The time-adaptive statistical testing for random number generators

Boris Ryabko

Institute of Computational Technology of Siberian Branch of Russian Academy of Science and Novosibirsk State University
email: boris@ryabko.net

Abstract

The problem of constructing effective statistical tests for random number generators (RNG) is considered. Currently, there are hundreds of RNG statistical tests that are often combined into so-called batteries, each containing from a dozen to more than one hundred tests. When a battery test is used, it is applied to a sequence generated by the RNG, and the calculation time is determined by the length of the sequence and the number of tests. Generally speaking, the longer the sequence, the smaller deviations from randomness can be found by a specific test. So, when a battery is applied, on the one hand, the "better" tests are in the battery, the more chances to reject a "bad" RNG. On the other hand, the larger the battery, the less time can be spent on each test and, therefore, the shorter the test sequence. In turn, this reduces the ability to find small deviations from randomness. To reduce this trade-off, we propose an adaptive way to use batteries (and other sets) of tests, which requires less time but, in a certain sense, preserves the power of the original battery. We call this method time-adaptive battery of tests.

The suggested method is based on the theorem which describes asymptotic properties of the so-called p-values of tests. Namely, the theorem claims that if the RNG can be modelled by a stationary ergodic source, the value $-log\,\pi(x_1x_2...x_n)/n$ goes to $1-h$ when n grows, where $x_1x_2...$ is the sequence, $\pi(\,)$ is the p- value of the most powerful test, h is the limit Shannon entropy of the stationary ergodic source.

Keywords: statistical test, randomness testing, random number generators, adaptive statistical test, battery of tests.

1. Introduction

Random number generators (RNG) and pseudo-random number generators (PRNG) are widely used in many applications.RNGs are based on physical sources, while pseudo-random numbers are generated by computers. The goal of RNG and PRNG is to generate sequences of binary digits, which are distributed as a result of throwing an “honest” coin, or, more precisely, obey the Bernoulli distribution with parameters $(1/2, 1/2)$. As a rule, for practically used RNG and PRNG this property is verified experimentally with the help of statistical tests developed for this purpose.

Currently, there are more than one hundred applicable statistical tests, as well as dozens RNGs based on different physical processes, and an even greater number of PRNGs based on different mathematical algorithms; see for review [1, 2, 3]. Informally, an ideal RNG should generate sequences that pass all tests. In practice, especially in cryptographic applications, this requirement is formulated as follows: an RNG must pass a so-called battery of statistical tests, that is, some fixed set of tests. When a battery is applied, each test in the test battery is applied separately to the RNG. Among these batteries, we mention the Marshlia's Diehard battery, which contains 16 tests [4], the National Institute of Standards and Technology (NIST) battery of 15 tests [5], several batteries proposed by L'Ecuyer and Simard [2], which contain from 10 to 106 tests and many others (see for review [1, 2, 6]). In addition, these batteries contain many tests that can be used with different values of the parameters, potentially increasing the total number of tests in the battery. Note that practically used RNG should be tested from time to time like any physical equipment, and therefore these test batteries should be used continuously.

How to evaluate large batteries of tests? On the one hand, the larger the test battery, the more likely it is to find flaws in the tested RNG. On the other hand, the larger the battery, the more time is required for testing. (Thus, L'Ecuyer and Simard [2] remark the need for small batteries to increase computational efficiency.) Another view is as follows: in reality, the time available to study any RNG is limited. Given a certain time budget, one can either use more tests and relatively short sequences generated by the RNG, or use fewer tests, but longer sequences and, in turn, this gives more chances to find deviations of randomness of the considered RNG.

In order to reduce this trade-off, we propose a time-adaptive testing of RNG, in which, informally speaking, first all the tests are executed on relatively short sequences generated by the RNG, and then a few "promising" tests are applied for the final testing. Of course, the key question here is which tests are promising. For example, if a battery of two tests is applied to (relatively short) sequences of the same length, it can be assumed that the smaller the p-value, the more promising the test. But a more complicated situation may arise when we have to compare two tests that were applied to sequences of different lengths (for example, the first test was applied to a sequence of length l_1, and the second to a sequence of length of l_2, $l_1 \neq l_2$). We show that if our goal is to choose the most powerful test, then a good strategy is to choose the test i for which the ratio $-\log(p - value_i)/l_i$ is maximum. This recommendation is based on the following theorem: if an RNG can be modelled by a stationary ergodic source, the value $-log\, \pi(x_1x_2...x_n)/n$ goes to $1 - h$, if n grows, where $x_1x_2...$ is a generated sequence, $\pi(\,)$ is the p-value of the most powerful test, h is the limit Shannon entropy of the stationary ergodic source. This theorem plays an important rule in the suggested time-adaptive scheme and will be described in the first part of the paper, whereas the time-adaptive testing will be described afterwords. The description will be illustrated by experiments with the battery Rabbit from [2].

As far as we know, the proposed approach to testing RNGs is new, but the idea of finding the best test among many, testing the tests step by step in an increasing sequence, is widely used in algorithmic information theory, where the notion of random sequence is formally investigated and discussed [7, 8].

2. Hypothesis testing and properties of pi-values

2.1. Notation

We consider RNG which generates a sequence of letters $x = x_1x_2\ ...x_n$, $n \geq 1$, from a finite alphabet $\{0,1\}^n$. Two statistical hypotheses are considered: $H_0 = \{x\ \ \textit{obeys the uniform distribution } (\mu_U) \textit{ on } \{0,1\}^n\ \}$, and the alternative hypothesis $H_1 = \bar{H}_0$, that is, H_1 is the negation of H_0. It is a particular case of the so-called goodness-of-fit problem, and any test for it (that is, a function on $\{0,1\}^n$) is called a test of fit, see [13]. Let τ be a test. Then, by definition, the significance level α equals the probability of the Type I error, $\alpha \in (0,1)$. Denote a critical region of the test τ for the significance level α by $C_\tau(\alpha)$ and let $\bar{C}_\tau(\alpha) = \{0,1\}^n \setminus C_\tau(\alpha)$. (Recall that Type I error occurs if H_0 is true and is rejected. Type II error occurs if H_1 is true, but H_0 is accepted. Besides, for a certain $x = x_1x_2...x_n$ H_0 is rejected if and only if $x \in C_\tau(\alpha)$.)

Suppose that H_1 is true, and the investigated sequence $x = x_1x_2...x_n$ is generated by an (unknown) source ν. By definition, a test τ is consistent, if for any significance level $\alpha \in (0,1)$ the probability of Type II error goes to 0, that is

$$\lim_{n\to\infty} \nu(\bar{C}_\tau(\alpha)) = 0\,. \tag{1}$$

Suppose, that H_1 is true and the sequences $x \in \{0,1\}^n$ obey a certain distribution ν. It is well-known in mathematical statistics that the optimal test (Neyman-Pearson or NP test) is described by the Neyman-Pearson lemma and the critical region of this test is defined as follows:

$$C_{NP}(\alpha) = \{x : \ \mu_U(x)/\nu(x) \le \lambda_\alpha\}\,,$$

where $\alpha \in (0,1)$ is the significance level and the constant λ_α is chosen in such a way that $\mu_U(C_{NP}(\alpha)) = \alpha$, see [13]. (We did not take into account that the set $\{0,1\}^n$ is finite. Strictly speaking, in such a case a randomized test should be used, but in what follows we will consider asymptotic behaviour of tests for large n, and this effect will be negligible). Note that, by definition, $\mu_U(x) = 2^{-n}$ for any $x \in \{0,1\}^n$.

2.2. The p-value and its properties.

The notion of the critical region is connected with the so-called p-value, which we define for the NP-test by the following equation:

$$\pi_{NP}(x) = \mu_U\{y : \nu(y) > \nu(x)\} = |\{y : \nu(y) > \nu(x)\}|/2^n\,. \tag{2}$$

Informally, $\pi_{NP}(x)$ is the probability to meet a random point y which is worse than the observed when considering the null hypothesis.

The NP-test is optimal in the sense that its probability of a Type II error is minimal, but when testing an RNG the alternative distribution is unknown, and, hence, different tests are necessary. Let us consider a certain statistic τ, and define the p-value for this τ and x as follows:

$$\pi_\tau(x) = \mu_U\{y : \tau(y) > \tau(x)\} = |\{y : \tau(y) > \tau(x)\}|/2^n\,. \tag{3}$$

(Note, that the definition π_{NP} in (2) corresponds to this equation if the value $\nu(x)$ is considered as a statistic, i.e. $\tau(x) = \nu(x)$).

2.3. Choice of critical region.

We considered the case where the hypothesis H_0 is rejected if the p-value is small. Some authors suggest to reject H_0 for large p-values, too, that is, to use a two-tailed rather than an upper-tailed test [13]. This question has been discussed by many authors, see, for example, [13], part 30.6 and [2]. We do not address this question directly, but recall how the two-tailed test can be transferred to the upper-tailed one and then will consider the upper-tailed case only.

Suppose that there is a test of fit with the statistic τ and p-value π_τ. Then, for any integer n the critical set of the upper-tailed test is defined by the equation

$$C_\alpha = \{x_1...x_n : \pi_\tau(x_1...x_n) \geq \alpha\}\,, \tag{4}$$

whereas the critical set of the two-tailed test is defined by

$$C_\alpha = \{x_1...x_n : \pi_\tau(x_1...x_n) \geq \alpha/2 \textit{ or } \pi_\tau(x_1...x_n) \leq 1 - \alpha/2\}\,. \tag{5}$$

Now we define the following new statistic

$$\tau^*(x_1...x_n) = \begin{cases} 2\,\pi_\tau(x_1...x_n) & \text{if } \pi_\tau(x_1...x_n) < 1/2 \\ 2\,(1 - \pi_\tau(x_1...x_n)) & \text{if } \pi_\tau(x_1...x_n) \geq 1/2 \end{cases}.$$

We can see that the critical set for the two-tailed test (5) for the statistic τ is be equal to the critical set for the upper-tailed test (4) for τ^* and $\alpha/2$, and, hence, the two-tailed test τ is presented as upper-tailed test based on the statistic τ^*.

2.4. The p-value and Shannon entropy.

It turns out that there exist such tests whose asymptotic behaviour is close to that of the NP-test for any (unknown) stationary ergodic source ν, see [9]. Those tests are based on so-called universal codes (or data-compressors) and are described in [10, 11], where it is shown that they are consistent. We describe those tests in Appendix 1 and show that they are asymptotically optimal. The following theorem describes the asymptotic behaviour of p-values for stationary ergodic sources for NP test and the mentioned above tests which are based on universal codes (see Appendix 1). We use this theorem as the theoretical basis for adaptive statistical testing developed in this paper.

Theorem 1. *i) If ν is a stationary ergodic measure, then, with probability 1,*

$$\lim_{n\to\infty} -\frac{1}{n}\log \pi_{NP}(x) = 1 - h(\nu)\,, \tag{6}$$

where $h(\nu)$ is the Shannon entropy of ν, see for definition [12].

ii) There exists such a statistic τ that for any stationary ergodic measure ν, with probability 1,

$$\lim_{n\to\infty} -\frac{1}{n}\log \pi_{\tau}(x) = 1 - h(\nu)\,, \tag{7}$$

where p-values π_{NP} and π_τ are defined in (2) and (3), correspondingly.

The test of fit τ is described in Appendix 1, the proof of the theorem is given in Appendix 2, but here we note that this theorem gives some idea of the relation between the Shannon entropy of the (unknown) process ν and the required sample size. Indeed, suppose that the NP test is used and the desired significance level is α. Then, we can see that (asymptotically) α should be larger than $\pi_{NP}(x)$ and from (6) we obtain $n > -\log\alpha/(1-h(\nu))$ (for the most powerful test). It is known that the Shannon entropy is 1 if and only if ν is a uniform measure μ_u. Therefore, in a certain sense, the difference $1-h(\nu)$ estimates the distance between the distributions, and the last inequality shows that the sample size becomes infinite if ν approaches a uniform distribution.

The next simple example illustrates this theorem. Let there be a test τ and a generator (a measure ν) created sequences of binary digits which are independent and, say, $\nu(0) = 0.501, \nu(1) = 0.499$. Suppose, $\lim_{n\to\infty} -\frac{1}{n} \log \pi_\tau(x) = c$, where c is a positive constant. Let us consider the following "decimation test" $\tau^{1/2}$: an input sequence $x_1 x_2 .... x_n$ is transformed into $x_1 x_3 x_5 ... x_{2\lfloor n/2\rfloor -1}$ and then τ is applied to this transformed sequence. Obviously, for this test $\lim_{n\to\infty} -\frac{1}{n/2}\log \pi_{\tau^{1/2}}(x) = c$, and, hence, $\lim_{n\to\infty} -\frac{1}{n} \log \pi_{\tau^{1/2}}(x) = c/2$. Thus, the value $-\frac{1}{n}\log \pi_\tau(x_1 ... x_n)$ seems to be a reasonable estimate of the power of the test for a large n.

3. Time-adaptive statistical tests and their experimental investigation

3.1. Batteries of tests.

Let us consider a situation where the randomness testing is performed by conducting a battery of statistical tests for randomness. Suppose that the battery contains s tests and α_i is the significance level of $i-$th test, $i = 1, ..., s$. If the battery is applied in such a way that the hypothesis H_0 is rejected when at least one test in the battery rejects it, then the significance level α of this battery satisfies the following inequality:

$$\alpha \leq \sum_{i=1}^{s} \alpha_i \,. \tag{8}$$

If all the tests in the battery are independent, then the following equation is valid: $\alpha = 1 - \prod_{i=1}^{s}(1-\alpha_i)$. Clearly, the upper bound (8) is true for this case and $1-\prod_{i=1}^{s}(1-\alpha_i)$ is close to $\sum_{i=1}^{s}\alpha_i$, if each α_i is much smaller than $1/s$. That is why we will use the estimate (8) below.

We have considered a scenario in which a test is applied to a single sequence generated by an RNG, and then the researcher makes a decision on the RNG based on the test results. Another possibility that has been considered by several authors, e.g. [2, 5], is to use the following two-step procedure for testing RNGs. The idea is to generate r sequences $x^1, x^2, ..., x^r$ and apply one test (say, τ) to each of them independently. Then apply another test to the received data $\tau(x^1), \tau(x^2), ..., \tau(x^r)$ (as a rule, those values are converted into a sequence of corresponding p-values, and then the hypothesis of the uniform distribution of those p-values is tested). Then this procedure is repeated for the second test in the battery, and so on. The final decision is made on the basis of the results obtained. We do not consider this two-step procedure in detail, but note that time-adaptive testing can be applied in this situation, too.

3.2. The scheme of the time-adaptive testing.

Let there be an RNG which generates binary sequences, and a battery of s tests with statistics $\tau_1, \tau_2, ..., \tau_s$. In addition, suppose that the total available testing time is limited to a certain amount T and the level of significance is $\alpha \in (0, 1)$.

When the time-adaptive testing is applied, all the calculations are separated into a preliminary stage and a final one. The result of the preliminary stage is the list of values

$$\gamma_1 = \frac{-\log \pi_{\tau_1}(x_1^1 x_2^1 ... x_{n_1}^1)}{n_1}, \gamma_2 = \frac{-\log \pi_{\tau_2}(x_1^2 x_2^2 ... x_{n_2}^2)}{n_2}$$

$$, ..., \ \gamma_s = \frac{-\log \pi_{\tau_s}(x_1^s x_2^s ... x_{n_s}^s)}{n_s} \,, \tag{9}$$

where the sequences $x_1^1 x_2^1 ... x_{n_1}^1,\ ..., x_1^s x_2^s ... x_{n_s}^s$ may have common parts (for example, the first sequence may be the prefix of the second, etc.). Then, taking into account the values (9), it is possible to choose some tests from the battery and apply them to the longer sequence, calculate new values γ, and so on. When the preliminary stage is carried out, several tests from the battery should be chosen for the next stage.

The final stage is as follows. First, we divide the significance level α into $\alpha_1, \alpha_2, ..., \alpha_k$ in such a way that $\sum_{i=1}^k \alpha_i = \alpha$. Then, we obtain new sequence(s) $y_1^1 y_2^1 ... y_{m_1}^1, ..., y_1^k y_2^k ... y_{m_k}^k$, which may have common parts, but are independent of $x_1^1 x_2^1 ... x_{n_1}^1,\ ..., x_1^s x_2^s ... x_{n_s}^s$ and calculate

$$\pi_{\tau_{i_1}}(y_1^1 y_2^1 ... y_{m_1}^1), ..., \pi_{\tau_{i_k}}(y_1^k y_2^k ... y_{m_k}^k)\,. \tag{10}$$

The hypothesis H_0 will be accepted, if $\pi_{\tau_{i_j}}(y_1^j y_2^j ... y_{m_j}^j) > \alpha_j$ for all $j = 1, ..., k$. Otherwise, H_0 is rejected. The parameters of the test should be chosen in such a way that the total time of calculation is not grater than the given limit T.

Claim 1. *The significance level of the described time-adaptive test is not larger than α.*

Indeed, the sequences $y_1^1 y_2^1 ... y_{m_1}^1, ..., y_1^k y_2^k ... y_{m_k}^k$ and $x_1^1 x_2^1 ... x_{n_1}^1,\ ..., x_1^s x_2^s ... x_{n_s}^s$ are independent and, hence, the results of the final stage does not depend on the preliminary one. When the test battery $\tau_{i_1}, \tau_{i_2}, ..., \tau_{i_k}$ is applied, the significance level of τ_{i_j} equals α_j and the significance level of the battery equals $\sum_{i=1}^k \alpha_i$. From (8) we can see that the significance level of the battery (and, hence, of the described testing) is not grater than α.

Comment. The length of the sequences may depend on the speed of tests. For example, it can be done as follows: let v_i be the speed per bit of the test τ_i, $i = 1, ..., s$. One possible way to take into account the speed difference is to calculate

$$\hat{\gamma}_i = \frac{-\log \pi_{\tau_i}(x_1^i x_2^i ... x_{n_i}^i)}{n_i / v_i}, \quad i = 1, ..., s,$$

instead of (9) and similar expressions.

3.3. The experiments.

We carried out some experiments with the time-adaptive test basing on the battery Rabbit from [2], which contains 26 tests. Let us first describe the choice of the RNG for our experiments. Nowadays there are many “bad” PRNGs and “good” ones. In other words, the output sequences of some known PRNGs have some deviations from randomness, which are quite easy to detect with many known tests, while other PRNGs do not have deviations that can be detected by known tests [2]. So, we need to have some families of RNGs with such deviations from randomness that they can be detected only for quite large output sequences. To do this, we take a good generator MRG32k3a and a bad one LCG from [2], generate sequences $g_1 g_2 ...$ and $b_1 b_2 ...$ by these two generators and then prepared a ”mixed” sequence $m_1 m_2 ...$ in such a way that

$$m_i = \begin{cases} g_i & \text{if } i \mod D \neq 0 \\ b_i & \text{if } i \mod D = 0 \end{cases} \tag{11}$$

where D is a parameter.

The time-adaptive testing was organised as follows: during the preliminary stage we first generated a file $m_1 m_2 ... m_{l_1}$ with $l_1 = 2\,000\,000$ bytes, tested it by 25 tests from the Rabbit battery and calculated the values (9) with $\log \equiv \log_2$, see the left part of Table 1. (This battery contains 26 tests, but one of them cannot be applied to such a short sequence.) Then we chose 5 tests with the biggest value $-log\, \pi_{t_i}(m_1 ... m_{l_i})/l_1$ (let they be $t_{i_1}, ..., t_{i_5}$), generated a sequence $m_1 ... m_{l_2}$ with $l_2 = 6\,000\,000$ bytes and applied the tests $t_{i_1}, ..., t_{i_5}$ for testing this sequence (see the example in the right part of Table 1). After that we found a test t_f for which

$$-log \pi_{t_f}/l_f = \max_{r=1,...,25;\, j=i_1...i_5} \{-log \pi_r(m_1 ... m_{l_1})/l_1, -log \pi_j(m_1 ... m_{l_2})/l_2\} .$$

(In other words, for t_f the value $-log \pi_r(m_1 ... m_{l_k})/l_k$ is maximal for $k = 1, 2$ and all r (see the Table 1). The preliminary stage was finished. Then, during the second stage, we generated a 40 000 000 byte sequence, and applied the test t_f to it. If the obtained p-value was less than 0.001, the hypothesis H_0 was rejected. (Note that the sequence length $l_1 = 2\,000\,000$ and $l_2 = 6\,000\,000$ are 5% and 15% from the final length of 40 000 000 bytes. So, the total length of the sequences tested by all the tests during the preliminary stage is $25 \times 0.05 + 5 \times 0.15 = 2$ the final length, i.e. $2 \times 40\,000\,000$. On the other hand, if one applies the battery Rabbit to the sequence of the same length, the total length of investigated sequences is $25 \times 40\,000\,000$, i.e. 8,33 times more.

Let us consider one example in detail, taking $D = 2$ in (11).

Table 1 contains the results of all the calculations carried out during the preliminary stage. So, we can see that the value $-\log_2 \pi)/l$ is maximal for the test $t13$. Hence, at the final stage we applied the test $t13$ to the new 4 000 000 000-byte sequence. It turned out that $\pi_{t13} = 2.9\ 10^{-26}$ and, hence, H_0 is rejected. Besides, we estimated time of all calculation (during both stages).

After that, we conducted an additional experiment to get the full picture. Namely, we calculated p-values for all tests and for the same 4 000 000 000-byte sequence and the estimated the total time of calculations. It turned out that the p-values of the two tests were less than 0.001. Namely, $\pi_{t13} = 2.9\ 10^{-26}$, $\pi_{t22} = 1.1\ 10^{-6}$. Besides, we estimated time of calculations for all experiments. So, the described time-adaptive testing revealed one of the two most powerful tests, while the time used is 8 times.

We carried out similar experiments 20 times for $d = 2, 3, 4$ (in (11)) with different good and bad generators from [2]. Besides, we investigated several modifications of the considered scheme. In particular, we considered a case where during the preliminary stage we, as before, first chose 5 the best tests and them two of the best tests for the finale stage (instead of one, as in the experiment above). It turned out, that in all the cases considered the battery Rabbit rejects H_0 and the time-adaptive testing rejected H_0, too.

4. Conclusion

First of all, we note that the proposed time-adaptive testing does not suggest exact values of numerous parameters. Among these parameters, we note the number of steps at the preliminary stage (in the considered example there were two such steps: selecting five tests and then one), the number of tests compared in one step, the length of the tested sequences, the rule for choosing tests at different stages, etc. The problem of choosing the parameters may be considered a problem of multidimensional optimization. There are many methods available for solving such problems (for example, neural networks and other AI algorithms), and some of them can be used along with the time-adaptive testing.

As far as we know, no one has applied adaptive methods for testing randomness, but there are several well-known approaches that can be considered as steps in this direction. For example, L'Ecuyer and Simard recommend several batteries of different sizes that require different times (and the investigator may use them depending on how much time he has) [2]. Another popular battery recommended for cryptographic applications also has some parameters that allow one to adjust the testing time [5].

We believe that the proposed approach makes it possible to investigate and optimize time-adaptive testing.

5. Appendix 1. Consistent tests based on universal codes.

The considered tests are based on so-called universal codes, that is why we first briefly describe them. For any integer m a code ϕ is defined as such a map from the set of m-letter words to the set of all binary words that for any m-letter u and v $\phi(u) \neq \phi(u)$. This property gives a possibility to uniquely decode. (More formally, ϕ is injective mapping from $\{0,1\}^m$ to $\{0,1\}^*$, where $\{0,1\}^* = \bigcup_{i=1}^{\infty}\{0,1\}^i$.) We will consider so-called universal codes which have the two following properties:

$$\forall\, m > 0 \quad \sum_{u \in \{0,1\}^m} 2^{-|\phi(u)|} = 1 \tag{12}$$

and for any stationary ergodic ν defined on the set of all infinite binary words $x = x_1x_2...$, with probability one

$$\lim_{n\to\infty} \frac{1}{n} |\phi(x_1x_2...x_n)|/n = h(\nu) \tag{13}$$

where $h(\nu)$ is the Shannon entropy of ν. Such code exist, see [12]. Note, that a goal of codes is to " compress " sequences, i.e. make an average length of the codeword $\phi(x_1x_2...x_n)$ as small as possible. The second property (13) shows that the universal codes are asymptotically optimal, because the Shannon entropy is a low bound of the length of the compressed sequence (per letter), see [12].

Let us back to considered problem of hypothesis testing. Suppose, it is known that a sample sequence $x = x_1x_2...$ was generated by stationary ergodic source and, as before, we consider the same H_0 against the same H_1. Let ϕ be a universal code. The following test is suggested in [10]:

If the length $|\phi(x_1...x_n) \leq n - \log_2 \alpha$ then H_0 is rejected, otherwise accepted. Here, as before, α is the significance level, $|\phi(x_1...x_n)|$ is the length of encoded ("compressed") sequence. We denote this test by T_ϕ and its statistic by τ_ϕ, i.e.

$$\tau_\phi(x_1...x_n) = n - |\phi(x_1...x_n)| \,. \tag{14}$$

The following theorem is proven in [10, 11]:

Theorem 2. *For each stationary ergodic ν, alpha $\in (0,1)$ and a universal code ϕ, with probability 1 the Type I error of the described test is not larger than α and the Type II error goes to 0, when $n \to \infty$.*

6. Appendix 2. Proofs.

Proof of Theorem 1. The known Shannon-McMillan-Breiman (SMB) theorem claims that for the stationary ergodic source ν and any $\epsilon > 0, \delta > 0$ there exists such n' that

$$\nu\{x : x \in \{0,1\}^n \ \& \ h(\nu) - \epsilon < -\frac{1}{n} \log \nu(x) < h(\nu) + \epsilon \ \} > 1 - \delta \qquad (15)$$

for $n > n'$, see [12]. From this we obtain

$$\nu\{x : x \in \{0,1\}^n \ \& \ 2^{-n(h(\nu)-\epsilon)} > \ \nu(x) > 2^{-n(h(\nu)+\epsilon)}\} > 1 - \delta \qquad (16)$$

for $n > n'$. It will be convenient to define

$$\Phi_{\epsilon,\delta,n} = \{x : x \in \{0,1\}^n \ \& \ h(\nu) - \epsilon < -\frac{1}{n} \log \nu(x) < h(\nu) + \epsilon \ \} \qquad (17)$$

From this definition and (16) we obtain

$$(1-\delta)\, 2^{n(h(\nu)-\epsilon)} \le |\Phi_{\epsilon,\delta,n}| \le 2^{n(h(\nu)+\epsilon)} \,. \qquad (18)$$

For any $x \in \Phi_{\epsilon,\delta,n}$ define

$$\Lambda_x = \{y : \nu(y) > \nu(x) \ \} \bigcap \Phi_{\epsilon,\delta,n} \,. \qquad (19)$$

Note that, by definition, $|\Lambda_x| \le |\Phi_{\epsilon,\delta,n}|$ and from (18) we obtain

$$|\Lambda_x| \le 2^{n(h(\nu)+\epsilon)} \,. \qquad (20)$$

For any $\rho \in (0,1)$ we define $\Psi_\rho \subset \Phi_{\epsilon,\delta,n}$ such that

$$\nu(\Psi_\rho) = \rho \ \& \ \forall u \in \Psi_\rho \forall v \in (\Phi_{\epsilon,\delta,n} \setminus \Psi_\rho) \implies \nu(u)) \ge \nu(v) \,. \qquad (21)$$

(That is, Ψ_ρ contains the most probable words whose total probability equals ρ.) Let us consider any $x \in (\Phi_{\epsilon,\delta,n} \setminus \Psi_\rho)$. Taking into account the definition (21) and (18) we can see that for this x

$$|\Lambda_x| \ge \rho |\Phi_{\epsilon,\delta,n}| \ge \rho(1-\delta) 2^{n(h(\nu)-\epsilon)} \,. \qquad (22)$$

So, from this inequality and (20) we obtain

$$\rho(1-\delta)2^{n(h(\nu)-\epsilon)} \leq |\Lambda_x| \leq \; 2^{n(h(\nu)+\epsilon)} \,. \qquad (23)$$

From equation (16), (17) and (21) we can see that $\nu(\Phi_{\epsilon,\delta,n} \setminus \Psi_\rho) \geq (1-\delta)(1-\rho)$. Taking into account (23) and this inequality, we can see that

$$\nu\{x : x \in \{0,1\}^n \& h(\nu)-\epsilon-\log(\rho(1-\delta))/n \leq \log|\Lambda_x|/n \leq h(\nu)+\epsilon\} \geq (1-\delta)(1-\rho). \qquad (24)$$

From the definition (2) of $\pi_{NP}(x)$ and the definition (19) of Λ_x, we can see that $\pi_{NP}(x) = |\Lambda_x|/2^n$. Taking into account this equation and (24) we obtain the following:

$$\nu\{x : x \in \{0,1\}^n \,\&\, 1-(h(\nu)-\epsilon-\log(\rho(1-\delta))/n) \geq$$

$$-\log \pi_{NP}(x)/n \geq 1-(h(\nu)+\epsilon)\} \geq (1-\delta)(1-\rho). \qquad (25)$$

Having taken into account that this inequality is valid for all positive ϵ, δ and ρ we obtain the first statement of the theorem.

The proof of the second statement of the theorem is closed to the previous one. First, from the theorem 2 we see that for any $\epsilon > 0, \delta > 0$ we define

$$\hat{\Phi}_{\epsilon,\delta,n} = \{x : h(\nu)-\epsilon < |\phi(x_1...x_n)|/n < h(\nu)+\epsilon\} \,. \qquad (26)$$

Note that from (13) we can see that there exists such n'' that, for $n > n''$,

$$\nu(\hat{\Phi}_{\epsilon,\delta,n}) > 1-\delta \,. \qquad (27)$$

We will use the set $\Phi_{\epsilon,\delta,n}$ (see (17)). Having taken into account the SMB theorem (15) and (27), we can see that

$$\nu(\hat{\Phi}_{\epsilon,\delta,n} \cap \Phi_{\epsilon,\delta,n}) > 1-2\delta \,, \qquad (28)$$

if $n > \max(n', n'')$.

From this moment, the proof begins to repeat the proof of the first statement if we use the set $(\hat{\Phi}_{\epsilon,\delta,n} \cap \Phi_{\epsilon,\delta,n})$ instead of $\Phi_{\epsilon,\delta,n}$. The only difference is in the definitions (19) and (21) which should be changed as follows.

$$\Lambda_x = \{y : |\phi|(y)| < |\phi(x)| \,\} \cap (\hat{\Phi}_{\epsilon,\delta,n} \cap \Phi_{\epsilon,\delta,n})$$

and Ψ_ρ is such a subset of $(\hat{\Phi}_{\epsilon,\delta,n} \cap \Phi_{\epsilon,\delta,n})$ that

$$\nu(\Psi_\rho) = \rho \,\&\, \forall u \in \Psi_\rho \forall v \in (\Phi_{\epsilon,\delta,n} \setminus \Psi_\rho) \implies |\phi(u)| \leq |\phi(v)| \,.$$

If we replace π_{NP} with π_{τ_ϕ} and δ with 2δ, we obtain the proof of the second statement. Theorem is proven.

Acknowledgment

This work was supported by Russian Foundation for Basic Research (grant 18-29-03005).

References

[1] L'Ecuyer, P. History of uniform random number generation. In Proceedings of the WSC 2017-Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017.

[2] P. L'Ecuyer and R. Simard, "TestU01: AC library for empirical testing of random number generators," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, p.22, 2007.

[3] Herrero-Collantes, M., Garcia-Escartin, J.C. Quantum random number generators, *Rev. Mod. Phys.*, vol. 89, 015004, 2017.

[4] Marsaglia, G. Xorshift rngs. *J. Stat. Softw.* **2003**, *8*, 1–6.

[5] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.* National Institute of Standards and Technology, 2010.

[6] Demirhan, H., Bitirim, N. Statistical Testing of Cryptographic Randomness. *Journal of Statisticians: Statistics and Actuarial Sciences, IDIA* 2016, 9, 1, 1-11.

[7] Calude, C.S. *Information and Randomness—An Algorithmic Perspective*; Springer-Verlag: Berlin/Heidelberg, Germany, 2002.

[8] Downey, R.; Hirschfeldt, D.R.; Nies, A.; Terwijn, S.A. Calibrating randomness. *Bull. Symb. Log.* **2006**, *12*, 411–491.

[9] B. Ryabko, " On asymptotically optimal tests for random number generators," arXiv:1912.06542 [cs.IT], 2019.

[10] B. Ryabko and J. Astola, " Universal Codes as a Basis for Time Series Testing," *Statistical Methodology*, vol.3, pp. 375-397, 2006.

[11] B. Ryabko, J. Astola, M. Malyutov, *Compression-Based Methods of Statistical Analysis and Prediction of Time Series,* Springer International Publishing Switzerland, 2016.

[12] T. M. Cover and J. A. Thomas, *Elements of information theory.* New York, NY, USA: Wiley-Interscience, 2006.

[13] M. Kendall, A. Stuart, *The advanced theory of statistics; Vol.2: Inference and Relationship*; Hafner Publishing Company: New York, NY, USA, 1961.

Table 1: Time-adaptive testing. Preliminary stage.

test	length (l) (bytes)	p-value (π)	$-\log_2 \pi/l$	length (l) (bytes)	p-value	$-\log_2 \pi/l$
t1	$2\,10^6$	0.42	$6.3\ 10^{-7}$			
t2	$2\,10^6$	0.37	$7.3\ 10^{-7}$			
t3	$2\,10^6$	0.028	$26\ 10^{-7}$	$6\,10^6$	0,23	$3.6\ 10^{-7}$
t4	$2\,10^6$	0.78	$1.8\,10^{-7}$			
t5	$2\,10^6$	0.4	$6.5\,10^{-7}$			
t6	$2\,10^6$	0.37	$7.2\,10^{-7}$			
t7	$2\,10^6$	0.059	$20\,10^{-7}$			
t8	$2\,10^6$	0.026	$26\,10^{-7}$	$6\,10^6$	0.0037	$26\ 10^{-7}$
t9	$2\,10^6$	0.72	$2.4\,10^{-7}$			
t10	$2\,10^6$	0.72	$2.4\,10^{-7}$			
t11	$2\,10^6$	0.63	$3.3\,10^{-7}$			
t12	$2\,10^6$	0.74	$2.2\,10^{-7}$			
t13	$2\,10^6$	0.021	$28\,10^{-7}$	$6\,10^6$	0.0028	$14\,10^{-7}$
t14	$2\,10^6$	0.42	$6.2\,10^{-7}$			
t15	$2\,10^6$	0.9	$0.74\,10^{-7}$			
t16	$2\,10^6$	0.087	$18\,10^{-7}$			
t17	$2\,10^6$	0.72	$2.3\,10^{-7}$			
t18	$2\,10^6$	0.58	$3.9\,10^{-7}$			
t19	$2\,10^6$	0.89	$0.81\,10^{-7}$			
t20	$2\,10^6$	0.51	$4.9\,10^{-7}$			
t21	$2\,10^6$	0.047	$22\,10^{-7}$	$6\,10^6$	0.73	$0.76\,10^{-7}$
t22	$2\,10^6$	0.47	$0.47\,10^{-7}$			
t23	$2\,10^6$	0.18	$12\,10^{-7}$			
t24	$2\,10^6$	0.14	$14\,10^{-7}$			
t25	$2\,10^6$	0.024	$27\,10^{-7}$	$6\,10^6$	0.05	$7.2\,10^{-7}$