2007

Enhanced random testing for programs with high dimensional input domains

Fei-Ching Kuo
University of Wollongong, dkuo@uow.edu.au

K. Y. Sim
Swinburne University of Technology

Chang-ai Sun
University of Groningen

S.-F. Tang
Swinburne University of Technology

Zhiquan Zhou
University of Groningen

Recommended Citation

Kuo, Fei-Ching; Sim, K. Y.; Sun, Chang-ai; Tang, S.-F.; and Zhou, Zhiquan: Enhanced random testing for programs with high dimensional input domains 2007.
https://ro.uow.edu.au/infopapers/3086

Enhanced random testing for programs with high dimensional input domains

Abstract

Random Testing (RT) is a fundamental technique of software testing. Adaptive Random Testing (ART) has recently been developed as an enhancement of RT that has better fault detection effectiveness. Several methods (algorithms) have been developed to implement ART. In most ART algorithms, however, the above enhancement diminishes when the dimensionality of the input domain increases. In this paper, we investigate the nature of failure regions in high dimensional input domains and propose enhanced random testing algorithms that improve the fault detection effectiveness of RT in high dimensional input domains.

Disciplines

Physical Sciences and Mathematics

Publication Details

Kuo, F., Sim, K., Sun, C., Tang, S. & Zhou, Z. 2007, 'Enhanced random testing for programs with high dimensional input domains', International Conference on Software Engineering and Knowledge Engineering, Knowledge Systems Institute, USA, pp. 135-140.

Enhanced Random Testing for Programs with High Dimensional Input Domains

F.-C. Kuo[1], K.-Y. Sim[2*], Chang-ai Sun[3], S.-F. Tang[4] and Zhi Quan Zhou[5]
[1,4]Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia
[2]School of Engineering, Swinburne University of Technology, Sarawak Campus, Malaysia
[3]Department of Mathematics and Computing Science, University of Groningen, The Netherlands
[5]School of Computer Science and Software Engineering, University of Wollongong, Australia
[1,4]{dkuo,satang}@ict.swin.edu.au, [2]ksim@swinburne.edu.my, [3]csun@cs.rug.nl,
[5]zhiquan@uow.edu.au

Abstract

Random Testing (RT) is a fundamental technique of software testing. Adaptive Random Testing (ART) has recently been developed as an enhancement of RT that has better fault detection effectiveness. Several methods (algorithms) have been developed to implement ART. In most ART algorithms, however, the above enhancement diminishes when the dimensionality of the input domain increases. In this paper, we investigate the nature of failure regions in high dimensional input domains and propose enhanced random testing algorithms that improve the fault detection effectiveness of RT in high dimensional input domains.

1. Introduction

Effective test case selection strategies are essential to increase the chance of detecting failures and reduce the cost of software testing process. *Random Testing* (RT) is a simple test case selection strategy that treats the *Software Under Test* (SUT) as a black box [11]. In RT, test cases are selected randomly from the *input domain* (that is, the set of all possible inputs) of the SUT. Despite the criticism that RT may be ineffective as it does not make use of the information about the SUT [15] or previously executed test cases, RT has been a popular and successful testing method in many applications [8][9][10][12][13][14][16] as it is simple in concept and easy to implement.

Previous studies have shown that failure-causing inputs tend to be clustered in certain ways [1][2][9]. Regions formed by the failure-causing inputs are referred to as *failure regions* [1]. *Chan et al.* [3] classified these failure regions into three typical patterns: point, strip and block failure patterns.

Chen et al. [4] made use of the general information about the typical patterns of failure regions to improve the fault-detection effectiveness of RT. They found that when the failure-causing inputs cluster in a block or a reasonably thick strip area, the chance of detecting the first failure can be magnified by spreading the test cases widely and evenly across the input domain. This test case selection approach is known as *Adaptive Random Testing* (ART). ART can be implemented using the Fixed-Size-Candidate-Set (*FSCS*) algorithm [4].

In a study of the fundamental factors that may affect the fault-detection effectiveness of FSCS-ART [7], it was found that the performance of FSCS-ART deteriorates when the dimensionality of the input domain (that is, the number of input parameters) increases. Since real-life applications may have many input parameters, it is essential to create new algorithms that can improve the fault-detection effectiveness of RT in high dimensional input domains.

In this paper, we propose two enhanced RT algorithms for high dimensional input domains. These algorithms are designed based on our analysis on the locations of FSCS-ART test cases and the failure regions in high dimensional input domains. In addition, the fault-detection effectiveness of the proposed enhanced RT algorithms is evaluated through simulation experiments.

The rest of this paper is organized as follows: the next section presents the notations and evaluates the fault-detection effectiveness of FSCS-ART in high dimensional input domains. Section 3 analyzes the locations of FSCS-ART test cases and the failure regions

[*] Contact Author

in high dimensional input domains. Then Section 4 proposes two enhanced RT algorithms and evaluates their fault-detection effectiveness against RT through simulation experiments. Section 5 concludes the paper.

2. Background

In this section, we present the preliminaries of RT and ART performance evaluation and the problems associated with FSCS-ART algorithm in high dimensional input domains.

2.1 Preliminaries

Assume that a SUT has a set of n input parameters $\{x_1, x_2, \ldots, x_n\}$, where x_i (i=1, 2, …, n) has a bounded range $0 < x_i \le d_i$ (the situation where $u < x_i \le v$ and $u \neq 0$ can be mapped to $0 < x_i \le v - u$ as this mapping does not change the shape of the input domain). For an n-dimensional bounded input domain **D**, the size of the input domain is defined as $|\mathbf{D}| = \prod_{i=1}^{n} d_i$. With a failure region of size m, the failure rate Θ of the SUT is then defined as $\Theta = m/|\mathbf{D}|$.

F-measure [6] has been used to evaluate the performance of ART (in this paper, the term "fault-detection effectiveness" and "performance" are used interchangeably). F-measure is defined as the number of test cases required to detect the first failure. Let F_{ART} and F_{RT} denote the F-measure of ART and RT, respectively. Since ART is an enhanced version of RT, the *ART F-Ratio* (= F_{ART} / F_{RT}) was introduced to serve as the comparison metric to show how much improvement ART has over RT. Obviously, the smaller the ART F-Ratio is, the better the performance of ART will be.

For RT with uniform usage profile and replacement [6], the theoretical F_{RT} mean is $1/\Theta$. The F_{ART} can be obtained via empirical study. To obtain a statistically significant F_{ART} mean, all simulations were repeated until the F_{ART} mean has an accuracy range of 5% and a confidence level of 95%. For further details, please refer to [4].

ART is known to have the best performance in the block failure pattern [7]. In this paper, we will confine both empirical and analytical studies to a single block failure region with equal size for each of its dimensions.

2.2 Performance of FSCS-ART in high dimensional input domains

Figure 1 shows the performance of FCSC-ART when it is applied in 1(1D), 3(3D), 6(6D) and 9(9D) dimensional input domains for Θ ranging from 0.75 to 0.0005. For ease of presentation, Θ is plotted on a $\log_{0.5}$ scale.

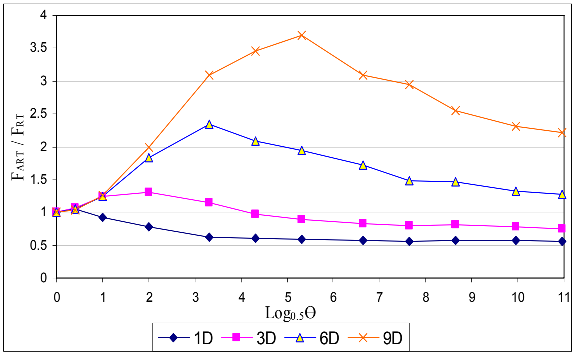

Figure 1: The ART F-ratios of FSCS-ART for input domains of different dimensionalities.

The experiment results in Figure 1 show that the existing FSCS-ART does not perform well when the failure rate is large or the number of dimensions of SUT is large. This observation was explained as a consequence of the "edge bias" of FSCS-ART in [7]. As the number of dimensions of the input domain increases, the performance of FSCS-ART deteriorates more significantly. Therefore, it is necessary to further investigate the "edge bias" problem and its impact on the performance of FSCS-ART.

3. Why the Performance of FSCS-ART Deteriorates in High Dimensional Input Domains?

To investigate the above problem, we examine the test case distribution of FSCS-ART and analyze the location of the failure region in the input domains.

3.1 Test case distribution of FSCS-ART

Based on the definitions in Section 2.1, we additionally define a *centre region* and an *edge region* within an n-dimensional input domain as follow.

The centre region, $\mathbf{D_c}$, is a sub-region of input domain **D** which shares the same centre with the input domain (denoted by O). The size of the centre region $\mathbf{D_c}$ is 50% of the size of input domain **D** (that is, $|\mathbf{D_c}| = 0.5 \times |\mathbf{D}|$). The width for each dimension of the centre region, cd_i, is given by $cd_i = \sqrt[n]{0.5}\, d_i$. The edge region, $\mathbf{D_e}$, is defined as the non-centre region in the input domain. The size of the edge region is equal to the centre region (that is, $|\mathbf{D_e}| = |\mathbf{D_c}| = 0.5 \times |\mathbf{D}|$). The "width" for each dimension of the edge region, ed_i, is defined as $ed_i = \left(1 - \sqrt[n]{0.5}\right) d_i / 2$.

Having introduced the concepts of the centre and edge regions, we would like to compare the numbers of test cases distributed over these two regions. Let *edgeCount* be the number of test cases located in the edge region, and *centreCount* be the number of test cases located in the centre region. Let $R_{E,C}$ be the ratio of *edgeCount* to *centreCount*. In our experiment, whenever FSCS-ART generates a test case, we will check its location and update the corresponding *centreCount* or an *edgeCount* accordingly. As the two regions are equal in size, an $R_{E,C}$ greater than 1 implies more test cases are being selected from the edge region. We conducted experiments to observe the values of $R_{E,C}$ produced by FSCS-ART for input domains with different dimensionalities when the number of test cases changed from 1 to 10,000. The results are shown in Figure 2.

From Figure 2, we can observe that, firstly, $R_{E,C}$ is almost always greater than 1, which means that FSCS-ART selects more test cases from the edge region than from the centre region. Secondly, $R_{E,C}$ becomes larger as the number of dimensions of the input domain increases. Intuitively, this is because the higher the dimensionality, the more corners/edges the input domain has, which will be filled up first by the test cases generated by the FSCS-ART algorithm. Thirdly, $R_{E,C}$ increases to a peak and then fluctuates prior to decreasing gradually towards limit 1.

Before we further analyze the impact of such "edge biased" test case distribution on the performance of FSCS-ART, we would like to examine the location of the failure region in the input domain.

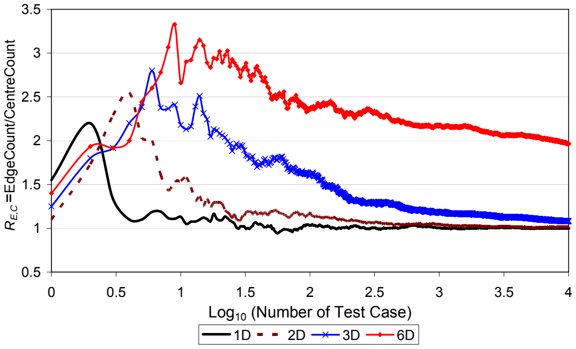

Figure 2: The ratio $R_{E,C}$ produced by FSCS-ART when the number of test cases ranged from 1 to 10,000

3.2 The location of the failure region in the input domain

When the number of dimensions of the input domain increases, the width of the centre region will increase, and in turn the width of the edge region will decrease. This observation brings up an important question: for a specific failure rate, as the edge width becomes narrower in high dimensional input domains, would the failure-causing inputs (points in a randomly located block failure region) have an equal chance of falling into the centre region and the edge region?

Let the *n*-dimensional input domain be homogeneous and have a unit size, that is, |D|=1 and d_i=1 (i = 1, 2, …, n). The size of a failure region F is given by |F|= Θ×|D|. Assuming that the failure region has the same orientation as the input domain and let the failure region be a block with equal width for each of its dimensions. Each dimension of the failure region is denoted by f_i (i = 1, 2, …, n), with the width $|f_i| = \sqrt[n]{\theta|D|}$. Figure 4 shows how $|f_i|$ varies against the number of dimensions for the failure rates Θ=0.005 and Θ=0.0005.

Figure 3 shows that, for the same failure rate Θ, when the number of dimensions of the input domain increases, $|f_i|$ will also increase but the width of the edge region, ed_i, will decrease. When Θ=0.0005, $|f_i|$ will become greater than ed_i for input domains of 4 dimensions and above; for a higher failure rate, say Θ=0.005, this will happen more quickly. When $|f_i| > ed_i$, it means that even in the situation where the failure region is attached to a border of the input domain, part of the failure region will still fall into the centre region. Therefore, the probability distribution for the location of failure region within the input domain warrants further analysis.

Referring to Figure 4, consider how f_i of different sizes can be located fully or partially within the centre region and edge region of dimension i. Let $P_{centre,i}$ denote the probability that some or all the elements of f_i fall into the centre region of dimension i. Similarly, the probability that some or all elements of f_i fall into the edge region of dimension i is denoted by $P_{edge,i}$. Equations (1) and (2) define $P_{centre,i}$ and $P_{edge,i}$ respectively.

$$P_{\text{centre, i}} = \begin{cases} \dfrac{b-a}{d_i - |f_i|} & 0 < |f_i| \le a \\ 1 - \dfrac{\dfrac{a^2}{|f_i|}}{d_i - |f_i|} & a < |f_i| \le b \\ \dfrac{(b-a)\left(\dfrac{d_i}{|f_i|}+1\right)}{d_i - |f_i|} & b < |f_i| \le d_i \end{cases} \tag{1}$$

$$P_{\text{edge,i}} = \begin{cases} 1 - \dfrac{(b-a)}{d_i - |f_i|} & 0 < |f_i| \le a \\ \dfrac{\dfrac{a^2}{|f_i|}}{d_i - |f_i|} & a < |f_i| \le b \\ 1 - \dfrac{(b-a)\left(\dfrac{d_i}{|f_i|}+1\right)}{d_i - |f_i|} & b < |f_i| \le d_i \end{cases} \tag{2}$$

For an n-dimensional input domain, the probability that some or all elements of the failure region are located in the centre region is denoted as P_{centre}. On the other hand, the probability that some or all elements of the failure region are located in the edge region is denoted as P_{edge}. Assuming that the input domain is homogeneous (that is, $d_1=d_2=...d_n$), P_{centre} and P_{edge} can be simplified into Equations (3) and (4), respectively.

$$P_{centre} = \prod_{i=1}^{n} P_{centre,i} \qquad (3)$$

$$P_{edge} = \prod_{i=1}^{n} P_{edge,i} + \sum_{i=1}^{n-1} {}^{n}C_i \left(P_{edge,i}\right)^{i} \left(P_{centre,i}\right)^{n-i} \qquad (4)$$

Figure 5 plots the ratio P_{centre}/P_{edge} against the failure rates, Θ, from 0.75 to 0.0005. When $P_{centre}/P_{edge} > 1$, it indicates that elements of failure region have a higher probability to occupy the centre region than the edge region. As mentioned earlier, when the dimensionality of the input domain increases, $R_{E,C}$ becomes higher, which indicates that FSCS-ART will select more and more test cases from the edge region than from the centre region. However, at the same time, the ratio P_{centre}/P_{edge} also becomes higher, which means that the failure region has higher probability to occupy the centre region than the edge region when the dimensionality of the input domain increases. As a result, FSCS-ART becomes less effective in high dimensional input domain.

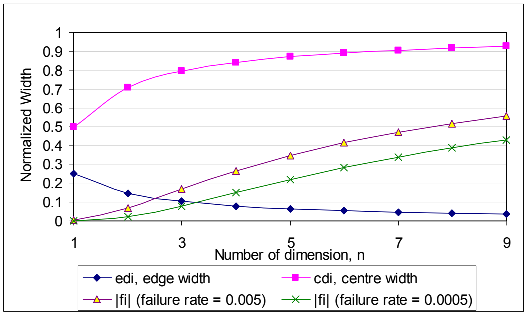

Figure 3: Widths of the edge region, centre region and failure region in different dimensionalities

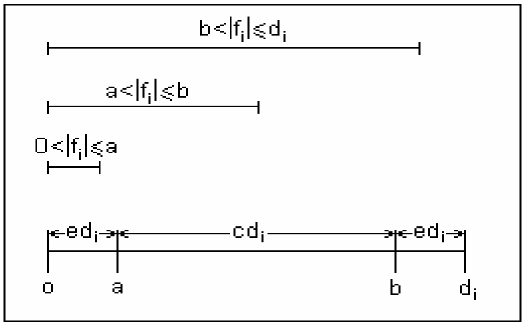

Figure 4: f_i of different sizes compared to the centre and edge widths.

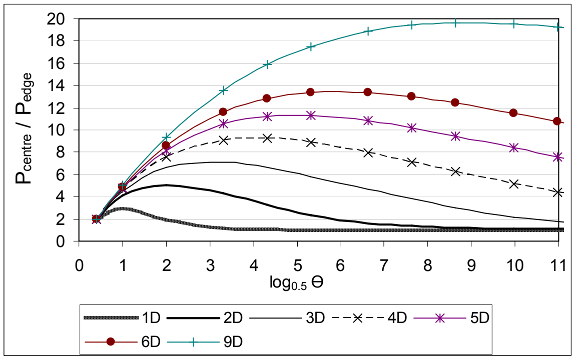

Figure 5: P_{centre} / P_{edge} for input domain of different dimensionalities

4. Enhanced RT Algorithms for High Dimensional Input Domains

The above analysis gives us an inspiration that if we select more test cases from the region where failure-causing inputs are more likely to fall into, then we will achieve better fault-detection effectiveness. In this section, we propose two methods (algorithms) to improve the fault detection effectiveness of RT in high dimensional input domains and report their performance through simulation experiments.

4.1 Inverted FSCS-ART

Since FSCS-ART selects more test cases from the edge region than from the centre region in high dimensional input domains, a simple approach to improve the fault detection effectiveness is to invert the edge/centre distribution of FSCS-ART test cases. This can be done by mapping the FSCS-ART test cases from the edge to the centre region and vice versa before executing them. We name this method *Inverted FSCS-ART*. This algorithm is outlined in Figure 6. Note that the core of the FSCS-ART test case selection algorithm remains unchanged. Equation (5) is one of the linear functions that can map the FSCS-ART test cases from the edge to the centre region and vice versa. Note that x_i is one of the input parameters (that is, one of the dimensions) in the input domain and $0 < x_i \le d_i$.

$$f(x_i) = \begin{cases} x_i + d_i/2 & 0 < x_i \le d_i/2 \\ x_i - d_i/2 & d_i/2 < x_i \le d_i \end{cases} \qquad (5)$$

To evaluate the performance of Inverted FSCS-ART, simulations were conducted for failure rates, Θ, ranging from 0.75 to 0.0005 for n-dimensional input domains where n = 1, 3, 6 and 9. The simulation results in Figure 7 show that Inverted FSCS-ART outperforms RT for all failure rates and dimensionalities of input domains under the study. The performance of Inverted FSCS-ART in 1D input domain is very similar to that of FSCS-ART.

However, for 3D input domain, it can be observed that the F_{IART}/F_{RT} ratio falls to a minimum before settling at 0.7. Similar trend can be observed for 6D and 9D input domains where the F_{IART}/F_{RT} minimums occur at smaller failure rates, Θ (that is, higher values on $\log_{0.5}\Theta$ scale). This observation concurs with the higher P_{centre}/P_{edge} and $R_{E,C}$ ratios in higher dimensional input domains. As the failure region has increasing chances of occupying the centre region in high dimensional input domains, inverting the edge-biased FSCS-ART test case distribution will increase the chance of detecting the failures region.

```
1. Initialize $E$ as an empty set.
2. Randomly choose a test case $t$. Add $t$ to $E$.
3. Test the SUT using test case $t$.
4. If a failure is detected, testing is stopped and debugging
   may start. Otherwise go to step 5.
5. Randomly generate $k$ candidates from the input domain to
   form a candidate set $C$, where $k$ is a constant integer
   greater than 0. The value of k was set to 10 in our
   experiments.
6. Find $c \in C$ such that, among all the elements in $C$, $c$ has
   the longest distance to its nearest neighbor in $E$.
7. Add $c$ to $E$.
8. Map $c$ to $c'$ using Equation (5).
9. Test the SUT using test case $c'$.
10. If testing resources are not exhausted, go to step 4.
```

Figure 6: Inverted FSCS-ART algorithm

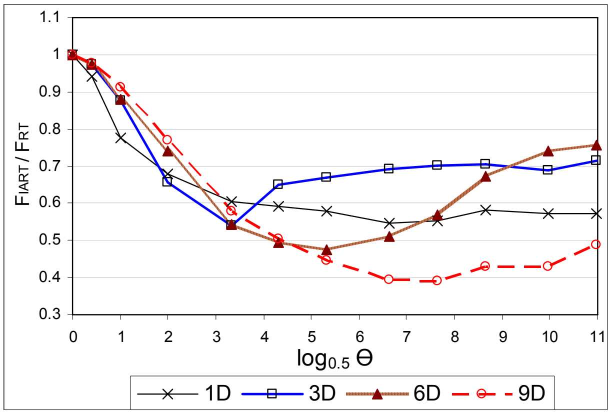

Figure 7: The ratio of the F-measure mean of Inverted FSCS-ART to the F-measure mean of RT.

4.2 Proportional Random Testing

The P_{centre}/P_{edge} ratio provides a useful guideline for the number of test cases that should be selected in the centre and in the edge region. Ideally, $1/(R_{E,C})$ should be in proportion to, and, as close as possible to the P_{centre}/P_{edge} ratio. Unfortunately, failure rate is unknown during testing. Therefore, it is impossible to determine the P_{centre}/P_{edge} ratio. However, the failure rate can be projected dynamically based on the number of test cases that have been executed. By taking the theoretical F-measure mean of random testing, after j test cases have been executed, the failure rate can be projected as $\Theta_{projected}=1/(j+1)$, assuming that next test case will detect the first failure. The ratio P_{centre}/P_{edge} can then be estimated based on $\Theta_{projected}$.

We propose the following algorithm in Figure 8 to select test cases randomly in the centre and in the edge region in proportion to the ratio P_{centre}/P_{edge}. We name this algorithm as "*Proportional Random Testing*" (PRT).

```
1. Initialize $j$ to 0, edgeCount to 1, centreCount to 1, and
   selectTestCaseInCentre to true, where $j$ is the number of
   test cases executed.
2. If selectTestCaseInCentre is true, randomly select and
   execute a test case in the centre region. Otherwise,
   randomly select and execute a test case in the edge
   region.
3. Increment $j$ by 1. If selectTestCaseInCentre is true,
   increment centreCount by 1. Otherwise, increment
   edgeCount by 1. Update $R_{E,C}$.
4. If a failure is detected, testing is stopped and debugging
   may start. Otherwise go to step 5.
5. Update the projected failure rate as $\Theta_{projected}=1/(j+1)$.
6. Calculate the ratio $P_{centre}/P_{edge}$ based on $\Theta_{projected}$.
7. If $1/(R_{E,C}) < P_{centre}/P_{edge}$, set
   selectTestCaseInCentre=true. Otherwise, set
   selectTestCaseInCentre=false.
8. If testing resources are not exhausted, go to step 2.
```

Figure 8: Proportional Random Testing algorithm

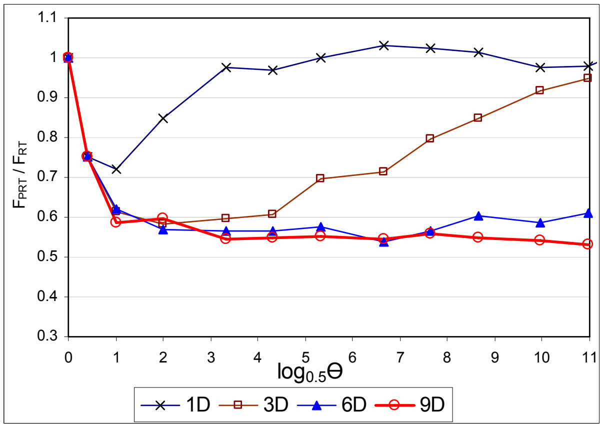

Figure 9: The ratio of the F-measure mean of Proportional Random Testing (PRT) to the F-measure mean of RT.

To evaluate the performance of the Proportional Random Testing, simulations were conducted for failure rates, Θ, ranging from 0.75 to 0.0005 for n-dimensional input domains where n = 1, 3, 6 and 9. The simulation results in Figure 9 show that Proportional Random Testing outperforms RT significantly when the number of dimensions is high (that is, when the P_{centre}/P_{edge} ratio is sufficiently large). As the dimensionality of the input domain increases, it can be observed that F_{PRT}/F_{RT} approaches 1 at smaller failure rates, Θ (that is, higher values on $\log_{0.5}\Theta$ scale). However, this algorithm is only as good as RT when the number of dimensions is low (that is, when $P_{centre}/P_{edge} \approx 1$).

5. Conclusion

In this paper, we proposed two enhanced RT algorithms for high dimensional input domains based on our analysis on two fundamental reasons that cause the performance of FSCS-ART to deteriorate in high dimensional input domains.

Proportional Random Testing is superior to Inverted FSCS-ART in computational cost for test case generation because Inverted FSCS-ART inherits the high computational cost in test case selection from FSCS-ART [5]. However, Proportional Random Testing does not give significant improvement over RT when the number of dimensions is low. Therefore, Proportional Random Testing should be used to generate test cases only when the number of dimension is high (that is, more than 3). On the other hand, Inverted FSCS-ART does not suffer from this setback. It can be used as a generic algorithm to generate test cases for programs with an input domain of any dimensionality.

For future work, we intend to evaluate the performance of the proposed algorithms for other patterns of failure regions in high dimensional input domains.

6. Acknowledgements

This research is partially supported by an Australia Research Council Discovery grant (Project No. DP0557246) and a Small Grant of the University of Wollongong.

7. References

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

[2] P.G. Bishop. The variation of Software Survival Times for different operation input profiles, *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, IEEE Computer Society Press, pp.98-107, 1993

[3] F.T. Chan, T.Y. Chen, I.K. Mak, and Y.T. Yu. Proportional Sampling Strategy: Guidelines for Software Testing Practitioners, *Information and Software Technology*, 38, (12), pp.775-782, 1996.

[4] T.Y. Chen, H. Leung, and I.K. Mak. Adaptive Random Testing. *Proceedings of the 9th Asian Computing Science Conference, LNCS 3321*, Springer-Verlag, pp.320-329. 2004.

[5] T.Y. Chen, F.-C. Kuo, R.G. Merkel and S.P. Ng, Mirror Adaptive Random Testing, *Information and Software Technology*. 46(15):pp.1001-1010. 2004.

[6] T.Y. Chen, F.-C. Kuo, R. Merkel. On the Statistical Properties of the F-measure, *Proceedings of the Fourth International Conference on Quality Software*. pp. 146 – 153. 2004,

[7] T.Y. Chen, F.-C. Kuo, and Z.Q. Zhou. On the Relationships between the Distribution of Failure-causing Inputs and Effectiveness of Adaptive Random Testing, *Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering*, pp.306-311, 2005.

[8] R. Cobb and H. D. Mills. Engineering software under statistical quality control. *IEEE Software*, 7(6):45-54, 1990.

[9] G.B. Finelli. NASA Software Failure Characterization Experiments, *Reliability Engineering and System Safety*, IEEE Computer Society Press, 32(1-2):pp.155-169, 1993.

[10] J. E. Forrester and B. P. Miller. An empirical study of the robustness of Windows NT applications using random testing. *Proceedings of the 4th USENIX Windows Systems Symposium*, pp. 59−68, Seattle, 2000.

[11] R. Hamlet. Random Testing, *Encyclopedia of Software Engineering*, Wiley, New York, pp. 970-978. 1994.

[12] N. Nyman. In defense of monkey testing: Random testing can find bugs, even in well engineered software. http://www.softtest.org/sigs/material/nnyman2.htm, *Microsoft Corporation*.

[13] B. P. Miller, D. Koski, C. P. Lee, V. Maganty, R. Murthy, A. Natarajan, and J. Steidl. Fuzz revisited: A reexamination of the reliability of UNIX utilities and services. Technical Report CS-TR-1995-1268, University of Wisconsin, 1995.

[14] H.D. Mills, M. Dyer, and R.C. Linger, “Cleanroom Software Engineering”, *IEEE Software*, 3, pp.19-24, 1986.

[15] G.J. Myers. *The Art of Software Testing*. John Wiley and Sons, Inc, ISBN0-471-04328-1. 1979.

[16] T. Yoshikawa, K. Shimura, and T. Ozawa. Random program generator for Java JIT compiler test system. *Proceedings of the 3rd International Conference on Quality Software*, pp.20−24. IEEE Computer Society Press, 2003.