources being available, it can be seen that AT-GD and prior AT strategy outperform of operational profile by providing more accurate and stable reliability estimates. It can also be observed that AT-GD provides more accurate and stable reliability estimates than previous AT strategy. The time cost incurred in decision-making of AT-GD can even be of the same order of enormity with those incurred by operational profile. More specifically, additional real-life mission-critical systems should be involved in the validation; besides, how to unwind the assumptions adopted to examine to improve the flexibility of the testing scaffold.

REFERENCES

Andrea Arcuri, 2012. Member, IEEE, and Lionel Briand, Fellow, "Formal Analysis of the Probability of Interaction Fault Detection Using random testing," IEEE transactions on software engineering, Vol. 38.

Brest, J., S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, 2016. "Adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," IEEE Transactions on Evolutionary Computer, Vol. 10, No. 6, Dec, pp. 646–657.

Chen, T, Kuo, F, Liu, H and Wong, E 2013. "Code coverage of adaptive random testing," IEEE Transactions on Reliability, Vol. 62, No. 1, pp. 226-237.

Chen, T. Y., F.-C. Kuo, R. Merkel, and T. H. Tse. 2010. "Adaptive random testing: The ART of test case diversity," *Journal of Systems and Software*, pp.60-66.

Fang, C., Z. Chen, K. Wu and Z. Zhao. 2014. "Similarity-based test case prioritization using ordered sequences of program entities," *Software Quality Journal*, No. 22, pp.335-361.

Junpeng Lv, Bei-Bei Yin, and Kai-Yuan Cai, 2014. "Estimating confidence interval of software reliability with adaptive testing strategy", *Journal Systems Software*, Vol. 97, Oct. pp. 192–206.

Junpeng Lv, Bei-Bei Yin, and Kai-Yuan Cai, 2014. "On The Asymptotic Behavior of Adaptive Testing", *IEEE Transactions on Software Engineering*, Vol. 40, No. 4, Apr, pp. 396-412.

Lv, J., B.-B. Yin, and K.-Y. Cai, 2014. "On the Asymptotic Behavior of Adaptive Testing Strategy for Software Reliability Assessment," IEEE Transactions on Software Engineering, vol. 40, no. 4, Apr. pp. 396–412,.

Musa, J. D. 2010. "Operational Profiles in Software-Reliability Engineering," *IEEE Software Engineering*, Vol. No. 2, pp. 14-32.

Paterlini S. and T. Krink, 2014. "High performance clustering with differential evolution," IEEE Congress on Evolutionary Computation, Vol. 2, pp. 204–201.

Tian L. and A. Noore, 2013. "Evolutionary neural network modeling for software cumulative failure time prediction," *Reliability Engineering & system safety*, Vol. 87, pp. 45-51.

****