# Which Type of Exclusion Region is Better for Restricted Random Testing? An Empirical Study

Weifeng Sun, Rubing Huang*, Chenhui Cui, Haibo Chen, Weijie Liu
School of Computer Science and Communication Engineering, Jiangsu University
Zhenjiang, Jiangsu, China
rbhuang@ujs.edu.cn

## ABSTRACT

As an enhancement of Random Testing (RT), Adaptive Random Testing (ART) has been well studied, which aims at achieving well-performing distribution of random test cases. Previous studies have indicated that ART can effectively enhance the failure-detection ability of RT, and many implementations of ART have been proposed based on different notations. Among them, Restricted Random Testing (RRT) is a popular version of ART, which generates test cases outside the exclusion regions (i.e., the constructed regions that are located around the executed but no failure-causing inputs). As we know, there are many factors that may influence the performance of RRT such as the size and shape of exclusion region. In this paper, we conducted a series of simulations to investigate the impact that the type of exclusion region has on the testing effectiveness of RRT, i.e., Which type of exclusion region is better for RRT? The results show that when the failure-causing inputs cluster into a few failure regions or a large predominant failure region exists within the input domain, the uniform exclusion region is a good choice for RRT. However, when the failure region is less compact, it appears to be less suitable, because the irregular exclusion region has much better performances.

## CCS CONCEPTS

• **Software and its engineering** → **Software testing and debugging**;

## KEYWORDS

adaptive random testing, restricted random testing, exclusion region, empirical study

**ACM Reference Format:**
Weifeng Sun, Rubing Huang*, Chenhui Cui, Haibo Chen, Weijie Liu. 2021. Which Type of Exclusion Region is Better for Restricted Random Testing? An Empirical Study. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3412841.3442128

*SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea*
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8104-8/21/03.
https://doi.org/10.1145/3412841.3442128

## 1 INTRODUCTION

*Software testing* is an essential part of the software development process, meanwhile providing important contributions to software quality assurance. *Random Testing* (RT) [11], one fundamental approach, generates test cases in a random manner from the input domain (i.e., the set of all possible program inputs). Although RT has been used extensively in industry and academia [2, 9, 12], its testing effectiveness is an ongoing controversy [10].

Many new strategies have been proposed to enhance the effectiveness of RT. For example, *Adaptive random testing* (ART) [6] increases the diversity of test cases by distributing them evenly across the input domain. A version of ART, namely Restricted Random Testing (RRT) [5], exploits an exclusion strategy to enhance the failure detection capability of RT. More specifically, the corresponding *exclusion region* (ER) is created for each *executed test cases* which have been executed but not find failures. Subsequent test cases are then drawn from outside these ERs. RRT ensures a minimum distance among all test cases, thereby achieving an even spreading of test cases.

In general, the uniform ER is commonly used for the RRT method, e.g., circular and spherical in 2D and 3D input space, respectively. As we know, the performance of RRT is heavily dependent on the ER, such as the size and shape of exclusion region, and there are many variations of the ER. Although it seems to be consensus on the choice of ER [4, 5], to date, no studies have specifically explored which type of exclusion region is better for RRT. To answer this question, in this paper, we conducted a series of simulations to investigate the impact of the ER and its many variations on the testing effectiveness of RRT. The results show that when the failure-causing inputs cluster into a few failure regions or a large predominant failure region exists within the input domain, the uniform ER is a good choice for RRT, but that irregular ERs may be better for the failure region with low compactness.

The rest of this paper is organized as follows: some background information is shown in Section 2. Section 3 presents the setting of the experiments; while Section 4 involves the discussions of the results. Finally, Section 5 summarizes the paper.

## 2 BACKGROUND

In this section, we present some background information about ART [6, 8] and the RRT [5].

## 2.1 Adaptive random testing

ART [6, 8] aims to enhance the failure detection capability of RT by incorporating the additional mechanism that the test cases should have an even distribution. ART skillfully utilizes the observation of the continuity of the failure region. If the failure region is contiguous, it can be concluded that non-failure-regions should also be contiguous. Therefore, if a test case $t_i$ is not a failure-causing input, then the test cases near $t_i$ are less likely to trigger failures than that far away from $t_i$. In other words, test cases far from the executed test cases should have a high probability of detecting a failure. Many different ART algorithms have been proposed [8] due to various notions and intuitions of even spreading, and RRT is one of the implementations of ART.

## 2.2 Restricted random testing

When testing through the RRT method [5], the test cases are generated following the exclusion strategy, except that the first test case is randomly selected from the whole input domain. More specifically, according to each executed test case, RRT creates its ER, and the next test case is forbidden to be selected from any ER. By employing the exclusion mechanism, RRT can guarantee a minimum distance between all test cases, that is, the radius of the ER.

The ER has a necessary impact on the performance of the RRT. Intuitively, RRT with various types of ER should have distinctive performances under different *failure patterns*. However, to date, no studies have specifically explored which type of exclusion region is better for RRT. Aiming at examining the effect of different ERs on the effectiveness of RRT, in this paper, we conducted a series of experiments.

# 3 EXPERIMENTAL STUDIES

In this section, we report on a series of simulations to investigate how different ERs influence the performance of RRT and thus answer the research question of which type of exclusion region is better for RRT. The detailed design and settings of the experiments are described as follows.

## 3.1 Experiments Objects

In this paper, the variations of ER is the independent variable for the experimental studies. We selected two uniform ERs: the circle and square exclusion region (abbreviated as CER and SER respectively), which have been widely adopted in previous research [1, 5]. In addition, we complemented a rectangle exclusion region (denoted by RER) with different compactness. The $\alpha$ ($\alpha \geq 1$) is to describe the compactness of RER. In other words, the ratio among edge lengths of the rectangular region can be described as $1 : \alpha$, where $\alpha$ is set as 2 and 80. Intuitively, the smaller $\alpha$ is, the more compact the RER. Incidentally, the SER can be considered as the special case of RER (i.e, $\alpha = 1$).

## 3.2 Experiments Setup

The studies attempt to mimic faulty programs in different situations. Generally, there are two basic features for a faulty program, including the *failure rate* and *failure pattern* [3]. The failure rate, normally denoted by $\theta$, refers to the ratio of the number of failure-causing inputs to the number of all possible inputs. In addition, the shape of the failure region together with its distribution over the input domain is called failure pattern. In our simulations, $\theta$ and failure patterns were predefined, and then the failure regions were randomly placed in the input domain. By the way, the size of exclusion region is not focus of this paper, thus, the exclusion ratio was uniformly set to 75%. If a point is selected from inside any failure region, a failure is said to be detected. For a $d$-dimensional input domain $D$, without loss of generality, we assume that $D = \{(x_1, x_2, \cdots, x_d) | 0 \leq x_i < 1.0, i = 1, 2, \cdots, d\}$, abbreviated as $[0, 1.0)^d$. Following the previous studies [7], we simulated the following four different failure patterns (FPs).

**FP-I:** The first failure pattern was set as the block pattern, which involved a single square/hypercubic failure region. The $\theta$ was set as 0.75, 0.5, 0.25, 0.1, 0.075, 0.05, 0.025, 0.01, 0.0075, 0.005, 0.0025, 0.001, 0.00075, 0.0005, 0.00025, 0.0001, 0.000075, and 0.00005.

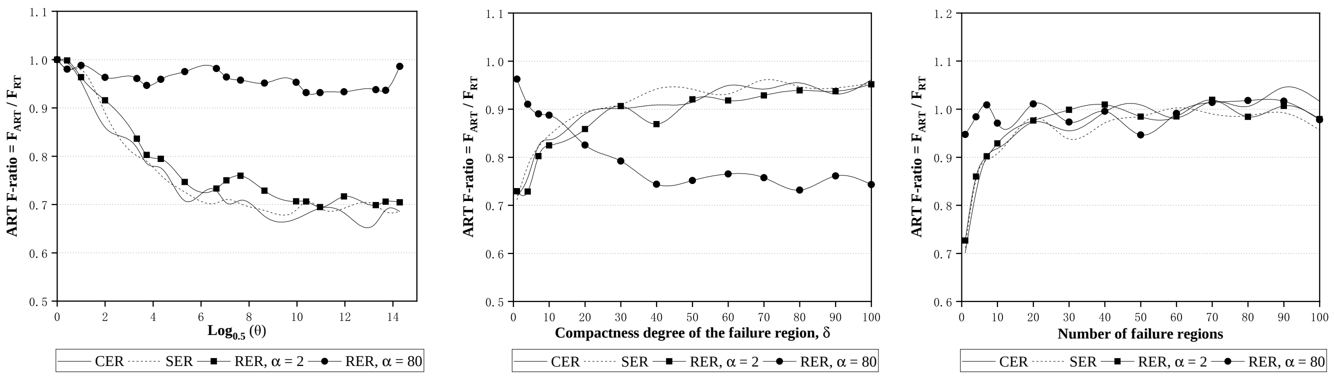
**FP-II:** The second failure pattern to be simulated is a single rectangular (strip) failure region randomly placed inside a two-dimensional input domain. Similar to RER, the $\delta$ describes the compactness of failure region, where $\delta$ is set as 1, 4, 7, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. In addition, $\theta$ was set as 0.005, and 0.001.

**FP-III:** The third failure pattern was set as a number of square regions (denoted by $R_1, R_2, \cdots, R_\beta$) with the equal size. $\beta$ is set as 1, 4, 7, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, respectively, and $\theta$ was set as 0.005 and 0.001.

**FP-IV:** In the fourth failure pattern, a number of square regions are randomly placed inside a two-dimensional input domain, but there exists a predominant region. Suppose that there are $\beta$ failure regions, denoted by $R_1, R_2, \cdots, R_\beta$, respectively. For one failure region $R_1$, we set $|R_1| = w \cdot \theta \cdot |D|$, where $w = 0.3, 0.5$, and 0.8. For all the other failure regions, we set $|R_i| = (\rho_i / \sum_{j=2}^{\beta} \rho_j) \cdot (1 - w) \cdot \theta \cdot |D|$, where $\rho_i$ is randomly selected from $[0, 1)$ according to uniform distribution, $i = 2, 3, \cdots, \beta$, and $|D|$ denotes the size of the input domain. The number of failure regions $\beta$ is set as 1, 4, 7, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, respectively. Similar to third failure pattern, $\theta$ was set as 0.005 and 0.001.

## 3.3 Evaluation Metrics

To evaluate and compare the failure-detection effectiveness of RRT under different ERs, we used the *F-measure* as the evaluation metric in this study, which also follows previous studies on ART [8]. The F-measure refers to the expected number of test cases required to find the first software failure. Meanwhile, $F_{RT}$ and $F_{ART}$ denotes the F-measure of RT and ART respectively. Finally, we used ART *F-ratio* to denote



(a) block pattern, $d = 2$

(b) strip pattern, $\theta = 0.005$, $d = 2$

(c) point pattern, $\theta = 0.005$, $d = 2$

**Figure 1: ART F-ratio results of RRT with different ERs on three failure patterns**

the ratio of $F_{ART}$ to $F_{RT}$, which measures the F-measure improvement of ART over RT. Obviously, a smaller F-ratio means better ART performance. Because of the randomness in ART, we ran each simulation 3000 times, and report the average result.

## 4 RESULTS

Due to the page limitation, only those results for some specific simulation settings were provided in this section[1].

### 4.1 FP-I

The simulation results regarding Experiment FP-I are reported in Figure 2(a), where $x$-axis represents $\theta$ in the logarithmic scale, and $y$-axis stands for the F-ratio. From the results of Figure 2(a), we can make the following observations:

(1) As the failure rate decreases, CER and high compactness RER (i.e., $\alpha = 1, 2$) perform better. However, this tendency is not evident for low compactness RER (i.e., $\alpha = 80$).

(2) Overall, CER and SER usually can obtain satisfactory results on a single square failure region, which means that the block pattern is a favorable condition for the uniform ERs. However, the CER has a slightly better performance compared to SER.

(3) As for the RER and its variations, we can observe that the performance of RRT gradually deteriorates along with the compactness decreases.

### 4.2 FP-II

Figure 1(b) shows the results of F-ratio for FP-II. As shown in Figure 1(b), we can observe that:

(1) Although using different ERs, the failure-detection effectiveness of RRT is also affected by the compactness of the failure region. More specifically, when the failure region is less compact, CER and high compactness RER has poorer performances; On the contrary, low compactness RER gradually has better performance. In particular, low compactness RER performances best among ERs when $\delta > 20$, with the differences becoming larger.

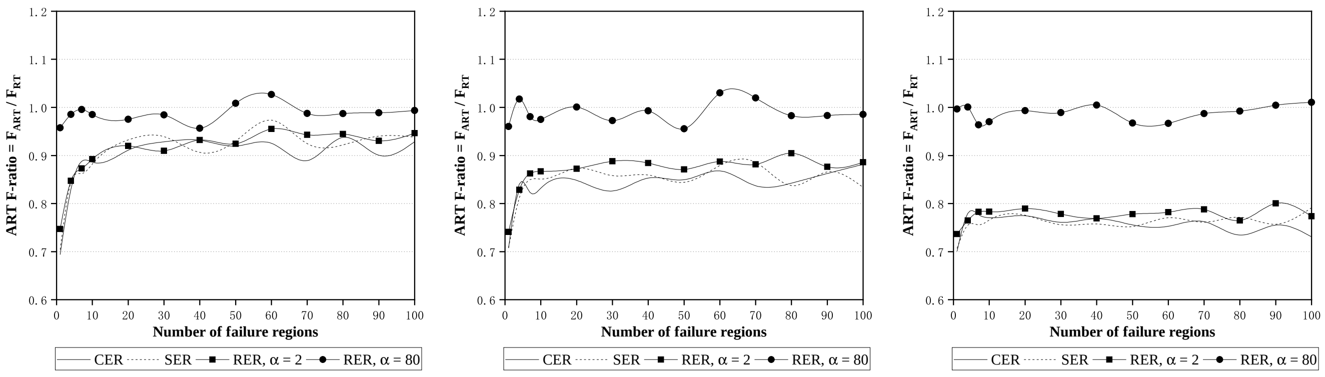
(2) In contrast to the observations of FP-I, for the RER and its variations, the performance of RRT gradually improves with the compactness of ER decreases.

Interestingly, low compactness RER obtains much better results than traditional ERs (i.e., CER and SER), when the failure region is less compact. Next, we will provide underlying reasons for this phenomenon. As mentioned in Section 2.2, R-RT ensures the minimum distance between test cases through the ER, thus achieving even test case distribution. However, this way of selecting test cases does not take dimensionality into consideration. The failure region becomes narrower when it is less compact, which means that some parameters may be less sensitive to failures than other parameters, even no related to failures. Therefore, besides the direct distance between test cases, the difference in each dimension is also important to measure the diversity of test cases. On the other hand, a low compactness RER prevents subsequent test cases from high similarity to previously executed test cases in certain dimensions, resulting in good failure-detection performance.

### 4.3 FP-III

The simulation results for FP-III are given in Figure 1(c), from which we can observe that when using the uniform ER, the F-measure of RRT becomes larger as the increase of the number of failure regions. As for low compactness RER, RRT has similar failure-detection capabilities to RT under any $\beta$. However, when the number of failure regions is small (such as, $1 \leq \beta \leq 20$), the uniform ER get a better performance than low RER with $\alpha = 80$.

[1] Readers may look up a complete set of results, which are available at https://github.com/huangrubing/SAC2021_RRT.



(a) $w = 0.3,\ \theta = 0.005$

(b) $w = 0.5,\ \theta = 0.005$

(c) $w = 0.8,\ \theta = 0.005$

**Figure 2: ART F-ratio results of RRT on multiple square failure regions with one predominant region**

### 4.4 FP-IV

Figure 2 reports the simulation results of RRT under different failure regions. The following observations we can observe are:

(1) In general, for the uniform ER or the high compactness RER, the F-measure of RRT depends on the number of failure regions and the size of predominant failure region. Overall, the CER and SER can get good F-measures. For low compactness RER, RRT has similar or slightly better performances than RT, regardless of $w$ and $\beta$.

(2) For the RER, the low compactness of ER provides negative contributions to the failure-detection effectiveness of RRT. In other words, the smaller compactness the RER is, the less the RRT performance is.

***To conclude:*** CER and SER achieve good results in most failure patterns, except for the single rectangle failure region. More specifically, when the failure-causing inputs cluster into a few failure regions or a large predominant failure region exists within the input domain, the CER and high compactness RER (including SER) take advantage in failure-detection; while for the RER, the less compactness, the worse the testing effectiveness. The case changes when the failure pattern is a single rectangle failure region: The low compactness RER has much better performance than the uniform ER, especially for low compactness of the failure region. Moreover, the difference is becoming larger as the compactness of the failure region decreases.

## 5 CONCLUSION

In this study, we conducted a series of simulations to investigate the impact of the exclusion regions (ERs) on the testing effectiveness of RRT, and hence answer which type of exclusion region is better for RRT. The results show that when the failure-causing inputs cluster into a few failure regions or a large predominant failure region exists within the input domain, the uniform ER is a good choice for RRT, but that irregular ER may be better for narrow failure region.

## ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China, under grant nos. 61502205 and 61872167, and the China Postdoctoral Science Foundation, under grant no. 2019T120396.

## REFERENCES

[1] H. Ackah-Arthur, J. Chen, D. Towey, M. Omari, J. Xi, and R. Huang. 2019. One-Domain-One-Input: Adaptive Random Testing by Orthogonal Recursive Bisection With Restriction. *IEEE Transactions on Reliability* 68, 4 (2019), 1404–1428.

[2] H. Bati, L. Giakoumakis, S. Herbert, and A. Surna. 2007. A Genetic Approach for Random Testing of Database Systems. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07)*. 1243–1251.

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

[4] K. P. Chan, T. Y. Chen, and D.e Towey. 2006. Forgetting Test Cases. In *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06)*. 485–494.

[5] K. P. Chan, T. Y. Chen, and D.e Towey. 2006. Restricted Random Testing: Adaptive Random Testing by Exclusion. *International Journal of Software Engineering and Knowledge Engineering* 16, 4 (2006), 553–584.

[6] T. Y. Chen, H. Leung, and I. K. Mak. 2004. Adaptive Random Testing. In *Proceedings of the 9th Asian Computing Science Conference (ASIAN'04)*. 320–329.

[7] R. Huang, H. Liu, X. Xie, and J. Chen. 2015. Enhancing mirror adaptive random testing through dynamic partitioning. *Information and Software Technology* 67 (2015), 13–29.

[8] R. Huang, W. Sun, Y. Xu, H. Chen, D. Towey, and X. Xia. to be published, 2019. A Survey on Adaptive Random Testing. *IEEE Transactions on Software Engineering* (to be published, 2019). https://doi.org/10.1109/TSE.2019.2942921.

[9] Woramet M. and Shingo T. 2017. Random GUI Testing of Android Application Using Behavioral Model. *International Journal of Software Engineering and Knowledge Engineering* 27, 9-10 (2017), 1603–1612.

[10] G. J. Myers. 2004. *The Art of Software Testing*. John Wiley & Sons: New York.

[11] A. Orso and G. Rothermel. 2014. Software Testing: A Research Travelogue (2000-2014). In *Proceedings of Future of Software Engineering (FOSE'14)*. 117–132.

[12] T. Yoshikawa, K. Shimura, and T. Ozawa. 2003. Random Program Generator for Java JIT Compiler Test System. In *proceedings of the 3rd International Conference on Quality Software (QSIC'03)*. 20–24.