Restricted Adaptive Random Testing by Random Partitioning

Johannes Mayer
Ulm University
Dept. of Applied Information Processing
89069 Ulm, Germany
Email: johannes.mayer@uni-ulm.de

***Abstract*— Adaptive Random Testing (ART) is designed to detect the first failure with fewer test cases than pure Random Testing. Since well-known ART methods, namely Distance-Based ART (D-ART) and Restriction-Based ART (RRT), have quadratic runtime, ART methods based on the idea of partitioning have been presented. ART by Random Partitioning is one of these partition-based ART algorithms. While having only a little bit more than linear asymptotic runtime, the number of test cases necessary to detect the first failure is substantially higher than that of D-ART and RRT. In the the present paper, an improved version of ART by Random Partitioning is presented employing the notion of restriction. The presented algorithm has the same very good runtime as ART by Random Partitioning while requiring significantly fewer test cases to exhibit the first failure.**

I. Introduction

Software testing [1] is an integral part of software quality assurance. Basically, the system under test is executed during the test with a lot of test cases. This task has to be automated in order to make testing efficient. However there are two important problems regarding automated software tests: The generation of test cases and the evaluation of test results. The present paper deals with the first problem.

Random Testing [2], [3], [4], [5], [6], i. e. testing with randomly generated test inputs, is a promising approach which has already been successfully applied in a lot of domains [7], [8], [9] to automate testing. However, Myers [1] criticized that Random Testing does not use information about the program under test. This can be seen as a hint for the improvement of Random Testing. Chan et al. [10] described three typical patterns of failure-causing inputs within the input domain. Figure 1 shows examples of these failure patterns each within a quadratic input domain. The regions shaded gray denote the failure patterns, i. e. the geometric shape of the failure-causing inputs within the input domain. Chan et al. also observed that the block and strip failure pattern are much more common than the point pattern, which only rarely occurs. Whereas these three failure patterns are only rough examples, one can say in general that failure-causing inputs tend to cluster within the input domain. This finding is independent of the system under test and can be used to speed up Random Testing. The basic idea is that a failure-causing input is less likely close to a non-failure-causing input. Therefore, test cases have to be selected wide-spread across the input domain. That is the (informal) intuition of Adaptive Random Testing (ART) [11], which has been implemented in different ways so far. Wide-spread test cases have first been implemented by distance-based (D-ART) [11], [12] and restriction-based (RRT) [13], [14], [15] approaches. They are very effective in terms of the mean F-measure, i. e. the average number of test cases necessary to detect the first failure. However, their runtime is at least asymptotically quadratic in the number of test cases executed. Therefore, Chen et al. [16] proposed algorithms based on partitioning. These algorithms are very fast—they have nearly linear runtime—, but their mean F-measure is not nearly as good as that of D-ART and RRT. These partition-based approaches have also been combined with D-ART resp. RRT [17], [18] employing the principle of localization. These algorithms are on the one hand very efficient, but on the other hand quite complex.

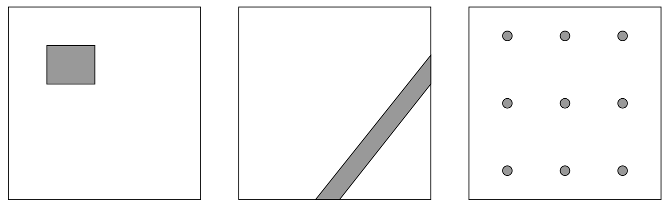

Fig. 1. Block, strip, and point failure patterns within a two-dimensional input domain

In the present paper, the principle of restriction is applied to ART by Random Partitioning. The resulting algorithm is quite simple, has nearly linear runtime, and its mean F-measure is in some cases even better than D-ART and RRT. With the presented algorithm, automation of Random Testing becomes feasible. Usually, the failure rate is very small which implies a large number of test cases. In this case, the gap between linear and quadratic runtime is huge. Furthermore, the presented algorithm also significantly reduces the number of test cases necessary to detect the first failure. In practice, one execution of the system under test is usually quite time-consuming—due to the non-trivial functionality (despite of test automation). Consequently, the test cases have to be chosen very carefully. The presented algorithm fullfills all these requirements.

The following section presents preliminaries regarding the notation. The improved algorithm is described in Section III.

An empirical evaluation of the proposed algorithm based on simulation is described and discussed in Section IV, followed by a conclusion.

II. Preliminaries

The input domain is assumed to be rectangular (for the sake of simplicity). The failure rate, i. e. the percentage of failure-causing inputs, is denoted θ. For a finite input domain of size d with m failure-causing inputs, $\theta = m/d$.

The *F-measure*—as already introduced—denotes the (random) number of test cases necessary to detect the first failure. This is a very natural measure for the performance of a testing strategy, since often testing is stopped when the first failure is detected. The F-measure has been used in all publications on ART. It is therefore ideal for comparison purposes.

For Random Testing with a uniform operational profile and replacement, the theoretical mean F-measure is equal to $1/\theta$. For example, for a failure rate of $\theta = 0.01$, the theoretical mean F-measure of Random Testing with replacement is 100, i. e. a failure will occur after the execution of 100 test cases on average.

The *relative F-measure* of a method is the F-measure of this method related to the theoretical mean F-measure of Random Testing, i. e. $1/\theta$. If a method has a mean F-measure no worse than Random Testing, its relative mean F-measure should always be below or at most 1. The advantage of the relative mean F-measure is that it is independent of the failure rate and can thus easily be compared for different failure rates.

III. The Algorithm

ART by Random Partitioning as proposed by Chen et al. [16] iteratively selects a test case from the largest partition (initially, the whole input domain) and subdivides this partition according to the test case. The algorithm starts with the whole input domain as the initial region and selects a test case (Test 1) from it (cf. Figure 2a). Thereafter, the region from which the test case has be chosen is replaced by four sub-regions induced by the test case as shown in Figure 2b. In the next step, the largest region is chosen (in this case Region 1) and a test case (Test 2) is randomly selected from it (cf. Figure 2c). Again, Region 1 is replaced by four sub-regions induced by Test 2 (cf. Figure 2d). Using an efficient implementation of a priority queue, such as a heap, for the regions, the algorithm can be implemented in time proportional to

$$\sum_{i=2}^{F} 5\log(i-1) = 5\log((F-1)!) = \mathcal{O}(\log F^F) = \mathcal{O}(F \log F),$$

where F denotes the number of test cases chosen. The algorithm has, thus, "nearly linear" runtime. However, this method is not suitable to avoid nearby test cases as Figure 3 demonstrates. Therefore, it fails to achieve wide-spread test cases in some situations. This explains, why this method performs not as good as D-ART and RRT regarding the mean F-measure.

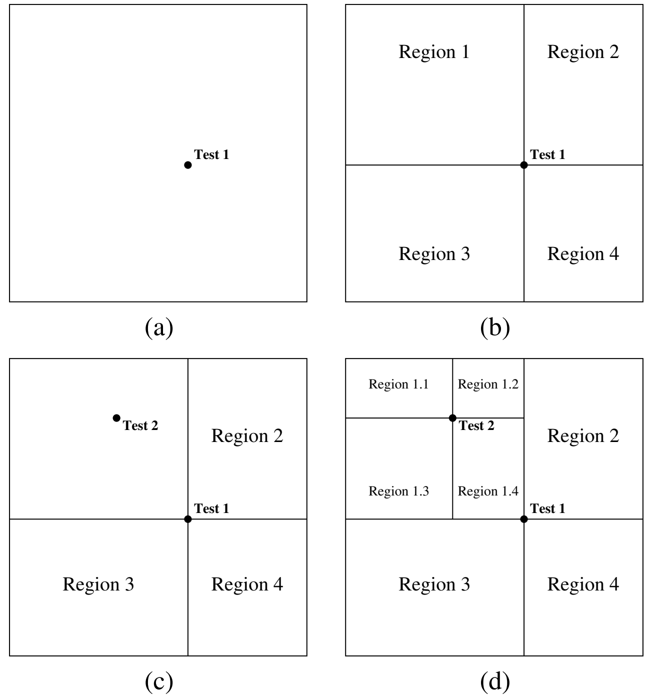

Fig. 2. ART by Random Partitioning: Some steps of the algorithm

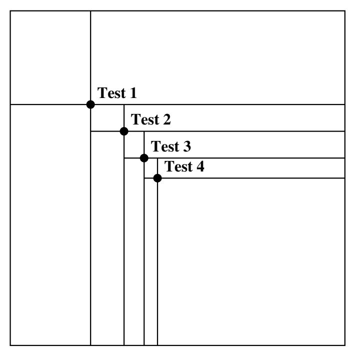

Fig. 3. Nearby test cases possible with ART by Random Partitioning

Using the idea of restriction, the following algorithm proceeds as ART by Random Partitioning, but it selects the test cases from restricted sub-domains. This restriction can easily be achieved and has the effect that the inputs have a (decreasing) minimum distance. Through this minimum distance, nearby consecutive test cases can be avoided other than with ART by Random Partitioning [16].

It is assumed that the two-dimensional input domain is rectangular with lower left corner $(x_{\min}, y_{\min})$ and upper right corner $(x_{\max}, y_{\max})$.[1] Therefore, the inputs are two-dimensional vectors (x, y) of real values with $x_{\min} \leq x \leq x_{\max}$ and $y_{\min} \leq y \leq y_{\max}$. It can trivially be adapted to a bounded region of integers or higher dimensional input

[1] Such rectangles are denoted $\{(x_{\min}, y_{\min})(x_{\max}, y_{\max})\}$ in the following.

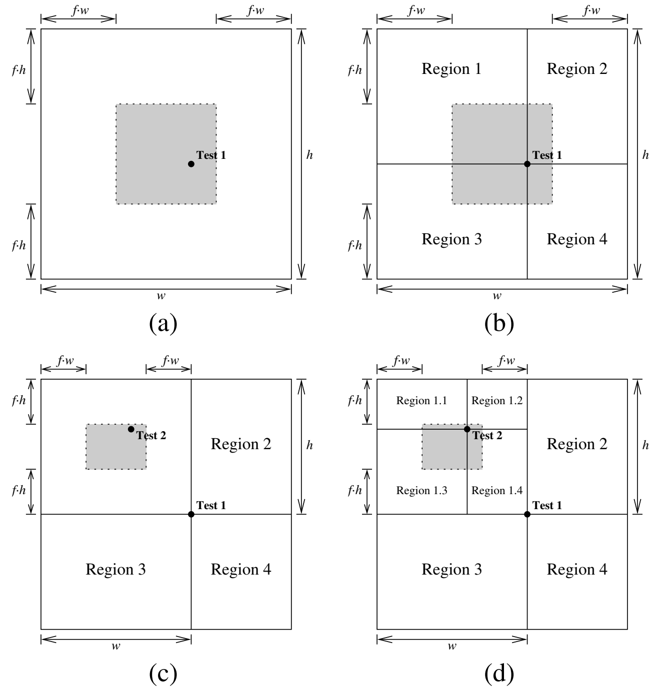

Fig. 4. Restricted ART by Random Partitioning: Some steps of the algorithm

domains. The exclusion factor f must be chosen from $[0, 0.5)$.

Algorithm 1: Restricted Adaptive Random Testing by Random Partitioning

1) Initialize the test region candidate list C with $\{\{(x_{\min}, y_{\min})(x_{\max}, y_{\max})\}\}$.
2) Select the test region $T = \{(x_0, y_0)(x_1, y_1)\}$ with maximal area[2] from C and remove it. If there are several such regions, choose one of them randomly. Let $w := x_1 - x_0$ be the width of T and $h := y_1 - y_0$ be the height of T.
3) Randomly select a point (x, y) from within the restricted test region $T' := \{(x_0+fw, y_0+fh)(x_1-fw, y_1-fh)\}$.
4) If the point (x, y) is a failure-causing input, report failure and terminate.
5) Otherwise divide the current test region $T = \{(x_0, y_0)(x_1, y_1)\}$ into four test regions $\{(x_0, y_0)(x, y)\}$, $\{(x_0, y)(x, y_1)\}$, $\{(x, y_0)(x_1, y)\}$, $\{(x, y)(x_1, y_1)\}$ and add them to C.
6) Proceed with step 2 unless the resources for testing are exhausted.

The algorithm starts with the whole input domain as the initial region (cf. Figure 4a). In each pass, a test case is randomly selected from a restriction of the biggest region. The biggest region is determined according to its area. After the selection of the test case, the region is subdivided into four sub-regions by the chosen test case. The original region is then replaced by the four sub-regions. In the following pass, again the biggest region is chosen and so on until a failure is detected or the ressources for testing are exhausted.

[2]The area of region $T = \{(x_0, y_0)(x_1, y_1)\}$ is computed as usual, i. e. $(x_1 - x_0)(y_1 - y_0) = w \cdot h$.

Figure 4 illustrates several steps of the algorithm. The initial region is the whole input domain. The algorithm selects the first test case (Test 1) from the restricted input domain shaded gray in Figure 4a. Thereafter, this region is subdivided into four regions by the first test case Test 1 (cf. Figure 4b). In the next step, the biggest region (Region 1) is chosen and a test case (Test 2) is randomly selected from within this region (cf. Figure 4c). Again, this region is replaced by the four induced sub-regions (cf. Figure 4d), and so on.

The runtime of the presented algorithm is the same as that of the original ART by Random Partitioning, $\mathcal{O}(F \log F)$, provided that a heap data structure or something like that is used to store the regions and retrieve the largest region. F denotes the number of executed test cases.

IV. Simulation Study

The mean F-measure has to be determined to measure the performance of the presented algorithm. However, this seems not to be straightforward—at least theoretically. Therefore, the Monte Carlo method is applied to determine this characteristic through simulation.

A. Preliminaries

Let $X_1, \ldots, X_n$ be idependent and identically distributed random variables. $\overline{X_n} := \frac{1}{n}\sum_1^n X_i$ is the sample mean of the X_i. According to the central limit theorem [19],

$$\frac{\overline{X_n} - \mu}{\sigma/\sqrt{n}}$$

is standard Gaussian distributed as n approaches infinity. It is also a good approximation for $n \geq 30$ (a common rule of thumb). μ denotes the true mean and σ^2 the true variance of the X_i. Since σ is unknown, it is replaced by the square root of the sample variance

$$S_n^2 := \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X_n})^2$$

as an approximation.

It follows that

$$|\overline{X_n} - \mu| \leq \frac{\sigma}{\sqrt{n}} \cdot \Phi^{-1}\left(\frac{2-\alpha}{2}\right)$$

on confidence level $1 - \alpha$ and for $n \geq 30$, where $\Phi^{-1}(\cdot)$ denotes the inverse standard Gaussian distribution function. Furthermore, σ can be approximated by S_n. Therefore, it is possible to determine the accuracy of an estimation $\overline{X_n}$ on confidence level $1 - \alpha$.

B. The Simulation Design

For the simulations, the sample size n was chosen 50,000, i. e. the algorithm was run with 50,000 randomly chosen failure patterns. The analysis of ART algorithms via simulation is common practice—however, such a big sample size has not always been used. The confidence level $1 - \alpha$ was chosen

0.99. In a table for the Gaussian distribution one can look up $\Phi^{-1}(0.995) \approx 2.58$. Therefore,

$$|\overline{X_n} - \mu| \leq \frac{S_n}{\sqrt{50000}} \cdot 2.58 \approx 0.01154 \cdot S_n.$$

The failure pattern was randomly generated. The area θA of the failure pattern was determined by the failure rate θ and the area A of the input domain. For the block pattern, a square was chosen randomly, totally within the input domain. For the strip pattern, two adjacent sides and two points on these sides were chosen randomly. The strip was then constructed centered on the line connecting these points and its width was computed so that the strip had the desired area θA. Points near the corners were rejected to avoid overly wide strips. For the point pattern, 50 non-overlapping discs with equal radius lying totally within the input domain where randomly generated to achieve the total area θA.

The first part of the simulations were to investigate the performance of the presented ART algorithm and to find suitable values for the factor f. These simulations were done for the following

- failure rates: $0.01, 0.005, 0.002, 0.001, 0.0005$
- failure patterns: block, strip, and point
- factors f: $0.0, 0.05, 0.1, 0.15, 0.2, \ldots, 0.45$.

The second part of the simulations was performed in order to compare the improved ART algorithm with related ART algorithms. The parameters of the various ART methods were chosen as suggested in the respective publications: RRT ($R = 1.5$), D-ART ($k = 10$), ART by Random Partitioning with Localization and RRT ($R = 0.4$) resp. D-ART ($k = 3$), ART by Bisection with Localization and RRT ($R = 0.7$) resp. D-ART ($k = 13$), ART by Bisection with Restriction ($f = 0.3$), and ART by Random Partitioning with Restriction ($f = 0.4$). (The last parameter has been determined by the first part of the simulations.) In this case, the above failure rates and patterns were also used complemented by the point pattern with 10 discs. All methods were implemented in Java for the study and the results were obtained for these implementations. Thereby, we wanted to ensure comparable results. These results are in perfect accordance with related publications.

C. Results and Discussion

Table I shows the results of the first part of the simulations (the tables for the strip and point pattern are omitted). This table contains the relative[3] empirical mean F-measure for the block failure pattern and all factors f and all failure rates θ. The accuracy of the mean is also given on confidence level 99% below the mean. The minimum of each column is in bold face.

The optimal factor f is between 0.35 and 0.45 for the block failure pattern, between 0.35 and 0.45 for the strip failure pattern, and between 0.25 and 0.45 for the point failure pattern.

To determine the best choice for the factor f, a relative squared distance criterion has been used as follows: For each failure pattern p, failure rate θ, and factor f, the relative empirical mean F-measure is denoted $F_{p,\theta,f}$. For fixed p and θ, let $F_{p,\theta}^{min} := \min_f\{F_{p,\theta,f}\}$. Then, the relative squared difference is

$$d_{f,p} := \sum_{\theta} \left(\frac{F_{p,\theta,f} - F_{p,\theta}^{min}}{F_{p,\theta}^{min}} \right)^2$$

for one single failure pattern, and

$$d_f := \sum_p d_{f,p}$$

for all failure patterns. The values of $F_{p,\theta,f}$ are given in Table I (and omitted for the strip and point pattern). The values of $d_{f,p}$ are given in the columns two to four of Table II; the fiveth column contains the values of d_f. The factor $f = 0.4$ has minimal squared difference d_f. Therefore, this choice for f is optimal for Restricted ART by Random Partitioning.

TABLE II
DETERMINATION OF THE OPTIMAL FACTOR f FOR RESTRICTED ART BY RANDOM PARTITIONING THROUGH THE RELATIVE SQUARED DIFFERENCE

	block	strip	point	**sum**
$f = 0$	0.36713	0.00739	0.00412	0.37864
$f = 0.05$	0.19791	0.00531	0.00123	0.20446
$f = 0.1$	0.10238	0.00295	0.00074	0.10607
$f = 0.15$	0.05353	0.00184	0.00028	0.05565
$f = 0.2$	0.02549	0.00137	0.00043	0.02729
$f = 0.25$	0.00929	0.00078	0.00019	0.01026
$f = 0.3$	0.00413	0.00055	0.00040	0.00508
$f = 0.35$	0.00150	**0.00013**	0.00020	0.00184
$f = 0.4$	**0.00054**	0.00023	0.00069	**0.00146**
$f = 0.45$	0.00973	0.00077	**0.00006**	0.01056

For the optimal factor $f = 0.4$ the F-measure is between 0.604 and 0.642 for the block failure pattern, between 0.909 and 0.977 for the strip failure pattern, and between 0.955 and 0.984 for the point pattern (with 50 discs).

Tables III–VI show the results of the second part of the simulation study—the comparison of ART by Random Partitioning with Restriction with other common ART methods. These results were obtained by the study with our Java implementations of all methods. The abbreviations are defined as follows: Restricted Random Testing (RRT), Distance-Based ART (D-ART), ART by Random Partitioning (ART-RP), ART by Bisection (ART-Bi.), ART by Random Partitioning with Localization and RRT (ART-RP Loc. RRT) resp. D-ART (ART-RP Loc. D-ART), ART by Bisection with Localization and RRT (ART-Bi. Loc. RRT) resp. D-ART (ART-Bi. Loc. D-ART), Restricted ART by Random Partitioning (ART-RP Res.), and ART by Bisection with Restriction (ART-Bi. Res.).

For the block failure pattern the relative mean F-measure is between 0.603 and 0.642 for the new ART method. This is better than RRT and D-ART for high failure rates. For smaller failure rates, the proposed method is also not much worse than D-ART, and only a little bit worse than RRT. For the ART methods with linear runtime or "nearly linear" runtime, the

[3] to the theoretical mean F-measure of pure Random Testing (with replacement)

TABLE I

THE RELATIVE MEAN F-MEASURE OF RESTRICTED ART BY RANDOM PARTITIONING FOR THE BLOCK FAILURE PATTERN

	$\theta = 0.01$	$\theta = 0.0050$	$\theta = 0.0020$	$\theta = 0.0010$	$\theta = 0.0005$
$f = 0$	0.769 (± 0.008)	0.782 (± 0.008)	0.782 (± 0.008)	0.793 (± 0.008)	0.799 (± 0.008)
$f = 0.05$	0.717 (± 0.007)	0.735 (± 0.007)	0.742 (± 0.007)	0.752 (± 0.008)	0.757 (± 0.008)
$f = 0.1$	0.683 (± 0.007)	0.696 (± 0.007)	0.709 (± 0.007)	0.718 (± 0.007)	0.725 (± 0.007)
$f = 0.15$	0.657 (± 0.006)	0.670 (± 0.007)	0.689 (± 0.007)	0.696 (± 0.007)	0.695 (± 0.007)
$f = 0.2$	0.632 (± 0.006)	0.653 (± 0.006)	0.669 (± 0.006)	0.671 (± 0.006)	0.684 (± 0.007)
$f = 0.25$	0.616 (± 0.006)	0.634 (± 0.006)	0.647 (± 0.006)	0.658 (± 0.006)	0.662 (± 0.006)
$f = 0.3$	0.606 (± 0.006)	0.624 (± 0.006)	0.637 (± 0.006)	0.647 (± 0.006)	0.657 (± 0.006)
$f = 0.35$	**0.598** (± 0.006)	0.613 (± 0.006)	0.632 (± 0.006)	0.640 (± 0.006)	0.643 (± 0.006)
$f = 0.4$	0.605 (± 0.006)	**0.604** (± 0.006)	**0.630** (± 0.006)	0.630 (± 0.006)	**0.642** (± 0.006)
$f = 0.45$	0.630 (± 0.006)	0.606 (± 0.005)	0.668 (± 0.006)	**0.618** (± 0.005)	0.677 (± 0.006)

TABLE III

THE RELATIVE MEAN F-MEASURE OF THE RESPECTIVE ART METHOD FOR THE BLOCK FAILURE PATTERN

	$\theta = 0.0100$	$\theta = 0.0050$	$\theta = 0.0020$	$\theta = 0.0010$	$\theta = 0.0005$
RRT	0.648 (± 0.005)	0.633 (± 0.005)	**0.612** (± 0.005)	**0.603** (± 0.005)	**0.595** (± 0.005)
D-ART	0.673 (± 0.006)	0.659 (± 0.006)	0.650 (± 0.006)	0.639 (± 0.006)	0.635 (± 0.006)
ART-RP	0.768 (± 0.008)	0.777 (± 0.008)	0.791 (± 0.008)	0.795 (± 0.008)	0.794 (± 0.008)
ART-Bi.	0.735 (± 0.007)	0.738 (± 0.007)	0.734 (± 0.007)	0.740 (± 0.007)	0.734 (± 0.007)
ART-RP Loc. RRT	0.681 (± 0.006)	0.686 (± 0.006)	0.690 (± 0.007)	0.697 (± 0.007)	0.698 (± 0.007)
ART-RP Loc. D-ART	0.707 (± 0.007)	0.713 (± 0.007)	0.721 (± 0.007)	0.725 (± 0.007)	0.731 (± 0.007)
ART-Bi. Loc. RRT	0.680 (± 0.006)	0.673 (± 0.006)	0.672 (± 0.006)	0.668 (± 0.006)	0.664 (± 0.006)
ART-Bi. Loc. D-ART	0.663 (± 0.005)	0.651 (± 0.006)	0.636 (± 0.005)	0.634 (± 0.005)	0.626 (± 0.005)
ART-RP Res.	**0.603** (± 0.006)	**0.609** (± 0.006)	0.631 (± 0.006)	0.633 (± 0.006)	0.642 (± 0.006)
ART-Bi. Res.	0.658 (± 0.007)	0.663 (± 0.006)	0.674 (± 0.006)	0.679 (± 0.007)	0.686 (± 0.006)

presented algorithm is nearly always the most effective one (regarding the mean F-measure).

The relative mean F-measure is between 0.906 and 0.969 for the presented ART algorithm and the strip failure pattern. These values are also only a little bit worse than those of the RRT and D-ART methods for lower failure rates. For the fast, i. e. (nearly) linear time ART algorithms, it is also among the best and for small failure rates it tends to be the best choice among them.

The results for the point pattern with 10 discs and with 50 discs are surprising: The proposed ART algorithm achieves the best relative mean F-measure for the pattern with 10 discs and is (considering the accuracy of the results) extremely close to the best method (ART by Bisection and Restriction) for the pattern with 50 discs.

The proposed ART method, thus, performs best (among all ART methods) for the point pattern, and best (among all "fast" ART methods) for the block and the strip pattern. For high failure rates it is even the best method also for the block failure pattern. Due to its computational efficiency, the very good mean F-measure, and its simplicity, Restricted ART by Random Partitioning is the best choice among the "fast" ART methods and altogether a very good choice.

As with all Adaptive Random Testing methods, the proposed algorithm only solves the generation of the test cases for bounded rectangular input domains of arbitrary dimension. More complex inputs need, however, special treatment.

The proposed algorithm provides a method to generate test inputs—as all other ART algorithms. A test oracle, i. e. a program that evaluates the outputs and decides "pass" or "fail",

TABLE IV

THE RELATIVE MEAN F-MEASURE OF THE RESPECTIVE ART METHOD FOR THE STRIP FAILURE PATTERN

	$\theta = 0.0100$	$\theta = 0.0050$	$\theta = 0.0020$	$\theta = 0.0010$	$\theta = 0.0005$
RRT	**0.866** (± 0.010)	0.910 (± 0.010)	**0.932** (± 0.011)	**0.943** (± 0.011)	0.962 (± 0.011)
D-ART	0.869 (± 0.010)	**0.903** (± 0.010)	0.934 (± 0.011)	0.958 (± 0.011)	**0.958** (± 0.011)
ART-RP	0.950 (± 0.010)	0.967 (± 0.011)	0.968 (± 0.011)	0.983 (± 0.011)	0.992 (± 0.011)
ART-Bi.	0.916 (± 0.010)	0.943 (± 0.011)	0.957 (± 0.011)	0.969 (± 0.011)	0.990 (± 0.011)
ART-RP Loc. RRT	0.927 (± 0.010)	0.937 (± 0.010)	0.958 (± 0.011)	0.969 (± 0.011)	0.978 (± 0.011)
ART-RP Loc. D-ART	0.922 (± 0.010)	0.948 (± 0.010)	0.967 (± 0.011)	0.973 (± 0.011)	0.979 (± 0.011)
ART-Bi. Loc. RRT	0.902 (± 0.010)	0.923 (± 0.010)	0.953 (± 0.011)	0.966 (± 0.011)	0.972 (± 0.011)
ART-Bi. Loc. D-ART	0.894 (± 0.010)	0.918 (± 0.010)	0.952 (± 0.011)	0.963 (± 0.011)	0.975 (± 0.011)
ART-RP Res.	0.906 (± 0.010)	0.930 (± 0.010)	0.944 (± 0.011)	0.966 (± 0.011)	0.969 (± 0.011)
ART-Bi. Res.	0.885 (± 0.009)	0.907 (± 0.010)	0.937 (± 0.010)	0.956 (± 0.011)	0.965 (± 0.011)

TABLE V

THE RELATIVE MEAN F-MEASURE OF THE RESPECTIVE ART METHOD FOR THE POINT FAILURE PATTERN WITH 10 DISCS

	$\theta = 0.0100$	$\theta = 0.0050$	$\theta = 0.0020$	$\theta = 0.0010$	$\theta = 0.0005$
RRT	0.975 (± 0.010)	0.958 (± 0.010)	0.942 (± 0.010)	0.936 (± 0.010)	0.924 (± 0.010)
D-ART	0.960 (± 0.010)	0.946 (± 0.010)	0.929 (± 0.010)	0.919 (± 0.010)	0.926 (± 0.010)
ART-RP	0.945 (± 0.011)	0.948 (± 0.011)	0.949 (± 0.011)	0.953 (± 0.011)	0.952 (± 0.011)
ART-Bi.	0.934 (± 0.010)	0.933 (± 0.010)	0.931 (± 0.010)	0.930 (± 0.010)	0.932 (± 0.010)
ART-RP Loc. RRT	0.926 (± 0.010)	0.927 (± 0.010)	0.926 (± 0.010)	0.921 (± 0.010)	0.929 (± 0.010)
ART-RP Loc. D-ART	0.930 (± 0.010)	0.929 (± 0.010)	0.932 (± 0.010)	0.936 (± 0.010)	0.936 (± 0.010)
ART-Bi. Loc. RRT	0.930 (± 0.010)	0.924 (± 0.010)	0.920 (± 0.010)	0.915 (± 0.010)	0.918 (± 0.010)
ART-Bi. Loc. D-ART	0.961 (± 0.010)	0.939 (± 0.010)	0.939 (± 0.010)	0.930 (± 0.010)	0.927 (± 0.010)
ART-RP Res.	**0.852** (± 0.009)	**0.869** (± 0.009)	**0.892** (± 0.010)	**0.892** (± 0.009)	**0.896** (± 0.009)
ART-Bi. Res.	0.871 (± 0.009)	0.881 (± 0.010)	0.899 (± 0.010)	0.900 (± 0.010)	0.903 (± 0.010)

is also required for efficient test execution. This is one of the key problems in software testing and not covered by the present paper.

V. CONCLUSION

An algorithm based on ART by Random Partitioning, which iteratively subdivides the largest region of the input domain by a newly generated test case, has been given. This method applies the principle of restriction to ART by Random Partitioning in order to avoid nearby test cases and, thus, to improve the wide-spreadness of test cases. Instead of selecting test cases from sub-regions of the input domain as in ART by Random Partitioning, they are chosen from a restricted version of this region. Since theoretical analysis of the algorithm's effectivity would be very complicated or even not feasible, a simulation study has been performed. The results are very encouraging. The mean F-measure, i. e. the average (random) number of test cases necessary to detect the first failure, of the proposed method is best among all related ART methods for the point failure pattern with 10 discs and for the block failure pattern with high failure rates. For the block pattern, the mean F-measure results are quite close to that of D-ART and RRT, which are considered as reference methods. The big advantage of the proposed algorithm is its runtime of $\mathcal{O}(F \log F)$, which is "nearly linear", where F denotes the number of executed test cases. D-ART and RRT have quadratic runtime, which is extremely bad for practically low failure rates (with many test cases necessary to detect the first failure, i. e. high values of F). Therefore, the proposed algorithm is simple, fast (with "nearly linear" runtime), and very effective

TABLE VI

THE RELATIVE MEAN F-MEASURE OF THE RESPECTIVE ART METHOD FOR THE POINT FAILURE PATTERN WITH 50 DISCS

	$\theta = 0.0100$	$\theta = 0.0050$	$\theta = 0.0020$	$\theta = 0.0010$	$\theta = 0.0005$
RRT	1.022 (± 0.011)	1.003 (± 0.011)	1.000 (± 0.011)	0.995 (± 0.011)	0.986 (± 0.011)
D-ART	1.002 (± 0.011)	1.006 (± 0.011)	0.989 (± 0.011)	0.986 (± 0.011)	0.985 (± 0.011)
ART-RP	0.979 (± 0.011)	0.980 (± 0.011)	0.982 (± 0.011)	0.986 (± 0.011)	0.992 (± 0.011)
ART-Bi.	0.986 (± 0.011)	0.986 (± 0.011)	0.977 (± 0.011)	0.977 (± 0.011)	0.975 (± 0.011)
ART-RP Loc. RRT	0.967 (± 0.011)	0.972 (± 0.011)	0.981 (± 0.011)	0.987 (± 0.011)	0.987 (± 0.011)
ART-RP Loc. D-ART	0.984 (± 0.011)	0.983 (± 0.011)	0.985 (± 0.011)	0.984 (± 0.011)	0.989 (± 0.011)
ART-Bi. Loc. RRT	0.984 (± 0.011)	0.985 (± 0.011)	0.991 (± 0.011)	0.985 (± 0.011)	0.985 (± 0.011)
ART-Bi. Loc. D-ART	1.012 (± 0.011)	0.998 (± 0.011)	0.986 (± 0.011)	0.988 (± 0.011)	0.990 (± 0.011)
ART-RP Res.	0.949 (± 0.011)	0.960 (± 0.011)	0.969 (± 0.011)	0.970 (± 0.011)	0.970 (± 0.011)
ART-Bi. Res.	**0.948** (± 0.011)	**0.948** (± 0.011)	**0.968** (± 0.011)	**0.964** (± 0.011)	**0.970** (± 0.011)

(regarding the F-measure). If automated Random Testing is to be used, the presented method is a very good choice. On the one hand, it has "nearly linear" runtime and needs, for this reason, only a short time to generate many test cases. On the other hand, it requires much fewer test cases than Random Testing to detect the first failure (e. g. between 60% and 64% for the block failure pattern) and saves, thus, precious time, since the execution of the system under test usually is very time-consuming for large systems (despite of test automation). The proposed method allows, therefore, for the construction of much more efficient Random Testing tools.

The simple rectangular input domain chosen in the present paper only serves as an illustration example which makes explanation simple. However, the random testing method explained in the present paper can be applied to arbitrary input domains. In this case, simply a strategy is necessary to subdivide partitions based on test cases and to restrict partitions. If the domain is transformed for this purpose, one has to be careful to retain the property that failure-causing inputs tend to cluster (also in the transformed domain).

REFERENCES

[1] G. J. Myers, *The Art of Software Testing*. New York: Wiley, 1979.

[2] V. D. Agrawal, "When to use random testing," *IEEE Transactions on Computers*, vol. 27, pp. 1054–1055, 1978.

[3] J. W. Duran and S. C. Ntafos, "An evaluation of random testing," *IEEE Transactions on Software Engineering*, vol. 10, pp. 438–444, 1984.

[4] R. Hamlet, "Random testing," in *Encylopedia of Software Engineering*. Wiley, 1994, pp. 970–978.

[5] P. S. Loo and W. K. Tsai, "Random testing revisited," *Information and Software Technology*, vol. 30, pp. 402–417, 1988.

[6] P. B. Schneck, "Comment on "when to use random testing"," *IEEE Transactions on Computers*, vol. 28, pp. 580–581, 1979.

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

[8] D. Slutz, "Massive stochastic testing of SQL," in *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB 98)*, 1998, pp. 618–622.

[9] 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)*. IEEE Computer Society, 2003, pp. 20–24.

[10] 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*, vol. 38, pp. 775–782, 1996.

[11] T. Y. Chen, T. H. Tse, and Y. T. Yu, "Proportional sampling strategy: A compendium and some insights," *The Journal of Systems and Software*, vol. 58, pp. 65–81, 2001.

[12] T. Y. Chen, H. Leung, and I. K. Mak, "Adaptive random testing," in *Proceedings of the 9th Asian Computing Science Conference (ASIAN 2004)*, ser. Lecture Notes in Computer Science, M. J. Maher, Ed., vol. 3321. Springer, 2004, pp. 320–329.

[13] K. P. Chan, T. Y. Chen, and D. Towey, "Restricted random testing," in *Proceedings of the 7th European Conference on Software Quality (ECSQ 2002)*, ser. Lecture Notes in Computer Science, J. Kontio and R. Conradi, Eds., vol. 2349. Springer, 2002, pp. 321–330.

[14] ——, "Normalized restricted random testing," in *Proceedings of the 18th Ada-Europe International Conference on Reliable Software Technologies*, ser. Lecture Notes in Computer Science, vol. 2655. Springer, 2003, pp. 368–381.

[15] K. P. Chan, T. Y. Chen, F.-C. Kuo, and D. Towey, "A revisit of adaptive random testing by restriction," in *Proceedings of the 28th International Computer Software and Applications Conference (COMPSAC 2004)*. IEEE Computer Society, September 2004, pp. 78–85.

[16] T. Y. Chen, G. Eddy, R. Merkel, and P. K. Wong, "Adaptive random testing through dynamic partitioning," in *Proceedings of the 4th International Conference on Quality Software (QSIC 2004)*. IEEE Computer Society, September 2004, pp. 79–86.

[17] T. Y. Chen and D. Huang, "Adaptive random testing by localization," in *Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC 2004)*. IEEE Computer Society, November/December 2004, pp. 292–298.

[18] J. Mayer, "Adaptive random testing by bisection and localization," in *Proceedings of the Fifth International Workshop on Formal Approaches to Testing of Software (FATES 2005)*, ser. Lecture Notes in Computer Science, vol. 3997. Springer-Verlag, Berlin, 2005, (in press).

[19] G. Casella and R. L. Berger, *Statistical Inference*. CA, USA: Wadsworth Group, Duxbury, 2002.