



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:

Chen, T, Kuo, F and Liu, H 2007, 'Distribution metric driven adaptive random testing', in Aditya Mathur, W. Ericwong, And M. F. Lau (ed.) Proceedings of the 7th International Conference on Quality Software (QSIC2007), Portland, United States, 11-12 October, 2007, pp. 274-279

See this record in the RMIT Research Repository at:

<http://researchbank.rmit.edu.au/view/rmit:21272>

Version: Accepted Manuscript

Copyright Statement: © 2007 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved

Link to Published Version:

<http://dx.doi.org/10.1109/QSIC.2007.4385507>

PLEASE DO NOT REMOVE THIS PAGE

Distribution Metric Driven Adaptive Random Testing

Tsong Yueh Chen Fei-Ching Kuo Huai Liu*
Faculty of ICT, Swinburne University of Technology, Australia
{tchen, dkuo, hliu}@ict.swin.edu.au

Abstract

Adaptive Random Testing (ART) was developed to enhance the failure detection capability of Random Testing. The basic principle of ART is to enforce random test cases evenly spread inside the input domain. Various distribution metrics have been used to measure different aspects of the evenness of test case distribution. As expected, it has been observed that the failure detection capability of an ART algorithm is related to how evenly test cases are distributed. Motivated by such an observation, we propose a new family of ART algorithms, namely distribution metric driven ART, in which, distribution metrics are key drivers for evenly spreading test cases inside ART. Our study uncovers several interesting results and shows that the new algorithms can spread test cases more evenly, and also have better failure detection capabilities.

1. Introduction

Random Testing (RT) is a basic software testing technique. It simply generates test cases in a random manner from the whole *input domain* (the set of all possible inputs) [9, 14]. RT has been used in different areas to detect software failures. For example, RT was used to test standard UNIX utilities and it was reported that lots of utility programs had been crashed or hanged by random test data [12, 13]. RT was also applied to test Java JIT compilers [17], SQL database systems [15], image processing applications [11], communications protocols implementations [16], and so on.

However, RT has been criticized as inefficient because it uses little information of system under test when generating test cases. One common characteristic of faulty programs is that the *failure-causing inputs* (program inputs that can reveal failures) are usually clustered together, as reported in [1, 2, 8]. Chen *et al.* [6] investigated how to improve the failure detection capability of RT under such a situation. They proposed a new approach, namely *Adaptive Random*

Testing (ART). Like RT, ART also randomly generates test cases from the input domain. But ART uses additional criteria to guide the test case selection such that test case are evenly spread over the whole input domain. Different test case selection criteria give rise to different ART algorithms, such as *Fixed-Sized-Candidate-Set ART* (FSCS-ART) [6], *Restricted Random Testing* (RRT) [3], and *Lattice-based ART* [10]. Previous simulations and empirical studies conducted on these algorithms have shown that in general, when failure-causing inputs are clustered into contiguous regions (namely *failure regions* [1]), ART could use fewer test cases to detect the first failure than pure RT.

It has been generally believed that how evenly an ART algorithm spreads test cases has an impact on its failure detection capability, and an even distribution of test cases brings a high fault detection capability for ART. Chen *et al.* [4] have used several metrics to measure and compare the test case distributions of various ART algorithms. Among these metrics, *discrepancy* and *dispersion* are two metrics commonly used to measure the equidistribution of sample points. It has been further observed that there is a correlation between the ART performance and the values of these distribution metrics. For example, FSCS-ART generally has a small value of dispersion, but its discrepancy is large when the dimension of input domain is high; while the failure detection capability of FSCS-ART is fairly good for low dimensional cases, but the capability becomes worse with the increase of the dimension of input domain.

Since distribution metrics can reflect not only how evenly test cases are spread, but also the failure detection capability of an ART algorithm to certain degrees, we propose to drive the test case selection process of ART by these metrics to enhance the ART performance. In this paper, we adopt discrepancy and dispersion as new criteria for selecting test cases, instead of using them for measuring the test case distribution. It is expected that the research conducted in this paper can help us answer the following questions.

- How ART performs if discrepancy or dispersion is used as the standalone test case selection criterion?

*Corresponding author

- How ART performs if discrepancy or dispersion works together with other selection criteria?

We conduct our work on a particular ART algorithm, namely FSCS-ART. The paper is organized as follows. In Section 2, we give some background information of FSCS-ART and introduce the basic concepts of discrepancy and dispersion. In Section 3, we propose two new test case selection criteria based on discrepancy and dispersion, and then investigate how to adopt the new criteria in FSCS-ART. Some new ART algorithms are proposed. The simulation studies on these new algorithms and the related results will also be reported in this section. Finally, in Section 4, we conclude the paper.

2. Background

2.1. Notations

For ease of discussion, we introduce the following notations, which will be used in the sequel.

- E denotes the set of already executed test cases.
- D denotes the input domain.
- dD denotes d -dimension, where $d = 1, 2, 3, 4, \dots$, and the dimension of input domain means the number of inputs parameters of the program under test.
- $|E|$ and $|D|$ denote the size of E and D , respectively.
- $dist(p, q)$ denotes the distance between two points p and q .
- $\rho(p, E)$ denotes p 's nearest neighbour in E .

2.2. FSCS-ART

Generally speaking, besides randomly generating program inputs, ART uses additional criteria to select inputs as test cases in order to ensure the even spreading of test cases. The test case selection process in *Fixed-Sized-Candidate-Set ART* (FSCS-ART) [6] is conducted as follows. There exist two sets of test cases, the *executed set* denoted by E and the *candidate set* denoted by C . E contains all test cases which were already executed but have not revealed any failure; while C contains k randomly generated inputs, where k is fixed throughout the testing process. The next test case will be the candidate that has the longest distance to its nearest neighbour in E . Figure 1 gives the detailed algorithm of FSCS-ART. In this paper, the default value of k is set as 10, as recommended in [6].

In this paper, we will follow previous studies [3, 6, 10] to use F-measure (the expected number of test cases required to detect the first failure) for measuring the failure detection capability of ART (the preference of F-measure to other measures on ART/RT was justified in [7]). The F-measure of ART (denoted by F_{ART}) depends on many factors, so it is very difficult to theoretically derive the value of F_{ART} . In [5], F_{ART} for FSCS-ART was studied via a series of simulations. In each

1. Input an integer k , where $k > 1$.
2. Set $n = 0$ and $E = \{\}$.
3. Randomly generate a test case t from D , according to uniform distribution.
4. Test the program with t as the program input.
5. **while** (no failure has been revealed)
6. Store t into E , and increment n by 1.
7. Randomly generate k program inputs (candidates) from D , according to uniform distribution, and construct C with these candidates.
8. Find $c_b \in C$ such that $\forall j = 1, \dots, k, dist(c_b, \rho(c_b, E)) \geq dist(c_j, \rho(c_j, E))$.
9. Set $t = c_b$.
10. Test the program with t as the program input.
11. **end_while**
12. Report the detected failure.
13. Exit.

Figure 1. The algorithm of FSCS-ART

simulation, the failure rate θ (the ratio of the number of failure-causing inputs to the number of all possible inputs) and the failure pattern (the shapes of failure regions together with their distribution over the input domain D) were predefined. Test cases were generated one by one until a point inside the failure region was picked by ART (that is, a failure was detected). Such a process was repeated for a sufficient number of times until the mean value of F_{ART} was reliable within 95% confidence level and $\pm 5\%$ accuracy range (details of simulations can be found in [5]). We will use similar experiment setting to investigate the F_{ART} of the algorithms developed in this paper.

ART was originally proposed to enhance the failure detection capability of RT, whose F-measure (denoted by F_{RT}) is theoretically equal to $1/\theta$ when test cases are selected with replacement, according to uniform distribution. In this paper, we will use ART F-ratio ($= F_{ART}/F_{RT}$) to measure the enhancement of ART over RT.

2.3. Discrepancy and dispersion

Chen *et al.* [4] have used discrepancy and dispersion to measure the test case distribution of FSCS-ART (as well as some other ART algorithms). For ease of discussion, the detailed definitions of these metrics are given as follows.

- **Discrepancy.**

$$M_{Discrepancy} = \max_{i=0..m} \left| \frac{|E_i|}{|E|} - \frac{|D_i|}{|D|} \right|, \quad (1)$$

where D_1, D_2, \dots, D_m denote m randomly defined subsets of D , with their corresponding sets of test cases being denoted by E_1, E_2, \dots, E_m , which are subsets of E . The value of m was set as 1000 in [4]. Discrepancy intuitively indicates whether different regions in D have an equal density of points. A low discrepancy implies E is reasonably equidistributed.

- **Dispersion.**

$$M_{Dispersion} = \max_{i=1\dots|E|} dist(e_i, \mathcal{N}(e_i, E \setminus \{e_i\})), \quad (2)$$

where $e_i \in E$.

Dispersion intuitively indicates whether there is a large empty spherical region (containing no point) in D . A low dispersion also implies a reasonable equidistribution of E .

It has been reported in [4] that the performance of an ART algorithm has a correlation with the test case distribution. This has motivated us to consider whether the performance of ART can be enhanced if we enforce a lower discrepancy or a lower dispersion during its test case selection process. Obviously, such an enforcement can be achieved by using discrepancy or dispersion as the test case selection criterion in an ART algorithm, with details reported in the next section.

3. Enhancing FSCS-ART by Using Discrepancy and Dispersion as Test Case Selection Criteria

3.1. Adopting discrepancy and dispersion as test case selection criteria

The following outlines how discrepancy and dispersion could be used as test case selection criteria in FSCS-ART.

- **Test case selection criterion based on discrepancy** (denoted by $S_{Discrepancy}$). Given the candidate set C in FSCS-ART, for any $c_j \in C$, we define

$$d_{discrepancy}^j = \max_{i=0\dots m} \left| \frac{|E'_i|}{|E'|} - \frac{|D_i|}{|D|} \right|, \quad (3)$$

where $E' = E \cup \{c_j\}$, and D_1, D_2, \dots, D_m denote m randomly defined subsets of D , with their corresponding sets of test cases being denoted by E'_1, E'_2, \dots, E'_m , which are subsets of E' . We choose a candidate c_b as the next test case, if $\forall j = 1, \dots, k, d_{discrepancy}^b \leq d_{discrepancy}^j$. To be consistent with the previous study [4], the value of m in Equation 3 is also set as 1000 in this paper.

- **Test case selection criterion based on dispersion** (denoted by $S_{Dispersion}$). Given the candidate set C in FSCS-ART, for any $c_j \in C$, we define

$$d_{dispersion}^j = \max_{i=1\dots|E'|} dist(e'_i, \mathcal{N}(e'_i, E' \setminus \{e'_i\})), \quad (4)$$

where $E' = E \cup \{c_j\}$ and $e'_i \in E'$. We choose a candidate c_b as the next test case, if $\forall j = 1, \dots, k, d_{dispersion}^b \leq d_{dispersion}^j$.

For consistency, we use $S_{distance}$ to denote the test case selection criterion in the original FSCS-ART algorithm (Line 8 in Figure 1) in the sequel.

3.2. Study on FSCS-ART using discrepancy or dispersion as the standalone criterion to select test cases

We replaced $S_{distance}$ in FSCS-ART algorithm (Figure 1) by $S_{discrepancy}$ and $S_{dispersion}$, and got two new algorithms, namely *FSCS-ART with selecting by $S_{discrepancy}$* (abbreviated as FSCS-ART-disc) and *FSCS-ART with selecting by $S_{dispersion}$* (abbreviated as FSCS-ART-disp), respectively. For clarify, we use FSCS-ART-dist to denote the original FSCS-ART algorithm in the sequel.

We conducted some simulations to study the failure detection capabilities of FSCS-ART-disc and FSCS-ART-disp. We found that FSCS-ART-disc only outperforms RT marginally, and FSCS-ART-disp always has a higher F-measure than RT. Simply speaking, neither discrepancy nor dispersion will result in a high failure detection capability for ART when each of them is applied as the standalone test case selection criterion.

In order to find the reasons why these two algorithms cannot perform better than RT, we further investigated their test case distributions (the experimental setting can be found in [4]). It was observed that FSCS-ART-disc always has a smaller $M_{Discrepancy}$ than FSCS-ART-dist, as intuitively expected; however, its $M_{Dispersion}$ is larger than that of FSCS-ART-dist, and similar to that of RT. As explained in [4], $M_{Dispersion}$ is better than $M_{Discrepancy}$ to indicate the correlation between the test case distribution and the performance of an ART algorithm. FSCS-ART-disc does not have a smaller $M_{dispersion}$ than RT, although it has a small $M_{discrepancy}$. Therefore, it is understandable that FSCS-ART-disc does not significantly outperform RT. This also tells us that a small $M_{Discrepancy}$ alone is not enough to ensure a good failure detection capability of ART.

As far as FSCS-ART-disp is concerned, we found that FSCS-ART-disp normally has a fairly large $M_{Discrepancy}$ although its $M_{Dispersion}$ is smaller than that of FSCS-ART-dist. The large value of $M_{Discrepancy}$ for FSCS-ART-disp may be due to the definition of dispersion used in this study. The intuition of dispersion is to measure the largest empty spherical region inside D . Given that the sample points are uniformly distributed, the largest nearest neighbour distance is a good metric to reflect the size of this empty spherical region. However, when FSCS-ART-disp solely uses such a definition to select test cases without considering the uniform distribution, it is quite likely that the selected test cases would be clustered into some regions inside D (a large $M_{Discrepancy}$). As a result, FSCS-ART-disp has a poor failure detection capability.

Briefly speaking, although discrepancy and dispersion measure certain aspects of the evenness of test case

distribution, neither of them can ensure the even spreading of test cases if each of them is solely used as the test case selection criterion in ART. This is due to the fact that a low discrepancy and a low dispersion are just necessary characteristics of the even spreading of test cases, not vice versa. Compared with these two criteria, $S_{distance}$ is a better test case selection criterion, because it gives FSCS-ART-dist a small $M_{Dispersion}$, and also a high failure detection capability. However, as pointed out in [4], $S_{distance}$ is not perfect, because it may result in a relatively large $M_{Discrepancy}$ and the performance of FSCS-ART-dist may be deteriorated in some special cases. In the next section, we will investigate how to boost up the performance of ART by using $S_{discrepancy}$ and $S_{dispersion}$ as additional test case selection criteria to supplement $S_{distance}$.

3.3. Integration of discrepancy and dispersion with the test case selection criterion in FSCS-ART

We here propose new algorithms (Figure 2), which use $S_{discrepancy}$ or $S_{dispersion}$ to select test cases together with $S_{distance}$.

1. Input two integer k_1 and k_2 , where $k_1 > k_2 > 1$.
2. Set $n = 0$ and $E = \{\}$.
3. Randomly generate a test case t from D , according to uniform distribution.
4. Test the program with t as the program input.
5. **while** (no failure has been revealed)
6. Store t into E , and increment n by 1.
7. Randomly generate k_1 program inputs (candidates) from D , according to uniform distribution, and construct C with these candidates.
8. Find k_2 best candidates c'_1, \dots, c'_{k_2} from C , according to $S_{distance}$, and construct a new candidate set $C' = \{c'_1, \dots, c'_{k_2}\}$.
9. Find the best candidate c'_b from C' , according to S_{metric} , where $S_{metric} = S_{discrepancy}$ or $S_{dispersion}$.
10. Set $t = c'_b$.
11. Test the program with t as the program input.
12. **end_while**
13. Report the detected failure.
14. Exit.

Figure 2. The algorithm of distribution metric driven FSCS-ART

In the new algorithms, $S_{distance}$ is used as the primary test case selection criterion (Line 8 in Figure 1), while $S_{discrepancy}$ and $S_{dispersion}$ are adopted as secondary criteria (Line 9). Since some of selection criteria in the new algorithms are originally from some distribution metrics, we term the new algorithms as *distribution metric driven ART*. We totally have two new ART algorithms, which are *distribution metric driven FSCS-ART with selecting by $S_{distance}$ and then $S_{discrepancy}$* (abbreviated as FSCS-ART-dist-disc), and *distribution metric driven*

FSCS-ART with selecting by $S_{distance}$ and then $S_{dispersion}$ (abbreviated as FSCS-ART-dist-disp), respectively. In the next two sections, the test case distributions and the failure detection capabilities of the new algorithms are examined, respectively.

3.4. Test case distributions of distribution metric driven FSCS-ART

We conducted a series of simulations to measure the test case distributions of new ART algorithms (with $k_1 = 10$ and $k_2 = 3$), and got the values of $M_{Discrepancy}$ and $M_{Dispersion}$. The simulation results are summarised in Tables 1 and 2, which also include the data of FSCS-ART-dist and RT for ease of comparison. Due to page limit, we cannot present all data in graphs, but we wish to point out that the variations of $M_{Discrepancy}$ and $M_{Dispersion}$ with respect to $|E|$ are very similar for all scenarios.

Table 1. Comparing $M_{Discrepancy}$ of FSCS-ART-dist-disc, FSCS-ART-dist-disp, FSCS-ART-dist and RT

dimension	Value Range of $M_{Discrepancy}$	RT	FSCS-ART-dist	FSCS-ART-dist-disc	FSCS-ART-dist-disp
1D	max	1.06E-01	4.37E-02	2.97E-02	4.29E-02
	min	1.05E-02	4.02E-03	1.13E-03	4.22E-03
	max-min	9.51E-02	3.96E-02	2.86E-02	3.86E-02
2D	max	1.09E-01	7.19E-02	4.71E-02	7.89E-02
	min	1.06E-02	6.19E-03	1.85E-03	6.92E-03
	max-min	9.87E-02	6.57E-02	4.53E-02	7.20E-02
3D	max	9.25E-02	8.76E-02	5.01E-02	1.05E-01
	min	9.18E-03	1.45E-02	2.63E-03	2.09E-02
	max-min	8.33E-02	7.31E-02	4.75E-02	8.42E-02
4D	max	7.85E-02	7.93E-02	4.57E-02	9.19E-02
	min	7.69E-03	1.88E-02	3.90E-03	2.96E-02
	max-min	7.09E-02	6.05E-02	4.18E-02	6.23E-02

Table 2. Comparing $M_{Dispersion}$ of FSCS-ART-dist-disc, FSCS-ART-dist-disp, FSCS-ART-dist and RT

dimension	Value Range of $M_{Dispersion}$	RT	FSCS-ART-dist	FSCS-ART-dist-disc	FSCS-ART-dist-disp
1D	max	2.72E-02	1.60E-02	1.66E-02	1.51E-02
	min	4.89E-04	2.07E-04	2.35E-04	2.01E-04
	max-min	2.67E-02	1.58E-02	1.64E-02	1.49E-02
2D	max	1.49E-01	1.28E-01	1.33E-01	1.19E-01
	min	1.89E-02	1.41E-02	1.45E-02	1.36E-02
	max-min	1.30E-01	1.14E-01	1.18E-01	1.05E-01
3D	max	2.85E-01	2.73E-01	2.79E-01	2.54E-01
	min	7.14E-02	6.01E-02	6.22E-02	5.86E-02
	max-min	2.14E-01	2.13E-01	2.17E-01	1.96E-01
4D	max	4.14E-01	4.14E-01	4.12E-01	3.85E-01
	min	1.46E-01	1.30E-01	1.32E-01	1.26E-01
	max-min	2.68E-01	2.84E-01	2.80E-01	2.59E-01

Based on these data, we have the following observations.

- As expected, FSCS-ART-dist-disc always has the smallest $M_{Discrepancy}$. Its $M_{Dispersion}$ is slightly larger than that of FSCS-ART-dist, but smaller than that of RT.
- FSCS-ART-dist-disp normally has the smallest $M_{Dispersion}$. Its $M_{Discrepancy}$ increases with the increase of the dimension, like that of FSCS-ART-dist, but generally larger than that of FSCS-ART-dist.

Briefly speaking, both FSCS-ART-dist-disc and FSCS-ART-dist-disp distribute their test cases not only very well with respect to their own distribution metrics

(namely discrepancy and dispersion, respectively), but also well with respect to the other metrics.

3.5. Failure detection capabilities of distribution metric driven FSCS-ART

We also conducted some simulations to study the performance of FSCS-ART-dist-disc and FSCS-ART-dist-disp. In these simulations, the input domain D was set to be square and the dimension of D was set as either 1D, 2D, 3D or 4D. A single square failure region was randomly placed inside D . The size of the failure region was decided by the failure rate θ , where θ was set from 0.75 to 0.00005. The simulations results are summarized in Figure 3, which also includes F-ratios of FSCS-ART-dist for ease of comparison.

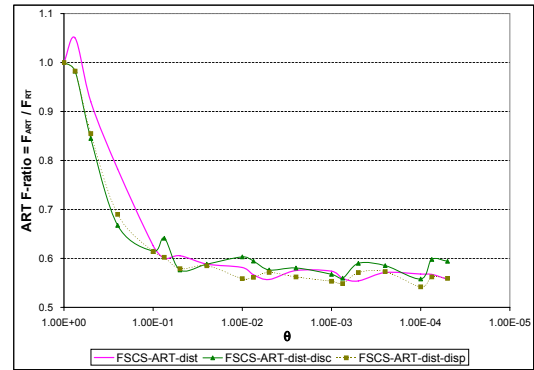
Based on the data in Figure 3, we have the following findings.

- All three ART algorithms have similar failure detection capabilities when failure rate is low.
- Both FSCS-ART-dist-disc and FSCS-ART-dist-disp have a smaller F_{ART} than FSCS-ART-dist when the failure rate is high.
- The performance improvement of the two new algorithms over FSCS-ART-dist increases with the increase of the dimension of D .

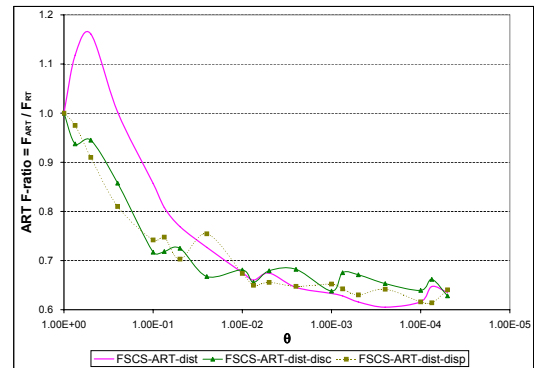
In summary, FSCS-ART-dist-disc and FSCS-ART-dist-disp generally have a better failure detection capability than the original FSCS-ART algorithm (FSCS-ART-dist), especially when the failure rate is high or the dimension of D is high. Compared with FSCS-ART-dist, FSCS-ART-dist-disc has a significantly smaller $M_{Discrepancy}$, but a comparable $M_{Dispersion}$. On the other hand, compared with FSCS-ART-dist, FSCS-ART-dist-disp has a smaller $M_{Dispersion}$, but a comparable $M_{Discrepancy}$. These results reinforce the intuition of ART that the more evenly test cases are spread, the better the failure detection capability is.

4. Discussion and Conclusion

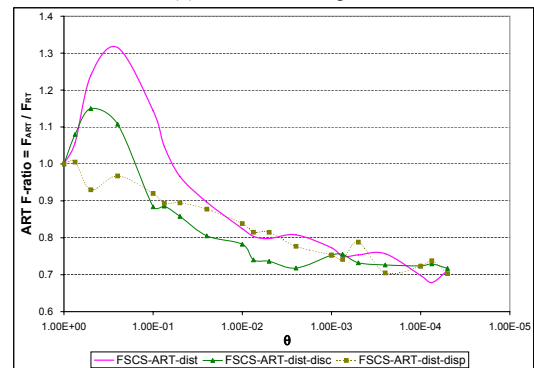
ART was proposed as an enhancement of RT. ART improves the failure detection capability by evenly spreading random test cases inside the input domain. In ART, some test case selection criteria are used to ensure the even spreading of random test cases. Though even spreading is intuitively simple, there does not exist a standard definition of even spreading, needless to say the existence of a standard measurement for the evenness of test case distribution. Research [4] has been attempted to use various distribution metrics to reflect, if not measure, how evenly an ART algorithm spreads test cases. Previous studies have conclusively shown that some ART algorithms, which are not regarded by certain distribution metrics as evenly spreading their test cases, usually perform poorly. This correlation between



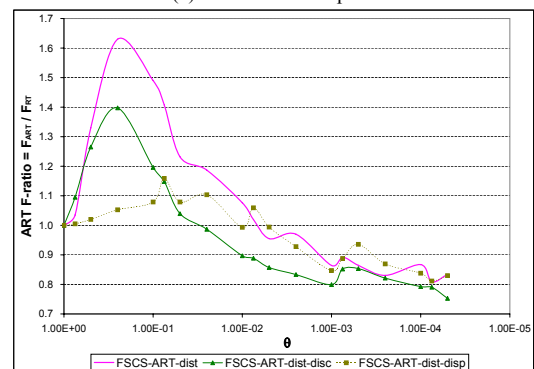
(a) F-ratios in 1D space



(b) F-ratios in 2D space



(c) F-ratios in 3D space



(d) F-ratios in 4D space

Figure 3. Comparing failure detection capabilities of FSCS-ART-dist-disc, FSCS-ART-dist-disp, and FSCS-ART-dist

test case distribution and failure detection capability has motivated us to develop some new ART algorithms, which apply these distribution metrics as test case selection criteria in ART.

We first developed some algorithms using each of these metrics as the standalone criterion to select test cases in ART. Our simulation results showed that these ART algorithms not only have poor performances, but also unevenly distribute their test cases. Such results should not be surprising, because even spreading implies both low discrepancy and low dispersion, but neither low discrepancy nor low dispersion is sufficient on its own to imply even spreading.

We further investigated the integration of these metrics and the notion of “far apart” in FSCS-ART (that is, keeping test cases as far apart from one another as possible), and proposed a new family of ART algorithms, namely *distribution metric driven ART*. The simulation results showed that these new algorithms do improve the evenness of test case distribution and enhance the failure detection capability.

There are various definitions of discrepancy and dispersion in the literature, and we have only adopted the most commonly used definitions in this study. In the future work, we can investigate the impacts of other definitions of discrepancy and dispersion. Moreover, some parameters, such as m in Equations 3 and k_1 , k_2 in the distribution metric driven ART algorithm (Figure 2), were arbitrarily set in this study. It is interesting to further investigate the impact of other settings of these parameters on the performance of the new algorithms. Chen *et al.* have pointed out in [4] that the ART algorithms under their study may not well satisfy the definitions of some distribution metrics. RRT, for example, generally has a small $M_{Dispersion}$, but a relatively large $M_{Discrepancy}$, just like FSCS-ART. Therefore, the innovative approach of this study, that is, adopting test case distribution metrics as test case selection criteria in ART, can be equally applied to enhance other existing ART algorithms.

Acknowledgment

This research project is supported by an Australian Research Council Discovery Grant (DP0557246).

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] P. G. Bishop. The variation of software survival times for different operational input profiles. In *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, pages 98–107, 1993.
- [3] 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.

- [4] 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*, pages 141–144, 2007.
- [5] T. Y. Chen, F.-C. Kuo, and Z. Q. Zhou. On favourable conditions for adaptive random testing. *Accepted to appear in International Journal of Software Engineering and Knowledge Engineering*.
- [6] T. Y. Chen, H. Leung, and I. K. Mak. Adaptive random testing. In *Proceedings of the 9th Asian Computing Science Conference*, pages 320–329, 2004.
- [7] T. Y. Chen and R. Merkel. An upper bound on software testing effectiveness. *Accepted to appear in ACM Transactions on Software Engineering and Methodology*.
- [8] G. B. Finelli. NASA software failure characterization experiments. *Reliability Engineering and System Safety*, 32(1–2):155–169, 1991.
- [9] R. Hamlet. Random testing. In J. Marciniak, editor, *Encyclopedia of Software Engineering*. John Wiley & Sons, second edition, 2002.
- [10] J. Mayer. Lattice-based adaptive random testing. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, pages 333–336, New York, USA, 2005. ACM.
- [11] J. Mayer and R. Guderlie. On random testing of image processing applications. In *Proceedings of the 6th International Conference on Quality Software*, pages 85–92. IEEE Computer Society, 2006.
- [12] B. P. Miller, L. Fredriksen, and B. So. An empirical study of the reliability of UNIX utilities. *Communications of the ACM*, 33(12):32–44, 1990.
- [13] B. P. Miller, D. Koski, C. P. Lee, V. Maganty, R. Murthy, A. Natarajan, and J. Steidl. Fuzz revisited: A re-examination of the reliability of UNIX utilities and services. Technical Report CS-TR-1995-1268, University of Wisconsin, 1995.
- [14] G. J. Myers. *The Art of Software Testing*. Wiley, New York, second edition, 1979.
- [15] D. Slutz. Massive stochastic testing of SQL. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 618–622, 1998.
- [16] C. H. West and A. Tosi. Experiences with a random test driver. *Computer Networks and ISDN Systems*, 27:1163–1174, 1995.
- [17] 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*, pages 20–24, 2003.