International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN : 2456-3307

Prioritization on Software Reliability Assessment using Adaptive Testing Strategy

Dr. P. Maragathavalli, A.Saranya

Department of Information Technology, Pondicherry Engineering College, Pillaichavadi, Puducherry, Tamil Nadu, India

ABSTRACT

Regression testing assures changed programs against spontaneous improvement. Rearranging the execution order of test cases is a key idea to improve their effectiveness. Many test case prioritization techniques determine test cases using the random selection appear, and yet random ordering of test cases has been considered as ineffective. Adaptive random testing (ART) is a talented aspirant that may deputy random testing (RT). Coverage-based ART techniques are statistically better than Random testing in detecting faults. ART prioritization techniques used for coverage-based prioritization to reduce time consumption.

Keywords : Adaptive Random Testing; Test Case Prioritization

I. INTRODUCTION

Regression testing is an important and however time consuming software development activity. It executes and test suite (T) on a changed program (P) to assure that the program is not harmfully affected by unintended modification. For instance, the retest-all strategy executes all available test cases. Test suites can be large and conduct regression tests is monotonous. To address this difficulty, existing research studies consider different magnitude to make regression testing more feasible to software development. Techniques may execute a subset of T on P, remove some test cases from T enduringly, assign the execution priority of the test cases in T, or use a combination of these Test case selection and diminution may not execute P over certain test cases of T. Although either strategy can make regression testing faster to absolute, the fault detection ability of T is generally compromise. Test case prioritization reorders T for execution to maximize a chosen testing goal. Maximizing the code coverage rate on a given version of the software or business-oriented (e.g., minimizing early human participation in the testing phase). Test case prioritization does not reject any test case, and hence the fault detection ability of T is not compromised. Suppose T = {t1, t2, …, tn} is a regression test suite with n test cases. A test sequence S is an ordered set of test, S=s1, s2, …, sk cases. Furthermore, the notation T⁄S=s1, s2, …, sk, to be to represent the maximal subset of T whose elements are not in S. Without loss of generality, assume the larger such a number, the better S satisfies G. When discuss test case prioritization techniques, discriminate two cases, general prioritization and version specific prioritization. The previous aims at selecting a test case ordering that will be effective (on average) over a sequence of consequent versions of the software. It is particularly applicable when the code bases of consequent versions are unavailable at the time of test case prioritization.

Greedy algorithms [3] are a class of coverage-based test case prioritization techniques. Examples include the total-statement coverage technique and the additional-statement coverage technique. Suppose T is the given failure test suite and a test sequence S has been selected using a technique in this class. Such a technique pick t’ from T/S as the next test case to add to S yields the maximum value in appraisal goal. The additional-statement prioritization technique the coverage information of the enduring test cases when none of them improves the coverage of the test cases already selected.

Although g is no longer a monotonic function, for each round of selection of new test cases, g could still be used as if it were monotonic. In any case, the "collective" adoption of random selection to resolve test cases remains unchanged.

(i) It recommends the first set of coverage-based ART techniques for test case prioritization.
(ii) It reports the first experiential study on ART-based prioritization techniques.

The results show that techniques are better to random ordering in terms of earlier detection of failures. One of the studied ART prioritization Fig1techniques is statistically akin to the best-studied coverage-based prioritization techniques in terms of the fault detection rate, and is much more efficient.

Adaptive Random Testing (ART) [1], which is a challenge to improve the failure-detection effectiveness of random testing. ART is based on various experimental observations showing that many program faults result in failures in proximate areas of the input domain, known as failure patterns. ART analytically guides, or sieve, randomly generated candidates, to take advantage of the likely attendance of such patterns.

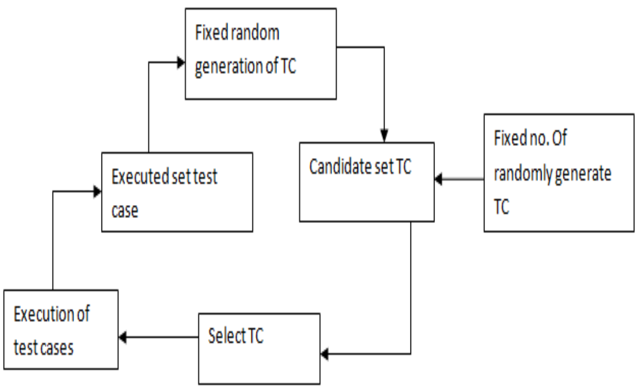

Figure 1. Adaptive Random Testing

II. ADAPTIVE RANDOM TESTING

If contiguous failure regions are indeed common, it would propose that one way to progress the failure detection effectiveness of random testing is to someway taking advantage of this confrontation. One consequence of the existence of contiguous failure regions is that "non-failure regions", that is, constituency of the input domain where the software produces outputs according to specification, will also be proximate.

Therefore, given a set of previously executed test cases that have not exposed any failures, new test cases located away from these old ones are more likely to expose failures in other words; test cases should be more evenly spread during the input domain. Based on this intuition, Adaptive Random Testing (ART) was developed to improve the failure-detection effectiveness of random testing. The first ART method proposed, the Fixed Size Candidate Set ART algorithm (FSCS-ART) [4] candidates are randomly generated. For each candidate ci, the closest previously executed test is located, and the distance di is determined. The aspirant with the largest di is selected, and the other candidates are discarded.

The process is repeated until the desired stopping criterion, be it the tiredness of testing resources or the detection of enough fail is reached. To assess the effectiveness of the FSCS-ART method, the failure detection effectiveness of FSCS-ART to random testing that is, testing by uniform random sampling with replacement on a sample of 12 error-seeded statistical programs.

The original, unmodified programs were used as a testing the correctness of the outputs. The statistic used to compare the methods was the average number of tests required to distinguish the first failure, which is commonly known as the F-measure. In most cases, the F-measure of FSCS-ART was 30–50% lower than that of random testing.

A randomly generated input will be used as the next test case if it lies outside all prohibiting regions; otherwise it will be remaining and the process will be repeated. The efficiency of RRT is very analogous to that of FSCS-ART.

ART by Partitioning uses a rather different perception — in essence, that partitioning the input domain, and allocating test cases evenly to partitions, will achieve even spread. The advantage of failure region contiguity, but using various other intuitions to achieve the "even distribution" of test cases, includes Quasi-Random Testing.

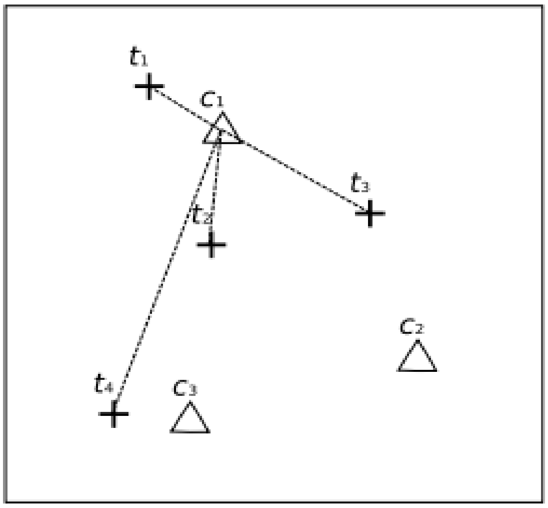

Figure 2. Fixed Size Candidate Set operation

A. Test Case Prioritization Strategies

Test case prioritization test cases so that those with the higher priority, according to some criterion, are executed earlier in the regression testing process. Given a test suite, test case prioritization will find a permutation [6] of the ingenious test suite, aiming to maximize the objective function. There are various strategies based on different intuitions. For example, history-based prioritization techniques use information from prior to executions to determine test priorities; knowledge-based techniques use human knowledge to determine test priorities and model-based techniques use a model of the system to determine test priorities.

B. Adaptive Random Sequence

ART is aimed to improve the fault-detection effectiveness of random testing through the concept of even distribution of test cases in the input domain. It is motivated by the experimental observation that failure-causing inputs are frequently clustered [7] into contiguous failure regions. In other words, if a test case is established to be non-failure-causing, it is very likely that its neighbors will not divulge any failures. Thus, preference should be given to select the contribution far away from the non-failure causing inputs as the subsequently test case. ART can be implemented using various notions of even spread, such as Fixed Size Candidate Set ART (FSCS-ART), constrained random testing, ART by dynamic partitioning, [5] lattice-based ART, and so on. In order to reduce the generation overhead for these algorithms, some general diminution techniques have been developed, such as clustering, mirroring, and forgetting. Since its initiation, ART has been applied into many different types of programs.

III. EXISTING WORK

Test case prioritization schedules test cases with an objective to achieve some performance goal. Various test case prioritization techniques have been proposed using different perception. Among these techniques, execution information acquired in previous test runs to define test case priority and they defined various techniques. Their techniques are shown to be effective at achieving higher values for APFD. Furthermore, several non-greedy algorithms, including hill climbing algorithm and genetic algorithm. Obviously, all of these prioritizations require the test history information of the previous versions. Test case prioritization by ARS uses code coverage. They used Jaccard space and Manhattan distance respectively, to measure the dissimilarity of code exposure. The experimental results showed that they are statistically superior to the random sequence in detecting faults. ART to prioritize test cases based on effecting frequency profiles using frequency Manhattan distance.

A similarity-based test case prioritization technique based on farthest-first regimented progression, which is similar to adaptive random testing. However, these white-box methods presume the availability of certain coverage information or execution frequency profiles. Prioritization technique used string distances to measure the test case diversity, and hence solely depended on the black-box information. However, their algorithm computes the distances for each pair off of test cases to find the first test case with the utmost distance, and then it repeatedly chooses a test case which is most distant from the set of already ordered test cases. Therefore, their prioritized test sequence is deterministic but acquires pricey overhead. Majority of the endure test case prioritization techniques are applied offline. That is, after the prioritization is finished, the test case progression is finalized, and then the regression testing is behavior according to the prioritized test cases until testing resources exhaust. The comparison of existing work is given in table 1

Table 1. Comparison of Existing Work

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	Memoryconsumption	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

IV.CONCLUSION

Adaptive random testing combines random candidate selection with a filtering process to encourage an even spread of test cases throughout the input domain. Based on pragmatic observations that contiguous failure regions are common, tentative studies have shown that ART can detect failures test cases than random testing. In fact, ART methods accomplish close to the hypothetical maximum test case effectiveness by any possible testing method using the same in sequence. Early work on ART strenuous mainly on numeric input domains. As such, we hope that it represents an effective, efficient alternative to random testing in many applications. The Adaptive Random testing is a promising, general method of incremental ordering. The success of ART illustrates the prospective of the approach of failure-based testing, and the brunt and importance that assortment has on the effectiveness of test suites. ART overlay the way for a more rigorous and systematic analysis of the relationships between the in sequence available to the software tester and the effectiveness of families of testing strategies bequest to the foundations of software testing.

V. REFERENCES

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

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

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

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

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

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

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

[8]. Andrea Arcuri, 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, 2012.

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

Author Biography

Dr. P. Maragathavalli
She received her B.E. degree in CSE from Bharathidasan University, M.Tech. degree in CSE from Pondicherry University and Ph.D degree in CSE from Pondicherry University. She is working as Assistant Professor in the Department of Information Technology, Pondicherry Engineering College. She is a Life member of ISTE.

A. Saranya
She is pursuing her M.Tech degree in the Department of Information Technology from Pondicherry Engineering College.