



**Thank you for downloading this document from the RMIT Research Repository.**

The RMIT Research Repository is an open access database showcasing the research outputs of RMIT University researchers.

RMIT Research Repository: <http://researchbank.rmit.edu.au/>

**Citation:**

Liu, H, Kuo, F and Chen, T 2009, 'Dynamic test profiles in adaptive random testing: A case study', in Taghi M. Khoshgoftaar (ed.) Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering, SEKE 2009, Boston, United States, 1-3 July 2009, pp. 418-421.

**See this record in the RMIT Research Repository at:**

<https://researchbank.rmit.edu.au/view/rmit:21081>

**Version:** Published Version

**Copyright Statement:**

© 2009 by Knowledge Systems Institute Graduate School

**Link to Published Version:**

<http://www.ksi.edu/seke/seke09.html>

**PLEASE DO NOT REMOVE THIS PAGE**

# Dynamic Test Profiles in Adaptive Random Testing: A Case Study

Huai Liu\*, Fei-Ching Kuo, and Tsong Yueh Chen

Centre for Software Analysis and Testing, Swinburne University of Technology, Australia

{hliu, dkuo, tychen}@swin.edu.au

## Abstract

*Random testing (RT) is a basic software testing method. When used to detect software failures, RT usually generates random test cases according to a uniform distribution. Adaptive random testing (ART) is an innovative approach to enhancing the failure-detection capability of RT. Most ART algorithms are composed of two independent processes, namely the candidate generation process and the test case identification process. In these ART algorithms, some program inputs are first randomly generated as the test case candidates; then test cases are identified from these candidates in order to ensure an even spread of test cases across the input domain. Most previous studies on ART focused on the enhancement of the test case identification process, while using the uniform distribution in the candidate generation process. A recent study has shown that using a dynamic test profile in the candidate generation process can also improve the failure-detection capability of ART. In this paper, we develop various test profiles and integrate them with the test case identification process of a particular ART algorithm, namely fixed-size-candidate-set ART. It is observed that all these test profiles can significantly improve the failure-detection capability of ART.*

## 1. Introduction

*Random testing (RT)*, a fundamental software testing approach [13], can be used as both a reliability assessment technique [15] and a debug testing method (that is, a method aiming at detecting software failures so that program bugs can be removed [11]). In the context of debug testing, RT usually generates *test cases* (that is, program inputs for testing) based on a uniform distribution from the whole *input domain* (that is, the set of all possible inputs). In other words, all program inputs have the same probability to be generated as test cases.

Although RT has been used in various areas to detect software failures [14, 16], some researchers considered RT as ineffective because RT simply detects failures by chance [13]. Many independent studies [1, 10] have shown that failure-causing inputs (that is, inputs that cause the program under test to exhibit failure behaviors) tend to cluster into contiguous regions (known as *failure regions* [1]) in the input domain. Chen *et al.* [8] made use of such a common characteristic of failure-causing inputs to improve the failure-detection capability of RT. They proposed a novel approach, namely *adaptive random testing (ART)*, where test cases are not only randomly generated, but also evenly spread over the input domain. The basic intuition of ART, that is, the even spread of random test cases, is essentially a form of test cases' diversity across the input domain [5]. In fact, the diversity of test cases is the key concept for most test case selection strategies (such as coverage-based testing methods [17]). ART has been used for testing various programs, from software with numeric inputs [2, 8] to that with complex non-numeric inputs [9].

Various ART algorithms have been proposed to achieve the goal of evenly spreading test cases, such as *fixed-sized-candidate-set ART (FSCS-ART)* [8], *restricted random testing (RRT)* [2] and *lattice-based ART* [12]. Most ART algorithms consist of two independent processes – (a) *candidate generation process*, where some program inputs are randomly generated as *test case candidates*, or briefly *candidates*, and (b) *test case identification process*, where some test case identification criteria are applied to identify test cases amongst these candidates such that the identified test cases are evenly spread over the input domain. Different test case identification criteria lead to different ART algorithms. Previous studies have shown that ART can detect failures more effectively than RT in many cases.

Most studies on ART used the uniform distribution as the test profile in the candidate generation process. Recently, Chen *et al.* [4] proposed a new approach, namely *ART with dynamic non-uniform candidate dis-*

---

\*Corresponding author

*tribution* (ART-DNC). In ART-DNC, the candidate generation process is no longer conducted based on a uniform distribution, but on a dynamic non-uniform test profile. They selected one particular test profile and integrated such a profile with the test case identification processes of FSCS-ART and RRT algorithms. Their simulation studies showed that using the new test profile can significantly improve the effectiveness of the original ART algorithms.

In this paper, we further investigate into various test profiles that may be suitable for ART, and combine them with the test case identification process of FSCS-ART. We attempt to see whether and to what extent these profiles can enhance the failure-detection capability of ART. The rest of the paper is organized as follows. Section 2 introduces the background information on FSCS-ART and ART-DNC. In Section 3, we propose different test profiles, and investigate the effectiveness of ART that uses these test profiles in the candidate generation process. Section 4 concludes the paper.

## 2. Background

Fixed-size-candidate-set ART (FSCS-ART) [8] is a typical ART algorithm. In FSCS-ART, two sets of test cases are maintained, namely the *executed set*  $E$  and the *candidate set*  $C$ .  $E$  is composed of all the previously executed test cases, while  $C$  contains a fixed number of test case candidates that are normally generated in a random manner according to a uniform distribution. A candidate in  $C$  is identified as the next test case if its nearest neighbor distance to  $E$  is the longest. The details of FSCS-ART algorithm can be found in [8]. In our study, the size of the candidate set is set to 10, as recommended in [8].

ART aims to evenly spread random test cases over the whole input domain, but no ART algorithm is guaranteed to achieve such a goal under all possible scenarios [3]. Most previous studies on ART focused on the enhancement of test case identification process, but kept using the uniform distribution in the candidate generation process. Chen *et al.* [4] recently proposed ART with dynamic non-uniform candidate distribution (ART-DNC), which uses a test profile different from the uniform distribution in the candidate generation process. The aim of the new test profile in ART-DNC is to improve the evenness of test case distribution, and thus enhance the failure-detection capability.

FSCS-ART algorithm normally has a bias of identifying test cases from the edge part of the input domain rather than from the centre, and such an *edge bias* results in a certain degree of uneven test case distribution. FSCS-ART-DNC [4] was developed to integrate a new test profile with the test case identification

process of FSCS-ART algorithm. It has been suggested that the test profile used in FSCS-ART-DNC should (i) be dynamic along the testing process, (ii) assign a higher probability to the candidates from the central part of the input domain than those from the edge part (namely, the *centre bias*); and (iii) have a symmetric probability distribution with respect to the centre of the input domain. Readers who are interested may refer to [4] for the details of how to implement FSCS-ART-DNC algorithm.

The failure-detection capability of ART is normally measured by F-measure, which refers to the expected number of test cases required to detect the first software failure. Most previous studies of ART [2, 6, 8] estimated the F-measure of ART (denoted by  $F_{ART}$  in the rest of the paper) via simulations. In order to simulate faulty programs, these simulations first predefine two basic features of a faulty program, namely *failure rate* (denoted by  $\theta$ , which refers to the ratio between the number of failure-causing inputs and the number of all possible inputs) and *failure pattern* (which refers to the failure regions together with their distribution over the input domain). The size and shape of the failure region can then be decided based on  $\theta$  and the failure pattern, and the location of the failure region is randomly chosen inside the input domain. After setting up these parameters, ART is applied until the first failure is detected (that is, a point is picked from the failure region), and the number of test cases that ART has generated will be recorded. Such a process is repeated until we can get a statistically reliable value of  $F_{ART}$ . The details of how to conduct simulations can be found in [6]. The improvement of ART over RT is always evaluated by the *ART F-ratio*  $= F_{ART}/F_{RT}$ , where  $F_{RT}$  denotes the F-measure of RT that is theoretically equal to  $1/\theta$ .

## 3. Effectiveness of FSCS-ART-DNC with various dynamic test profiles

There exist many distributions that have the features mentioned in Section 2 (that is, features (i) to (iii)). Chen *et al.* only selected one dynamic distribution profile to illustrate the new ART-DNC approach. In this study, we propose three other test profiles, namely *triangle*, *cosine* and *semicircle* profiles, for FSCS-ART-DNC. These profiles are named after the basic shapes of the curves of their probability density functions (pdf), which are given in Formulas (1), (2), and (3). Obviously, the probability distributions of these profiles can be adjusted by changing the value of the parameter  $\alpha$ .

**Triangle profile:**

$$f_X(x) = \begin{cases} 4\alpha x + (1 - \alpha) & , 0 \leq x < 0.5 \\ -4\alpha x + (1 + 3\alpha) & , 0.5 \leq x < 1 \\ 0 & , x < 0 \text{ or } x \geq 1 \end{cases} \quad (1)$$

where  $0 \leq \alpha \leq 1$ .

**Cosine profile:**

$$f_X(x) = \begin{cases} \alpha \sin \pi x + \left(1 - \frac{2\alpha}{\pi}\right), & 0 \leq x < 1 \\ 0, & x < 0 \text{ or } x \geq 1 \end{cases} \quad (2)$$

where  $0 \leq \alpha \leq \frac{\pi}{2}$ .

**Semicircle profile:**

$$f_X(x) = \begin{cases} \alpha \sqrt{1 - (2x - 1)^2} + \left(1 - \frac{\alpha\pi}{4}\right), & 0 \leq x < 1 \\ 0, & x < 0 \text{ or } x \geq 1 \end{cases} \quad (3)$$

where  $0 \leq \alpha \leq \frac{4}{\pi}$ .

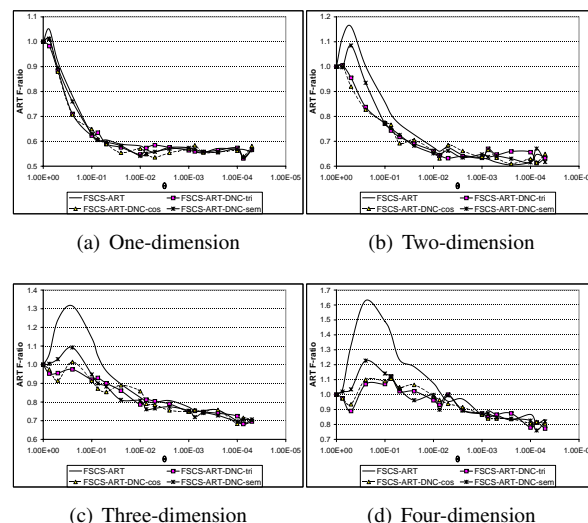
We attempted to integrate the above-mentioned three test profiles with the test case identification process of FSCS-ART, and then get three new ART algorithms, namely FSCS-ART-DNC with triangle, cosine and semicircle profiles. In these algorithms,  $\alpha$  in Formulas (1), (2), and (3) is dynamically adjusted along the testing process. In this paper, we use triangle profile to illustrate how to adjust the value of  $\alpha$ . For ease of illustration, assume that each dimension of input domain has the value range  $[0, 1)$ , and is equally divided into two subranges, namely the *centre subrange* consisting of  $[0.25, 0.75)$ ; and the *edge subrange* consisting of  $[0, 0.25)$  and  $[0.75, 1)$ . For each dimension, successively after each new test case is identified, the following three steps are conducted to adjust  $\alpha$ . First, we measure the ratio ( $r$ ) of the number of executed test cases from the edge subrange over the total number of executed test cases. Second, we calculate the probability ( $p$ ) of an element being generated from the centre subrange. From Formula (1), we can get

$$p = 0.25\alpha + 0.5 \quad (4)$$

Since  $0 \leq \alpha \leq 1$ ,  $p$  is within the value range  $[0.5, 0.75]$ . If  $0.5 \leq r \leq 0.75$ , we set  $p = r$ ; otherwise, we set  $p = 0.75$  (if  $r > 0.75$ ) or  $0.5$  (if  $r < 0.5$ ). Finally,  $\alpha$  can be determined from Formula 4.

We conducted a series of simulations to evaluate the failure-detection capabilities of these new FSCS-ART-DNC algorithms. In these simulations, the dimension of the input domain is one, two, three or four; the shape of the input domain is set as hyper-cube; the failure pattern is one hyper-cube randomly placed inside the input domain; and  $\theta$  is set from 1 to 0.00005. The simulation results are given in Figure 1, which also includes the previous results of the original FSCS-ART algorithm with uniform candidate distribution (denoted by ‘‘FSCS-ART’’ in the figure). The results of

three new FSCS-ART-DNC algorithms are represented by ‘‘FSCS-ART-DNC-tri’’, ‘‘FSCS-ART-DNC-cos’’ and ‘‘FSCS-ART-DNC-sem’’. In the figure, the x- and y-axes denote  $\theta$  and the ART F-ratio, respectively.



**Figure 1. Failure-detection capabilities of FSCS-ART-DNC with various test profiles**

Based on the simulation results, we can observe that all three FSCS-ART-DNC algorithms outperform the original FSCS-ART algorithm when the dimension of the input domain is high or  $\theta$  is high, and the performance improvement increases with the increase in dimension or  $\theta$ . For the cases of low dimension and low  $\theta$ , the failure-detection capability of the original FSCS-ART algorithm is very close to the theoretical bound that can be reached by an optimal testing method without prior information about the failure region’s location [7]. Therefore, it is expected that these FSCS-ART-DNC algorithms cannot significantly improve the performance of ART when dimension or  $\theta$  is low. Briefly speaking, using some proper dynamic test profiles in the candidate generation process does help to improve the failure-detection capability of ART, especially for the cases of high dimension and high  $\theta$ .

It can also be observed that there are some differences in the effectiveness of the three FSCS-ART-DNC algorithms. FSCS-ART-DNC-tri and FSCS-ART-DNC-cos always have similar failure-detection capabilities, but FSCS-ART-DNC-sem does not perform as well as the other two under the conditions of high dimension and high  $\theta$ . For example, when  $\theta = 0.25$  and the dimension is 4, the ART F-ratios of FSCS-ART-DNC-tri, FSCS-ART-DNC-cos, and FSCS-ART-DNC-sem are 1.07, 1.10, and 1.23, respectively. Such a phenomenon can be explained as follows. As shown in [3],

the original FSCS-ART algorithm has an edge bias, which becomes higher with the increase in dimension or  $\theta$ . The test profiles used in FSCS-ART-DNC all have a centre bias. From Formulas (1), (2), and (3), we can calculate that the triangle profile has the highest centre bias, followed by the cosine and semicircle profiles in descending order. When the dimension or  $\theta$  is low, the test case identification process of the original FSCS-ART algorithm does not deliver a very high degree of edge bias. In such a situation, all three test profiles can provide a sufficient degree of centre bias in the candidate generation process to offset the edge bias in the test case identification process. On the other hand, when the dimension and  $\theta$  are high, the low centre bias offered by the semicircle profile may not fully offset the extraordinary edge bias caused by the test case identification process. Therefore, it is intuitively expected that FSCS-ART-DNC-sem does not perform very well under the conditions of high dimension and high  $\theta$ . The similar performances of FSCS-ART-DNC-cos and FSCS-ART-DNC-tri imply that although the centre bias of the cosine profile is lower than that of the triangle profile, the former is sufficient to offset the edge bias in the test case identification process.

#### 4. Conclusions

Adaptive random testing (ART) was proposed to enhance the failure-detection capability of random testing as a debug testing method. Most previous studies have used the uniform distribution as the test profile for ART. A recent study has shown that using a dynamic test profile can further improve the failure-detection capability of ART. In this paper, we conducted some case studies on the application of three dynamic test profiles into ART algorithms. Simulation studies showed that all these three test profiles help to improve the failure-detection capability of ART.

Our experimental results also showed that different test profiles may bring out different failure-detection capabilities of ART. In the future work, we will analyze the statistical features of a variety of dynamic profiles and their impacts on the effectiveness of ART with various test case identification criteria. These investigations will provide new guidelines for how to develop and apply appropriate test profiles for different ART algorithms.

#### Acknowledgment

This research project is partially supported by an Australian Research Council Grant (DP0880295).

#### References

- [1] P. E. Ammann and J. C. Knight. Data diversity: an approach to software fault tolerance. *IEEE Transactions on Computers*, 37(4):418–425, 1988.

- [2] K. P. Chan, T. Y. Chen, and D. Towey. Restricted random testing: adaptive random testing by exclusion. *International Journal of Software Engineering and Knowledge Engineering*, 16(4):553–584, 2006.
- [3] T. Y. Chen, F.-C. Kuo, and H. Liu. On test case distributions of adaptive random testing. In *Proceedings of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE 2007)*, pages 141–144, 2007.
- [4] T. Y. Chen, F.-C. Kuo, and H. Liu. Application of a failure driven test profile in random testing. *IEEE Transactions on Reliability*, 58(1):179–192, 2009.
- [5] T. Y. Chen, F.-C. Kuo, R. Merkel, and T. H. Tse. Adaptive random testing: the ART of test case diversity. Accepted to appear in *Journal of Systems and Software*.
- [6] T. Y. Chen, F.-C. Kuo, and Z. Q. Zhou. On favorable conditions for adaptive random testing. *International Journal of Software Engineering and Knowledge Engineering*, 17(6):805–825, 2007.
- [7] T. Y. Chen and R. Merkel. An upper bound on software testing effectiveness. *ACM Transactions on Software Engineering and Methodology*, 17(3):16:1–16:27, 2008.
- [8] T. Y. Chen, T. H. Tse, and Y. T. Yu. Proportional sampling strategy: a compendium and some insights. *Journal of Systems and Software*, 58(1):65–81, 2001.
- [9] I. Ciupa, A. Leitner, M. Oriol, and B. Meyer. AR-TOO: adaptive random testing for object-oriented software. In *Proceedings of the 30th International Conference on Software Engineering (ICSE'08)*, pages 71–80. ACM Press, 2008.
- [10] G. B. Finelli. NASA software failure characterization experiments. *Reliability Engineering and System Safety*, 32(1–2):155–169, 1991.
- [11] P. G. Frankl, R. G. Hamlet, B. Littlewood, and L. Strigini. Evaluating testing methods by delivered reliability. *IEEE Transactions on Software Engineering*, 24(8):586–601, 1998.
- [12] J. Mayer. Lattice-based adaptive random testing. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering (ASE 2005)*, pages 333–336, New York, USA, 2005. ACM.
- [13] G. J. Myers. *The Art of Software Testing*. John Wiley and Sons, second edition, 2004. Revised and updated by T. Badgett and T. M. Thomas with C. Sandler.
- [14] D. Slutz. Massive stochastic testing of SQL. In *Proceedings of the 24th International Conference on Very Large Databases (VLDB 1998)*, pages 618–622, 1998.
- [15] R. A. Thayer, M. Lipow, and E. C. Nelson. *Software Reliability*. North-Holland, 1978.
- [16] T. Yoshikawa, K. Shimura, and T. Ozawa. Random program generator for Java JIT compiler test system. In *Proceedings of the 3rd International Conference on Quality Software (QSIC 2003)*, pages 20–24, 2003.
- [17] H. Zhu, P. A. V. Hall, and J. H. R. May. Software unit test coverage and adequacy. *ACM computing surveys*, 29(4):366–427, 1997.