Information and Software Technology 67 (2015) 13–29

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Enhancing mirror adaptive random testing through dynamic partitioning

Rubing Huang [a], Huai Liu [b,*], Xiaodong Xie [c], Jinfu Chen [a]

[a] School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, Jiangsu, China
[b] Australia-India Research Centre for Automation Software Engineering, RMIT University, Melbourne 3001, VIC, Australia
[c] School of Computer Science and Technology, Huaqiao University, Xiamen 361021, Fujian, China

A R T I C L E I N F O

Article history:
Received 16 December 2014
Received in revised form 4 June 2015
Accepted 12 June 2015
Available online 19 June 2015

Keywords:
Software testing
Random testing
Adaptive random testing
Mirror adaptive random testing

A B S T R A C T

Context: Adaptive random testing (ART), originally proposed as an enhancement of random testing, is often criticized for the high computation overhead of many ART algorithms. Mirror ART (MART) is a novel approach that can be generally applied to improve the efficiency of various ART algorithms based on the combination of "divide-and-conquer" and "heuristic" strategies.
Objective: The computation overhead of the existing MART methods is actually on the same order of magnitude as that of the original ART algorithms. In this paper, we aim to further decrease the order of computation overhead for MART.
Method: We conjecture that the mirroring scheme in MART should be dynamic instead of static to deliver a higher efficiency. We thus propose a new approach, namely dynamic mirror ART (DMART), which incrementally partitions the input domain and adopts new mirror functions.
Results: Our simulations demonstrate that the new DMART approach delivers comparable failure-detection effectiveness as the original MART and ART algorithms while having much lower computation overhead. The experimental studies further show that the new approach also delivers a better and more reliable performance on programs with failure-unrelated parameters.
Conclusion: In general, DMART is much more cost-effective than MART. Since its mirroring scheme is independent of concrete ART algorithms, DMART can be generally applied to improve the cost-effectiveness of various ART algorithms.

1. Introduction

Software testing has been widely acknowledged as a mainstream technique for assessing and improving software quality. One basic approach to testing is to randomly generate test cases from the set of all possible program inputs (namely the *input domain*). Though very simple, *random testing* (RT) is still considered as one of the state-of-the-art testing techniques, along with other more complicated and systematic testing methods [1,2]. RT may be the unique testing method that can be used for both operational testing (where the software reliability is estimated) and debug testing (where software failures are actively detected with the purpose of removing relevant bugs) [3]. Despite the controversies in the effectiveness of RT as a debug testing method [4], it has been popularly used to test various systems, such as UNIX utility programs [5], Windows NT applications [6], Java Just-In-Time compilers [7], embedded software systems [8], SQL database systems [9].

Besides the applications of RT into different domains, much research has been conducted on how to improve its effectiveness in detecting failures. *Adaptive random testing* (ART) [10] is one major approach to enhancing RT. The basic idea of ART was motivated by the common observation made by researchers from different areas: The *failure-causing inputs* (i.e., program inputs that can reveal failures) tend to be clustered into contiguous *failure regions* [11–14]. Given that the failure regions are contiguous, the non-failure regions should also be contiguous. In other words, adjacent program inputs show a certain degree of similarity in failure-revealing behaviors. According to this intuition, Chen et al. [10] conjectured that test cases should be evenly spread across the whole input domain for achieving high failure-detection effectiveness, and proposed ART to implement the notion of "even spread". Since the inception of ART, many ART algorithms have been proposed, such as *fixed-sized-candidate-set ART* (FSCS-ART) [10], *lattice-based ART* (LART) [15], and *restricted*

* Corresponding author.
E-mail addresses: rbhuang@ujs.edu.cn (R. Huang), huai.liu@rmit.edu.au (H. Liu), xiaodongxie@hqu.edu.cn (X. Xie), jinfuchen@ujs.edu.cn (J. Chen).

http://dx.doi.org/10.1016/j.infsof.2015.06.003
0950-5849/

random testing (RRT) [16]. ART has also been applied to test various programs [17–19].

Previous studies [10,15–19] have shown that ART can use fewer test cases than RT to detect the first software failure. However, the high computation overhead of many ART algorithms brings severe criticism and limits ART's adoption in practice [20]. In order to improve the testing efficiency of ART, many overhead reduction strategies have been proposed [21–26]. Among these strategies, a well-studied testing method is *mirror adaptive random testing* (MART) [21], which is a novel approach based on the combination of "divide-and-conquer" and "heuristic" strategies. MART first divides the whole input domain into equal-sized disjoint subdomains. Then, one subdomain is chosen as the *source domain* while others as the *mirror domains*. MART generates test cases in the source domain according to one existing ART algorithm, and then maps each test case from the source domain into the so-called mirror test cases in the mirror domains. As shown in previous studies [21,27], MART can reduce computation overhead of the original ART algorithms while maintaining similar failure-detection effectiveness.

However, the computation overhead of the existing MART methods actually has the same order of magnitude as that of the original ART algorithms. For example, one ART algorithm, FSCS-ART, has the computation overhead of $O(n^2)$ for generating n test cases. According to previous investigations [21], the MART based on the original FSCS-ART algorithm requires about $O(n^2/m^2)$ time to generate n test cases, where m is the number of subdomains. In other words, the computation overhead of MART based on FSCS-ART is also in the quadratic order.

In this paper, we propose an enhanced MART method, namely *dynamic mirror adaptive random testing* (DMART), which divides the input domain incrementally along the testing process. The simulation results indicate that compared with original MART and ART algorithms, DMART requires much less computation overhead while delivering comparable failure-detection effectiveness. Our empirical studies further show that the new method also has a better and more reliable performance than original MART algorithms on real-life programs especially when there exist some input parameters that are not related to failures.

The paper is organized as follows. Section 2 introduces some background information of ART and MART. Section 3 discusses drawbacks of MART, and then proposes our new DMART method. Section 4 reports our experimental studies, which examine the computational overhead and failure-detection effectiveness of the new approach. The experimental results are given in Section 5. Section 6 discusses the threats to validity of our study. Section 7 presents some related work. Section 8 summarizes the paper.

2. Background

2.1. Adaptive random testing

Similar to RT, *adaptive random testing* (ART) [10] also randomly generates program inputs from the input domain. However, ART makes use of additional criteria to choose inputs as test cases in order to evenly spread test cases over the input domain. There are many criteria to guide the selection of test cases, one criterion of which is by distance. *Fixed-sized-candidate-set ART* (FSCS-ART) [10] is one typical algorithm of ART by distance. FSCS-ART uses two test case sets, the *executed set* denoted by E and the *candidate set* denoted by $C = \{c_1, c_2, \ldots, c_k\}$. E contains all test cases which were already executed without revealing any failure; while C contains k randomly generated inputs, where k is assigned by testers before testing and keeps unchanged throughout the testing process. An input in C will be chosen as the next test case in E if it has the longest distance to its nearest neighbor in E.

Previous simulations and empirical studies [10,15,16] have demonstrated that the failure-detection effectiveness of ART is better than that of RT in terms of detecting the first software failure using fewer test cases. However, there exists a criticism [20] of ART due to the high computation overhead of many ART algorithms. For example, FSCS-ART requires $O(n^2)$ time to generate n test cases.

2.2. Mirror adaptive random testing

As discussed before, many ART algorithms may face the criticism of high computation overhead. To improve the efficiency of ART, Chen et al. [21] proposed a novel overhead reduction strategy, namely *mirror adaptive random testing* (MART), which could be generally applied to many existing ART algorithms.

Before testing, MART first divides the input domain into some disjoint and equal-size subdomains, and then assigns one subdomain as the *source domain* while others as the *mirror domains*. After that, MART applies original ART algorithm in the source domain to generate a test case tc (namely *source test case*). Then, MART uses a function to map tc from the source domain into all mirror domains, to construct other test cases (namely *mirror test cases*).

According to previous studies [21,27], there are three major components of the *mirroring scheme* in MART, namely *mirror partitioning*, *mirror function*, and *mirror selection order*.

2.2.1. Mirror partitioning

Suppose that the dimension of the input domain is $d \geqslant 1$. In MART, each coordinate of the input domain is divided into $u_i \geqslant 1$ $(i = 1, 2, \ldots, d)$ parts of the equal length. Totally, the input domain is partitioned into $u_1 \times u_2 \times \cdots \times u_d$ subdomains. Fig. 1 shows some simple ways of mirror partitioning that could be used for MART with the 2-dimensional input domain. Since the use of a large number of mirror domains "may introduce duplicated test case patterns" (i.e., the distribution of test cases is duplicated in each subdomain) that "may destroy the overall randomness of test case selection", a small number of mirror domains would be more appropriate for MART [21]. In our experimental studies, therefore, we follow the practice adopted in previous studies [21] of choosing mirror partitioning with a small number of mirror domains for MART.

2.2.2. Mirror function

There exist two commonly used mirror functions in MART, namely *Translate* and *Reflect*. Fig. 2 illustrates these two mirror functions in the 2-dimensional input domain. Suppose that $(0, 0)$ and (v_1, v_2) are the minimum and maximum coordinate values of the input domain, respectively. The mirror partitioning is 2×1, where the shaded region D_1 is the source domain, and D_2 is the mirror domain. The *Translate* function will map a test case (x, y) in D_1 into $\left(x + \frac{v_1}{2}, y\right)$ in D_2, while the *Reflect* function will map (x, y) in D_1 into $(v_1 - x, y)$ in D_2. Previous simulation results have indicated that there is no significant performance difference between the Translate and Reflect mirror functions [21]. In this study, we use the Translate mirror function for MART.

2.2.3. Mirror selection order

For the mirror selection order, there exist three ways to guide the selection order of mirror domains [27]: (1) sequential order, i.e., mirror domains are chosen according to sequential-ordered sections in each dimension; (2) random order, i.e., mirror domains are selected randomly for generating the next test case; and (3) adaptive-random order, i.e., mirror domains are chosen in a