The time-adaptive statistical testing for random number generators

Ryabko Boris
Institute of Computational Technologies of SB RAS
Novosibirsk State University
Novosibirsk, Russian Federation
Email: boris@ryabko.net

Zhuravlev Viacheslav
Novosibirsk State University
Novosibirsk, Russian Federation
Email: slawajur@mail.ru

***Abstract*—Currently, there are dozens of random number generators (RNGs) and hundreds of statistical tests designed to test the generators. These tests are often combined into so-called batteries, each of which contains from a dozen to more than a 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 that can be used in such a way as to increase the testing power.**

I. 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, to 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 of RNGs based on different physical processes, and an even greater number of PRNGs based on different mathematical algorithms; see for review [1]–[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 Marsaglia'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. 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 the randomness of the considered RNG.

In order to reduce this trade-off, we propose time-adaptive testing of RNGs, 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 the 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 [7]: 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.

In this paper we describe a time-adaptive test and some experiments designed to study its capabilities. We first look at a case where the goal is to reduce testing time. It turns out that testing time for the adaptive test can be done much less than on the original battery without losing test power. Secondly, we consider the case when the testing time is fixed. It turns out

that the power of the adaptive test may be greater than that of the original battery of tests.

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 the random sequence is formally investigated and discussed [8], [9].

II. Time-adaptive statistical tests

A. 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:

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

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 (1) 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 (1) 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.

B. 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

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

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 (2), 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

$$\pi_{\tau_{i_1}}(y_1^1 y_2^1 ... y_{m_1}^1), ..., \pi_{\tau_{i_k}}(y_1^k y_2^k ... y_{m_k}^k) \,. \tag{3}$$

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 greater than the given limit of T.

Theorem. 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 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 (1) 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

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

instead of (2) and similar expressions.

III. The experiments.

We carried out some experiments with the time-adaptive test basing on the battery Rabbit from [2] in order to compare the original Rabbit battery and its new adaptive test.

Let us first describe the choice of 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 the 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 the 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

$$m_i = \begin{cases} g_i & \text{if } i \mod D \neq 0 \\ b_i & \text{if } i \mod D = 0 \end{cases} \tag{4}$$

where D is a parameter. In different experiments, we used different "good" generators (MRG32k3a, Java 48, lfsr 113) and "bad" ones (LCG, Taus and Visual Basic) from [2].

A. Experiments designed to reduce calculation time

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 (2) 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 (4).

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 $40\,000\,000$-byte sequence. It turned out that $\pi_{t13} = 2.9\ 10^{-26}$ and, hence, H_0 is rejected. Besides, we estimated the time of all calculations (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 $40\,000\,000$-byte sequence and the estimated 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 the 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 (4)) 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 cases considered the battery Rabbit rejects H_0 and the time-adaptive testing rejected H_0, too.

TABLE I
TIME-ADAPTIVE TESTING. PRELIMINARY STAGE.

test	length (l) (bytes)	p-value (π)	$-\log \pi/l$	length (l) (bytes)	p-value	$-\log \pi/l$
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	$7.2\,10^{-7}$

B. Experiments designed to improve the quality of testing.

In the experiments described here, we first applied the original Rabbit battery to a specific generator, and then time-adaptive testing so that the total size of the investigated files was the same for both tests. For both cases, we took a significance level of 0.001. Then we generated 26 different files, every 120 megabytes (MB) in size, and applied one battery test to one file (note that the total size of all files is 3120 MB.). If at least one p-value was less than $0.001/26 (= 0.0000385)$, H_0 was rejected, otherwise accepted.

The time adaptive testing was as follows. As in the previous section, we applied a two-step preliminary stage. First, we generated 26 files with a length of 50 MB each, then we selected 5 tests with a minimum p-value and applied these tests to five new files with a length of 150 MB. Then, based on the results obtained, we select one test with a maximum value of $-\log \pi()/length)$ and use this test for a file of size 1000 MB (= 1 GB). If the p-value for this test was less than 0.001, H_0 was rejected, otherwise accepted. (Note that the total size of all files is 3050 MB, i.e., slightly less than 3120 when the Rabbit battery was used). The main results are presented in Table 2.

This table shows that there are several cases where both batteries either accept or reject H_0 together, and there are

TABLE II
RESULTS OF EXPERIMENTS FOR THE BATTERY RABBIT. (HERE H_0 MEANS H_0 IS ACCEPTED, H_1 MEANS H_0 IS REJECTED.)

Bad generator	Good generator	D in (4)	Result of adaptive testing	Result of original battery
LCG	Java 48	4	H_1	H_1
LCG	Java 48	16	H_1	H_1
LCG	Java 48	64	H_1	H_1
LCG	Java 48	256	H_1	H_0
LCG	Java 48	1024	H_0	H_0
LCG	lfsr 113	4	H_1	H_1
LCG	lfsr 113	16	H_1	H_1
LCG	lfsr 113	64	H_1	H_1
LCG	lfsr 113	256	H_1	H_0
LCG	lfsr 113	1024	H_0	H_0
LCG	MRG 32 k3a	4	H_1	H_1
LCG	MRG 32 k3a	16	H_1	H_0
LCG	MRG 32 k3a	64	H_0	H_0
LCG	MRG 32 k3a	256	H_0	H_0
LCG	MRG 32 k3a	1024	H_0	H_0
Taus	lfsr 113	4	H_1	H_1
Taus	lfsr 113	16	H_1	H_1
Taus	lfsr 113	64	H_1	H_1
Taus	lfsr 113	256	H_1	H_0
Taus	lfsr 113	1024	H_1	H_1
Taus	Java 48	4	H_1	H_1
Taus	Java 48	16	H_1	H_1
Taus	Java 48	64	H_1	H_1
Taus	Java 48	256	H_0	H_0
Taus	Java 48	1024	H_0	H_0
Taus	MRG 32 k3a	4	H_1	H_1
Taus	MRG 32 k3a	16	H_1	H_1
Taus	MRG 32 k3a	64	H_1	H_1
Taus	MRG 32 k3a	256	H_0	H_0
Taus	MRG 32 k3a	1024	H_0	H_0
Visual Basic	Java 48	4	H_1	H_1
Visual Basic	Java 48	16	H_1	H_1
Visual Basic	Java 48	64	H_1	H_1
Visual Basic	Java 48	256	H_1	H_1
Visual Basic	Java 48	1024	H_1	H_0
Visual Basic	lfsr 113	4	H_1	H_1
Visual Basic	lfsr 113	16	H_1	H_1
Visual Basic	lfsr 113	64	H_1	H_1
Visual Basic	lfsr 113	256	H_1	H_1
Visual Basic	lfsr 113	1024	H_1	H_0
Visual Basic	MRG 32 k3a	4	H_1	H_1
Visual Basic	MRG 32 k3a	16	H_1	H_1
Visual Basic	MRG 32 k3a	64	H_1	H_0
Visual Basic	MRG 32 k3a	256	H_0	H_0
Visual Basic	MRG 32 k3a	1024	H_0	H_0

cases where the adaptive battery rejects H_0, while the original battery accepts this hypothesis. Taking into account that in both cases the significance level (0.001) was the same, and the total sizes of the studied files were very close, we see that there are situations when the adaptive testing detects deviations from randomness, while the original battery does not find them.

IV. CONCLUSION

In this article, we showed that the proposed time-adaptive testing is promising for RNG testing. On the other hand, we note that the proposed time-adaptive testing does not offer exact values of numerous parameters for all possible batteries. 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 parameter selection can be considered as a problem of multidimensional optimization. There are many methods available to solve such problems (for example, neural networks and other AI algorithms), and some of them can be used along with time-adaptive testing.

We believe that the proposed approach, combined with multidimensional optimization, allows researchers to investigate and optimize time-adaptive testing.

ACKNOWLEDGEMENT

The research was supported by the Russian Foundation for Basic Research (grant no. 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] B. Ryabko, " On asymptotically optimal tests for random number generators," arXiv:1912.06542 [cs.IT], 2019.

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

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

[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.