Optimal Controlled Random Tests

Ireneusz Mrozek(✉) and Vyacheslav Yarmolik

Faculty of Computer Science, Bialystok University of Technology,
Wiejska 45A, 15-351 Bialystok, Poland
i.mrozek@pb.edu.pl, yarmolik10ru@yahoo.com
http://www.wi.pb.edu.pl

Abstract. Controlled random tests, methods of their generation, main criteria used for their synthesis, such as the Hamming distance and the Euclidean distance, as well as their application to the testing of both hardware and software systems are discussed. Available evidences suggest that high computational complexity is one of the main drawbacks of these methods. Therefore we propose a technique to overcome this problem. In the paper we propose the algorithm for optimal controlled random tests generation. Both experimental and analytical investigation clearly show the high efficiency of proposed solution especially for the multi-run tests with small number of iterations. The given tests can be applied for hardware and software testing but it seems they may be particularly interesting from the perspective of the effective detection of pattern sensitive faults in RAMs.

Keywords: Random tests · Controlled tests · Multi-run tests · Hemming distance · Euclidean distance · Pattern sensitive faults · RAM

1 Introduction

It is known that the testing problem is computationally most expensive and mathematically NP-complete [1,2]. The same time the complexity of the modern embedded systems is steadily growing. It is, therefore, important to consider how testing can be performed more effectively.

In case of single hard to detect faults that may have just a handful of unique tests in the entire search space there have been proposed probability based algorithms where new test vectors are generated based on the input probability correlation of previously unsuccessful test vectors [3].

In case of systems with limited number of inputs we can use exhaustive testing. However the exponential growth of the test length restricts the concept of the exhaustive testing to circuits with a limited number of inputs [4].

This paper was supported by grant S/WI/3/13 from Faculty of Computer Science at Bialystok University of Technology, Ministry of Science and Higer Education, Poland.

K. Saeed et al. (Eds.): CISIM 2017, LNCS 10244, pp. 27–38, 2017.
DOI: 10.1007/978-3-319-59105-6_3

Locally exhaustive [5] or pseudo exhaustive [6–8] testing is a concept to avoid the restricted number of inputs of the circuit under test. These approaches are the real alternatives to exhaustive testing. They allow for sufficiently reducing the number of the test vectors. It is possible by taking advantage of the fact that often many or all output variables depend only on a small subset of input variables [5].

As a good approximation of exhaustive and pseudo exhaustive testing the random testing have been widely used [9–12]. The advantages of random testing include its low cost, ease of implementation, ability to generate numerous test cases automatically, generation of test cases in the absence of the object specification and apart from these; it brings randomness into the testing process. Random testing and its variations have been extensively used and studied for both hardware and software systems. Unfortunately random testing is much less effective than other testing techniques such as equivalence partition testing, boundary value analysis testing [13].

Standard random testing does not exploit some information that is available in black box testing environment. Therefore controlled approach to random testing (Controlled Random Tests) may be used. One of the approaches for controlled random testing is Antirandom Testing [14]. Antirandom Testing is based on various empirical observations showing that many faults result in failures in contiguous areas of the input domain [15]. Therefore one way to improve the failure-detection effectiveness of random testing is somehow taking advantage of this fact. In this case each test vector is chosen such that its total distance from all previous vectors is maximum [14,16]. This approach has proved more efficient than random testing [14]. Unfortunately the basic antirandom method essentially requires enumeration of the input space and computation of distances for each potential input vector [14]. Therefore many modification of antirandom tests have been proposed [15–24]. But even for improved version of the method, very often computations become too expensive for real dimension N of the test vectors [25].

In this paper the efficiency of controlled random tests is investigated and validated. Unlike our parallel proposals presented in [26,27] where we focus on multiple controlled random tests consisting of r single controlled random tests, in this paper we investigate a single optimal controlled random test and we present the greedy-like algorithm for its generation. By optimal controlled random test we understand the set of test patterns where each pattern is chosen such that its total distance from all previous patterns is the greatest possible value. So, unlike the known solutions, our approach keeps the distances between the current and previous test patterns as large as possible. All the analytical results are validated through extensive simulation-based experiments.

This paper is divided into four more sections. Section 2 provides a brief overview of the principle concept behind Controlled Random Tests. Section 3 highlights the main idea of our algorithm. We analyze and discuss the experimental results in Sect. 4, and make a conclusion in Sect. 5.

2 Controlled Random Tests Investigations

The key feature of controlled random tests generation is the information, which can be obtained from the test and used for test patterns generation. There are numerous different useful approaches which exploited the information from the test itself [14,25,28]. For all this methods, considering the binary case, the next definitions can be done.

Definition 1. *Test (T) is a set of test patterns* $\{T_0, T_1, T_2, \ldots, T_{q-1}\}$*, where* $T_i = t_{i,N-1}, t_{i,N-2}, \ldots, t_{i,2}, t_{i,1}, t_{i,0}$*, with* $t_{i,l} \in \{0,1\}$*, and* N *is the size of patterns in bits.*

Definition 2. *Controlled Random Test* $T = \{T_0, T_1, T_2, \ldots, T_{q-1}\}$ *(CRT) is a test with randomly chosen the test patterns* T_i*,* $i \in \{0, 1, 2, \ldots, q-1\}$*, such that* T_i *satisfies some criterion.*

The first formally define approach for controlled random test generation have been presented in [14] by Y. Malaiya. The proposed approach was called antirandom testing, since selection of each test explicitly depends on the test patterns already generated. The next formal definition for antirandom tests have been used:

Definition 3. *Antirandom test (AT) is a test with a test pattern* T_i*,* $i \in \{0, 1, 2, \ldots, q-1\}$*, is chosen such that it satisfies some criterion with respect to all patterns* $T_0, T_1, T_2, \ldots, T_{i-1}$ *have been obtained before.*

To make each new pattern different compare with previously generated the Hamming and Cartesian distances as the measure of differences have been chosen, which can be defined as [14].

Definition 4. *The Hamming Distance HD*(T_i, T_j) *(HD) between two binary vectors* T_i *and* T_j *is calculated as a weight* $w(T_i \oplus T_j)$ *(number of ones) of vector* $T_i \oplus T_j$*.*

$$HD(T_i, T_j) = w(T_i \oplus T_j) = \sum_{l=0}^{N-1} (t_{i,l} \oplus t_{j,l}). \tag{1}$$

Definition 5. *The Cartesian Distance CD*(T_i, T_j) *(CD) between two binary vectors* T_i *and* T_j *is given by:*

$$\begin{aligned} \mathrm{CD}(T_i, T_j) &= \sqrt{(t_{i,0} - t_{j,0})^2 + (t_{i,1} - t_{j,1})^2 \ + \ldots + \ (t_{i,N-1} - t_{j,N-1})^2} \\ &= \sqrt{|t_{i,0} - t_{j,0}| + |t_{i,1} - t_{j,1}| \ + \ldots + \ |t_{i,N-1} - t_{j,N-1}|} \\ &= \sqrt{\mathrm{HD}(T_i, T_j)}. \end{aligned} \tag{2}$$

The antirandom testing scheme attempts to keep testing procedure as efficient as possible, taking into account the hypothesis that if two patterns have only a small distance between them then the sets of faults encountered by the two patterns is likely to have a number of faults in common. Conversely, if the distance between two patterns is large, then the set of faults detected by one is likely to contain only a few of the faults detected by the other [14,29]. For the set of more then two test patterns the next definitions have been proposed [14].

Definition 6. Total Hamming Distance *(THD),* Total Cartesian Distance *(TCD) for any pattern T_i is the sum of its Hamming (Cartesian) distances with respect to all previous patterns $T_0, T_1, T_2, \ldots, T_{i-1}$.*

$$\mathrm{THD(T_i)} = \sum_{j=0}^{i-1} \mathrm{HD}(T_i, T_j), \qquad \mathrm{TCD(T_i)} = \sum_{j=0}^{i-1} \mathrm{CD}(T_i, T_j). \tag{3}$$

Definition 7. Maximal Distance Antirandom Test *(MDAT) is a test such that each pattern T_i is chosen to make the total distance THD or TCD between T_i and patterns $T_0, T_1, T_2, \ldots, T_{i-1}$ maximal.*

The main properties of MHDAT and MCDAT are the following:

any permutation the patterns bits $t_{i,j}$ within the antirandom test T MHDAT (MCDAT) for all i patterns simultaneously will results in new MHDAT (MCDAT) antirandom test [14,29],

any MHDAT (MCDAT) will always contain complementary pair of patterns, i.e. T_{2k} will always be followed by T_{2k+1} which is complementary for all bits in T_{2k} where $k = 1, 2, \ldots,$ [14,29].

Above presented properties allow reducing the complexity of MHDAT (MCDAT) generation but, for general case, unfortunately the basic antirandom method essentially requires enumeration of the input space and computation of distances for each potential input pattern [14,29]. Even for improved version of the method computations, it becomes too expensive for real dimension N of the test patterns [16].

3 Optimal Controlled Random Test Generation

The key feature of controlled random test generation is the information obtained from the test and used for the current pattern T_i generation.

As shown in the previous section, the controlled random tests are generated based on a restricted number of metrics, including the Hamming distance between two patterns T_i and T_j ($\mathrm{HD}(T_i, T_j)$), the Cartesian distance ($\mathrm{CD}(T_i, T_j)$), the total Hamming distance for the next test pattern T_i ($\mathrm{THD}(T_i)$), the total Cartesian distance ($\mathrm{TCD}(T_i)$), the total Hamming distance for the test

T (THD(T)), and the total Cartesian distance (TCD(T)) [14,28,29]. These metrics are used for proposed greedy-like methods of controlled random test generation [14,16,25,28–30]. In all these algorithms, the best immediate, or local, solution has been taken to reach a globally optimal solution. The optimal choice at each stage based on simple metrics has been performed with the hope of finding the global optimum. The greedy-like algorithm was the only solution for controlled random test generation because it is faster than other optimization methods, like dynamic programming for example.

Based on the presented metrics, an optimal controlled random test (OCRT) will be constructed. Step by step, starting from one test pattern T_0, the consecutive patterns will be selected in terms of adopted earlier metrics according to the greedy-like algorithm.

At the beginning, the first test pattern T_0 is chosen as any arbitrary random N-bit binary vector out of 2^N possible. For example, let $T_0 = 000\ldots0$. This does not result in any loss of generality [14,16,28].

As the second pattern T_1, according to all metrics, an optimal value is the complement of T_0, then $T_1 = \overline{T}_0$, and $T = \{T_0, T_1\}$ for the previously adopted example $T_1 = 111\ldots1$. The optimality of this choice is supported by the maximal values of all metrics: $\mathrm{HD}(T_0, T_1) = \mathrm{THD}(T_1) = \mathrm{THD}(T) = N$, and $\mathrm{CD}(T_0, T_1) = \mathrm{TCD}(T_1) = \mathrm{TCD}(T) = \sqrt{N}$.

To obtain the third pattern T_2, the $\mathrm{HD}(T_i, T_j)$ and $\mathrm{CD}(T_i, T_j)$ measures cannot be used because the maximization of this value leads to the confusing result that $T_2 = T_0$. That is why, in our investigation, both characteristics will not be used. The same conclusion can be made for $\mathrm{THD}(T_i)$ and $\mathrm{THD}(T)$. It follows from the fact that for $T = \{T_0, T_1, T_2\}$, where $T_1 = \overline{T}_0$, any value of T_2 gives $\mathrm{THD}(T_2) = N$ and $\mathrm{THD}(T) = 2N$. It should be noticed that, even in cases $T_2 = T_0$ and $T_2 = T_1$, these metrics have the same values $\mathrm{THD}(T_2) = N$ and $\mathrm{THD}(T) = 2N$.

As the candidate for the third optimal pattern T_2, any pattern that satisfies to the next relations $T_2 \neq T_0$ and $T_2 \neq T_1$ can be chosen. For example, we take T_2 with $\mathrm{HD}(T_0, T_2) = Z$, then $\mathrm{HD}(T_1, T_2) = N - Z$. For Cartesian functions, the following results are $\mathrm{TCD}(T_i) = \mathrm{CD}(T_0, T_2) + \mathrm{CD}(T_1, T_2) = \sqrt{Z} + \sqrt{N - Z}$, $\mathrm{TCD}(T) = \mathrm{CD}(T_0, T_1) + \mathrm{CD}(T_0, T_2) + \mathrm{CD}(T_1, T_2) = \sqrt{N} + \sqrt{Z} + \sqrt{N - Z}$. Then, $\mathrm{maxTCD}(T_i)$ and $\mathrm{maxTCD}(T)$ can be achieved as the solution $Z = N/2$ of the equation $\delta(\sqrt{Z} + \sqrt{N - Z})/\delta Z = 0$. For further investigation, suppose that N is an even number, so that, in our example, the first $N/2$ bits of pattern $T_2 = 000\ldots01111$ take the value 0, and the rest take value 1.

To summarize the third pattern construction, it is quite important to emphasize that, for our test $T = \{T_0, T_1, T_2\} = \{000\ldots0, 111\ldots1, 000\ldots0111\ldots1\}$, the next patterns (fourth, fifth, and so on) should be selected from the set of patterns with the weight $w(T_i) = N/2$ due to the maximum values of $\mathrm{TCD}(T_2) = 2\sqrt{N/2}$ and $\mathrm{TCD}(T) = \sqrt{N} + 2\sqrt{N/2}$ for three pattern tests T. In this case, for new consecutive pattern T_i, the Hamming distances $\mathrm{HD}(T_i, T_0)$ and $\mathrm{HD}(T_i, T_1)$ have the values $N/2$, which allow for $\mathrm{TCD}(T_i)$ and $\mathrm{TCD}(T)$ maximization.

According to the previous discussion, as the potential fourth pattern, T_3 is the pattern with the weight $w(T_3) = N/2$ and should maximize the value of $\mathrm{TCD}(T_3)$ and $\mathrm{TCD}(T)$. The obvious solution exists, namely, $T_3 = \overline{T}_2$ and for our example $T_3 = 111\ldots1000\ldots0$.

The same result can be obtained based on the FAR algorithm [25]. According to this algorithm, using three previous patterns $T_0 = 000\ldots0$, $T_1 = 111\ldots1$ and $T_2 = 000\ldots0111\ldots1$, a centroid pattern will be obtained $C = c_{N-1}, c_{N-2}, \ldots, c_2, c_1, c_0 = 1/3, 1/3, 1/3, \ldots, 1/3, 2/3, 2/3, 2/3, \ldots, 2/3$. Thus, the FAR algorithm creates a binary centroid pattern $000\ldots01111$ [25]. At the final step of the FAR algorithm, a new anti-random pattern $T_3 = 111\ldots10000$ is constructed by complementing each bit of the binary centroid pattern.

For the pattern T_3, the total Cartesian distance is $\mathrm{TCD}(T_3) = \sqrt{N} + 2\sqrt{N/2}$ and $\mathrm{TCD}(T) = 2\sqrt{N} + 4\sqrt{N/2}$ for four pattern tests T.

This procedure can be generalized for the following steps. At all consecutive stages, the next pattern will be chosen to maximize two metrics $\mathrm{TCD}(T_i)$ and $\mathrm{TCD}(T)$. When pattern T_i with an even subscript number $i \in \{0, 2, 4, \ldots, 2k-2\}$ is chosen, the following relation should be true:

$$\begin{aligned} max\mathrm{TCD}(T_i) &= i \times \sqrt{N/2} \\ max\mathrm{TCD}(T) &= (i/2) \times \sqrt{N} + (i^2/2) \times \sqrt{N/2}. \end{aligned} \tag{4}$$

The pattern with the even subscript i is chosen so that $\mathrm{HD}(T_i, T_j) = N/2$ between pattern T_i and all previous patterns $T_j, j < i$. The pattern with an odd subscript $i \in \{1, 3, \ldots, 2k-1\}$ is the complement value of the previous pattern with an even subscript (i.e., $T_i = \overline{T}_{i-1}$). Then, both metrics are

$$\begin{aligned} max\mathrm{TCD}(T_i) &= \sqrt{N} + (i-1) \times \sqrt{N/2} \\ max\mathrm{TCD}(T) &= ((i+1)/2) \times \sqrt{N} + ((i^2-1)/2) \times \sqrt{N/2}. \end{aligned} \tag{5}$$

As an example, we take $N = 2^m$, then the number q of patterns within the OCRT $T = \{T_0, T_1, T_2, \ldots T_{q-1}\}$ equals $2(m+1)$. For $m = 3$, OCRT consisting of $2(m+1) = 2(3+1) = 8$ patterns with both metrics ($max\mathrm{TCD}(T_i)$ and $max\mathrm{TCD}(T)$ are presented in Table 1.

For the general case, the number OCRT patterns is calculated as $q = 2(\lceil log_2 N \rceil + 1)$, and the constructive algorithm for pattern generation is presented in [7].

The presented procedure of OCRT construction, based on the greedy-like algorithm, is the best solution that can be obtained according to the procedure of all known algorithms for *controlled random test* generation, namely, *anti-random test* generation [14], *fast anti-random test* generation [25], *orderly random test* construction [28], and numerous modifications, like the *maximal distance anti-random test*, *maximal Hamming distance test*, and so on.

To summarize, the optimal controlled random test generation (Algorithm 1) for construction of OCRT for $N = 2^m$ is proposed. By optimal controlled random test we understand the set of test patterns where each pattern is chosen such that its total distance ($\mathrm{TCD}(T_i)$ and $\mathrm{TCD}(T)$) from all previous patterns is the greatest possible value.

Table 1. Optimal controlled random test for $q = 8$.

T_i	$t_{i,7}$	$t_{i,6}$	$t_{i,5}$	$t_{i,4}$	$t_{i,3}$	$t_{i,2}$	$t_{i,1}$	$t_{i,0}$	$max\text{TCD}(T_i)$	$max\text{TCD}(T)$
T_0	0	0	0	0	0	0	0	0	–	–
T_1	1	1	1	1	1	1	1	1	$\sqrt{8}$	$\sqrt{8}$
T_2	0	0	0	0	1	1	1	1	$2\sqrt{4}$	$\sqrt{8}+2\sqrt{4}$
T_3	1	1	1	1	0	0	0	0	$\sqrt{8}+2\sqrt{4}$	$2\sqrt{8}+4\sqrt{4}$
T_4	0	0	1	1	0	0	1	1	$4\sqrt{4}$	$2\sqrt{8}+8\sqrt{4}$
T_5	1	1	0	0	1	1	0	0	$\sqrt{8}+4\sqrt{4}$	$3\sqrt{8}+12\sqrt{4}$
T_6	0	1	0	1	0	1	0	1	$6\sqrt{4}$	$3\sqrt{8}+18\sqrt{4}$
T_7	1	0	1	0	1	0	1	0	$\sqrt{8}+6\sqrt{4}$	$4\sqrt{8}+24\sqrt{4}$

ALGORITHM 1. Construction of OCRT for $N = 2^m$

1. Initial matrix M with N columns and $q = 2(m+1)$ rows should be constructed based on the trivial divide and conquer algorithm [31]. The first row M_0 takes the value with N zeros, and the second row (M_1) is all ones. The consecutive rows with even subscripts $i > 0$ are the set of even number blocks, where half are all zero blocks and another half are all blocks with one. The next row with an odd subscript is the negation of the previous row with an even subscript (e.g., M_1 is complement of M_0). On each iteration of this algorithm, the current row of matrix M with an even subscript is constructed from the previous row with an even subscript, so that all its blocks are divided into two equal blocks, when the first new block is all zeros and the second is all ones.
2. Based on some permutations, the algorithm conducts column reordering for matrix M to obtain matrix M^*, which is called a mask vector matrix.
3. Randomly chosen N bits pattern P out of 2^N possible patterns is regarded as the first pattern $T_0 = P$.
4. Each new i-th pattern $T_1, T_2, T_3, \ldots, T_{2m+1}$ is the bit-wise exclusive or sum $T_i = P \bigoplus M_i^*$.
5. Repeat Step 4 until all $2(m+1)$ patterns have been constructed.

It should be mentioned that column reordering for any matrix does not change the value of the Hamming distance between any two rows (patterns), which follows from the Eq. (1). We consider the random matrix M with three rows and four columns and the matrix M_1^* that is obtained by column reordering from the matrix M.

M

	C_1	C_2	C_3	C_4
T_0	0	0	1	0
T_1	1	0	0	1
T_2	1	1	0	0

M_1^*

	C_3	C_4	C_1	C_2
T_0^*	1	0	0	0
T_1^*	0	1	1	0
T_2^*	0	0	1	1

We notice that the value of the Hamming distance between any two rows of matrix M equals the distance between corresponding rows of matrix M_1^*:

$$HD(T_0, T_1) = HD(T_0^*, T_1^*) = 3,$$
$$HD(T_0, T_2) = HD(T_0^*, T_2^*) = 3$$
$$HD(T_1, T_2) = HD(T_1^*, T_2^*) = 2.$$

The same observation can be made for the operation used in Step 4 of Algorithm 1. Let us again consider the random matrix M, the random pattern T_r, and the matrix M_2^*, where each row is bit-wise exclusive of the corresponding row of matrix M and pattern T.

M

	C_1	C_2	C_3	C_4
T_0	0	0	1	0
T_1	1	0	0	1
T_2	1	1	0	0

$T_r = 1\ 0\ 1\ 1$

M_2^*

	C_1^*	C_2^*	C_3^*	C_4^*
T_0^*	1	0	0	1
T_1^*	0	0	1	0
T_2^*	0	1	1	1

We again notice that the value of the Hamming distance between any two rows of matrix M equals the distance between the corresponding rows of matrix M_2^*:

$$HD(T_0, T_1) = HD(T_0^*, T_1^*) = 3,$$
$$HD(T_0, T_2) = HD(T_0^*, T_2^*) = 3$$
$$HD(T_1, T_2) = HD(T_1^*, T_2^*) = 2.$$

We conclude that Steps 2–4 of the Algorithm 1 do not affect the main characteristic of the test patterns generated in Step 1. However, at the same time, the above steps allow us to generate different sets of test patterns characterized by maximum distances between the patterns. We can use those patterns to effectively detect different sorts of faults, for example, pattern sensitive faults in RAM.

Now, as an example, we generate OCRT for $N = 2^m = 2^3$ using Algorithm 1,

Example 3.1. OCRT generating for $N = 2^m = 2^3$

1. Initial matrix M with $N = 8$ columns and $q = 8$ rows is constructed, so that $M_0 = 00000000$ and $M_1 = 11111111$. The new M_2 row is obtained from M_0

Table 2. Masks vector matrices M^* for $m = 3$

M_i^*	$t_{i,2}$	$t_{i,6}$	$t_{i,1}$	$t_{i,4}$	$t_{i,3}$	$t_{i,7}$	$t_{i,5}$	$t_{i,0}$
M_0^*	0	0	0	0	0	0	0	0
M_1^*	1	1	1	1	1	1	1	1
M_2^*	1	0	1	0	1	0	0	1
M_3^*	0	1	0	1	0	1	1	0
M_4^*	0	0	1	1	0	0	1	1
M_5^*	1	1	0	0	1	1	0	0
M_6^*	1	1	0	1	0	0	0	1
M_7^*	0	0	1	0	1	1	1	0

by the dividing it on two blocks: one with all zero code and the second with all ones, where M_3 is the complement value of M_2, The next rows of matrices M (M_4, ..., M_7) are obtained in the same way (see Table 1).
2. Mask vector matrices M^*, as a result of column reordering of M (Table 1) are presented in Table 2.
3. If the randomly chosen $N = 8$-bit pattern is $P = 01111010$, then $T_0 = P = 01111010$.
4. Each new i-th pattern $T_1, T_2, T_3, \ldots, T_7$ is the bit-wise exclusive or sum $T_i = P \oplus M_i^*$, where M_i^* is taken from Table 2.
5. All $q = 2(m + 1) = 8$ patterns for OCRT are presented in Table 3 with the corresponding values of the metrics. ■

Table 3. OCRT patterns for $m = 3$

T_i	$t_{i,7}$	$t_{i,6}$	$t_{i,5}$	$t_{i,4}$	$t_{i,3}$	$t_{i,2}$	$t_{i,1}$	$t_{i,0}$	$max\text{TCD}(T_i)$	$max\text{TCD}(T)$
T_0	0	1	1	1	1	0	1	0	–	–
T_1	1	0	0	0	0	1	0	1	$\sqrt{8}$	$\sqrt{8}$
T_2	1	1	0	1	0	0	1	1	$2\sqrt{4}$	$\sqrt{8} + 2\sqrt{4}$
T_3	0	0	1	0	1	1	0	0	$\sqrt{8} + 2\sqrt{4}$	$2\sqrt{8} + 4\sqrt{4}$
T_4	0	1	0	0	1	0	0	1	$4\sqrt{4}$	$2\sqrt{8} + 8\sqrt{4}$
T_5	1	0	1	1	0	1	1	0	$\sqrt{8} + 4\sqrt{4}$	$3\sqrt{8} + 12\sqrt{4}$
T_6	1	0	1	0	1	0	1	1	$6\sqrt{4}$	$3\sqrt{8} + 18\sqrt{4}$
T_7	0	1	0	1	0	1	0	0	$\sqrt{8} + 6\sqrt{4}$	$4\sqrt{8} + 24\sqrt{4}$

The brief analyses demonstrate that the presented procedure can construct a set of OCRT depending on the initial pattern P. For the general case, the number of OCRT patterns is calculated as $q = 2(\lceil log_2 N \rceil + 1)$, and the algorithm for OCRT pattern generation is slightly different for the mask vector matrix generation.

The proposed algorithm seems to be efficient in terms of computation complexity. For any value of N, the most difficult procedure is the mask vector matrix (M) generation and performing column reordering to obtain M^*. Indeed, increasing N by two times needs only two additional rows for M. Then, the new M is constructed so that all N-bit rows for the previous M are extended to $2N$ bits using the simple doubling of all blocks with the same values 0 or 1 and adding two final rows $M_{2m-2} = 010101\ldots01$ and $M_{2m-1} = 101010\ldots10$.

4 Experimental Results

Finally, to confirm the proposed solution we have compared the coverage of several controlled random tests strategy (OCRT, pure antirandom tests [14], concatenated antirandom tests [14], STPG [22]) and random one in terms of

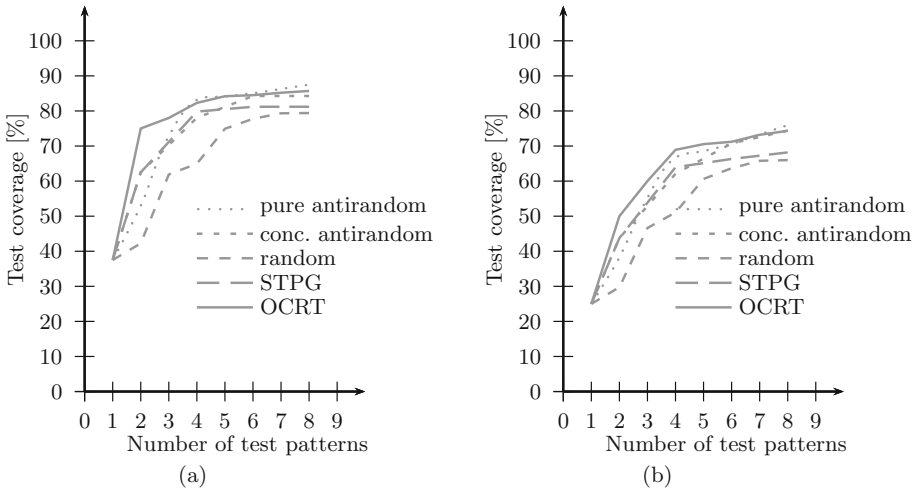

Fig. 1. The coverage of all arbitrary k out of N bits for different antirandom test schemes: (a) k=3 and N=64, (b) k=4 and N=64

number of generated binary combinations for all arbitrary k out of N bits. Using different methods, we had generated test sets consisting of eight test patterns and then we compared their coverage with each other. In the case of pure antirandom tests THD was used as a fitness function whilst concatenated antirandom vectors were employed from Example 7 in [14]. Due to the fact that the authors of STPG algorithm have not indicated how to determine the adding factors [22], a random value was used. The experiments have been done for $k = 3$ and $k = 4$. The obtained results are shown in the Fig. 1a and b. The x-axis represents the number of the test patterns, and the y-axis – the number (in percent) of binary combinations for all arbitrary $k = 3$ and k=4 out of $N = 64$ bits. For Fig. 1, we observe that all coverages curves rise sharply and them exhibits a smooth behavior. We observe high efficiency of OCRT especially for the first few patterns of the test. At the end of the test process OCRT gives us the same or slightly lower (see Fig. 1b) level of fault coverage as in case of pure antirandom patterns. The same time we should noted that OCRT is characterized by easier computational method in compare to other analyzed techniques. Most known antirandom techniques still need a lot of resources (strong CPU for computation of distances between vectors, additional memory to collect generated vectors, etc.) that may be unavailable in case of embedded systems and BIST technique.

5 Conclusions

In the paper a new approach called as Optimal Controlled Random Test has been introduced. It uses previously adopted metrics attempting to make all known Greedy like controlled random testing more effective. Unlike the known solutions the new approach keeps the distances TCD(T_i) and TCD(T) as large as

possible. Both experimental and analytical investigation clearly show the high efficiency of proposed solution especially for the first few iterations of the multi-run test. However the OCRT has an obvious advantageous compare with all known methods. It is more efficient in terms of computation complexity.

References

1. Venkatasubramanian, M., Agrawal, V.D.: Failures guide probabilistic search for a hard-to-find test. In: 25th IEEE North Atlantic Test Workshop, NATW 2016, Providence, RI, USA, May 9–11, 2016, pp. 18–23 (2016)
2. Fujiwara, H., Toida, S.: The complexity of fault detection problems for combinational logic circuits. IEEE Trans. Comput. **31**(6), 555–560 (1982)
3. Venkatasubramanian, M., Agrawal, V.D.: A new test vector search algorithm for a single stuck-at fault using probabilistic correlation. In: IEEE 23rd North Atlantic Test Workshop, NATW 2014, Johnson City, NY, USA, May 14–16, 2014, pp. 57–60 (2014)
4. Tang, D., Woo, L.: Exhaustive test pattern generation with constant weight vectors. IEEE Trans. Comput. **32**, 1145–1150 (1983)
5. Furuya, K.: A probabilistic approach to locally exhaustive testing. Trans. IEICE (Trans. Inst. Electron. Inf. Commun. Eng.) **E72**(5), 656–660 (1989)
6. Kuhn, R.D., Okum, V.: Pseudo-exhaustive testing for software. In: Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop, Computer Society, pp. 153–158. IEEE, Washington, DC (2006)
7. Das, D., Karpovsky, M.: Exhaustive and near-exhaustive memory testing techniques and their BIST implementations. J. Electron. Test. **10**(3), 215–229 (1997)
8. Yarmolik, S.V.: Iterative near pseudoexhaustive random testing. Informatics **2**(26), 66–75 (2010)
9. Duran, J.W., Ntafos, S.C.: An evaluation of random testing. IEEE Trans. Softw. Eng. **10**(4), 438–444 (1984)
10. Grindal, M., Offutt, J., Andler, S.F.: Combination testing strategies - a survey. Technical report ISE-TR-04-05, GMU Technical report, July 2004
11. Seth, S., Agrawal, V., Farhat, H.: A statistical theory of digital circuit testability. IEEE Trans. Comput. **39**, 582–586 (1990)
12. Sosnowski, J.: Experimental evaluation of pseudorandom test effectiveness. In: Proceedings of 24th Euromicro Conference, Vasteras, Sweden, pp. 184–187. IEEE Computer Society (1998)
13. Reid, S.C.: An empirical analysis of equivalence partitioning, boundary value analysis and random testing. In: Proceedings of the 4th International Symposium on Software Metrics, Computer Society, pp. 64–73. IEEE, Washington, DC (1997)
14. Malaiya, Y.K.: Antirandom testing: getting the most out of black-box testing. In: Proceedings of 6th IEEE International Symposium on Software Reliability Engineering, ISSRE 1995, pp. 86–95. IEEE Computer Society (1995)
15. Chen, T.Y., Kuo, F.C., Merkel, R.G., Tse, T.H.: Adaptive random testing: the art of test case diversity. J. Syst. Softw. **83**, 60–66 (2010)
16. Wu, S.H., Jandhyala, S., Malaiya, Y.K., Jayasumana, A.P.: Antirandom testing: a distance-based approach. VLSI Des. **2008**, 1–2 (2008)
17. Feldt, R., Poulding, S.M., Clark, D., Yoo, S.: Test set diameter: quantifying the diversity of sets of test cases. In: 2016 IEEE International Conference on Software Testing, Verification and Validation, ICST 2016, Chicago, IL, USA, April 11–15, 2016, pp. 223–233 (2016)

18. Choi, E.H., Artho, C., Kitamura, T., Mizuno, O., Yamada, A.: Distance-integrated combinatorial testing. In: Proceedings of 27th International Symposium on Software Reliability Engineering (ISSRE 2016), Ottawa, Canada, pp. 93–104, October 2016
19. Mrozek, I.: Analysis of multibackground memory testing techniques. Int. J. Appl. Math. Comput. Sci. **20**(1), 191–205 (2010)
20. Mrozek, I., Yarmolik, V.N.: Antirandom test vectors for BIST in hardware/software systems. Fundam. Inform. **119**(2), 163–185 (2012)
21. Mrozek, I., Yarmolik, V.N.: Iterative antirandom testing. J. Electron. Test **28**(3), 301–315 (2012)
22. Yiunn, D., Bin A'ain, A., Khor, Ghee, J.: Scalable test pattern generation (STPG). In: 2010 IEEE Symposium on Proceedings of the Industrial Electronics Applications, ISIEA 2010, pp. 433–435, October 2010
23. Zhou, Z.: Using coverage information to guide test case selection in adaptive random testing. In: Computer Software and Applications Conference Workshops, pp. 208–213 (2010)
24. Yarmolik, V.N., Mrozek, I., Yarmolik, S.V.: Controlled method of random test synthesis. Autom. Control Comput. Sci. **49**(6), 395–403 (2016)
25. Mayrhauser, A., von, Bai, A., Chen, T., Anderson, C., Hajjar, A.: Fast antirandom (FAR) test generation. In: Proceedings of 3rd IEEE International Symposium on High-Assurance Systems Engineering, HASE 1998, pp. 262–269. IEEE Computer Society, Washington, DC (1998)
26. Mrozek, I., Yarmolik, V.: Methods of synthesis of controlled random tests. In: Saeed, K., Homenda, W. (eds.) CISIM 2016. LNCS, vol. 9842, pp. 429–440. Springer, Cham (2016). doi:10.1007/978-3-319-45378-1_38
27. Yarmolik, V.N., Mrozek, I., Yarmolik, S.V.: Controlled method of random test synthesis. Autom. Control Comput. Sci. **49**(6), 395–403 (2015)
28. Xu, S.: Orderly random testing for both hardware and software. In: Proceedings of the 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing, Washington, DC, pp. 160–167. IEEE Computer Society (2008)
29. Wu, S.H., Malaiya, Y.K., Jayasumana, A.P.: Antirandom vs. pseudorandom testing. In: Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors, ICCD 1998, p. 221 (1998)
30. Xu, S., Chen, J.: Maximum distance testing. In: Asian Test Symposium, pp. 15–20 (2002)
31. Knuth, D.E.: The Art of Computer Programming: Sorting and Searching, vol. 3, 2nd edn. Addison Wesley Longman Publishing Co., Inc, Redwood City (1998)