Volume XXII 2019
ISSUE no.2
MBNA Publishing House Constanta 2019

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Multi-criteria method for evaluation of the pseudorandom number generators using thermodynamic systems behavior

To cite this article: V. Cornaciu and C. Răcuciu, Scientific Bulletin of Naval Academy, Vol. XXII 2019, pg. 305-312.

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-19-I2-036

Multi-criteria method for evaluation of the pseudorandom number generators using thermodynamic systems behavior

V. Cornaciu[1] and C. Răcuciu[2]

[1]PhD Candidate, Military Technical Academy-Electronic, Information and Communications Systems for Defense and Security, George Coșbuc nr. 39-49, Bucharest, 050141, Romania.

[2]Prof. Eng. PhD, Titu Maiorescu University-Faculty of Computer Science, Văcărești nr. 187, Bucharest, 004051, Romania.

[1]veronica_zanfir@yahoo.com

[2]ciprian.racuciu@gmail.com

Abstract. The choice of pseudo random number generators is a major problem in many areas of activity, one of the areas that uses them intensively it is the field of cryptography.
Although they are applicable to any programming paradigm, these generators can be used successfully in the statistical analysis of thermodynamic behaviors, so, they must be designed individually to meet the requirements of each type of client. The research for new pseudo-random number generators with higher level of security is a field in great expansion, but the factors that influence us to use a product are not always this high level of security, but features such as ease of implementation or rapidity of generation. The design of a pseudo-random number generator needs to consider various characteristics simultaneously, which can be regarded as a optimization problem. The purpose of the article is to give a overview of the characteristics that a pseudo-random generator must meet in different fields, to define an objective function that encompasses these features, and create a mathematical optimization problem in order to achieve the maximum of properties that a pseudo-random number generator can give.

Keywords: Pseudo-random generators, optimization problem.

1. Introduction

Randomness is one of the fundamental computational resources and appears everywhere. In general, we cannot talk of a single random number other than in a statistical context. The correct term is that of random numbers. Using the computer, reduces the term of random numbers to a sequence of randomly generated bites, grouped by a certain rule. Mathematically, there is no shorter way to specify the string than the sequence itself.

A random number generation is done by collecting and processing data obtained from a source of entropy outside the computer. The source of entropy can be very simple, such as variations of

mouse movements, or the time interval between pressing two keys. Very good sources of entropy can be radioactive sources or the ones using noise from the atmosphere.

The property of being random was introduced into computers with the help of generators of pseudo-random numbers (PRNG). There are many ways to define pseudo-random number generators. The usual form to define a PRNG is by a deterministic recurrent sequence in a finite field or ring, with an output function which is mapping each state to an input value. This value is often either a real number in the interval (0, 1) or an integer in some finite range [1].

Random numbers are applied in several sciences such as simulation modeling, computer sciences, statistical sampling and, not least, cryptography. For this reason, finding a method of choosing that certain PRNG that meets all the specific qualities of that domain is a impetuous necessary task.

Most optimization problems that occur in practice have a multi-criteria character because they need to be simultaneously optimized for several criteria. Multi-criteria optimization problems play an important role in engineering, management and many other areas. These criteria that must need to be optimized, in most cases, can be in conflict. For example, at the same time some of the criteria need to be maximized and others minimized. Finding solutions to compromise on a rational basis has been a challenge for researchers over time.

In this study, we are trying to present a large part of the classification criteria and the characteristics that a PRNG needs to have in various areas. After identifying the main features and identifying the importance of each, we build a multi-criteria optimization problem in which the objective function is defined in relation to these properties and their weights.

By creating this optimization problem, a selection model is available from the existing PRNGs on the market to that generator that exactly meets the requirements of a particular customer. Moreover, can provide a model for the design and realization of certain types of generators.

The structure of the article is the following. Section 2 analyzes the need for pseudo-random number generation and selected qualities that a PRNG needs to meet in different areas. In Section 3, starting from these features that a PRNG has to accomplish, we have built up a multi-criteria optimization problem and we have developed a method of determining the weights that the objective functions must have in order to transform the multi-object optimization problem, in a single-object optimization problem.

2. PRNGs in different sciences

Probability theory and mathematical statistics are grounded around the abstract concept of random variable.

Although both mathematical domains provide us powerful tools to analyze the properties of random variables, their implementation in computing systems seems unrealistic because of the finite representation of numerical values. Computationally, the purely arbitrary behavior of real-world stochastic processes is simulated using deterministic algorithms, referred to in the literature as pseudorandom number generators.

Pseudorandom number generators are applicable in different areas such as Monte Carlo applications, simulation methods, statistical applications and cryptographic applications (session key generation, authentication protocols, digital signatures, etc.).

a. Thermodynamic systems as a reference model for PRNGs

The quality in terms of the randomness level of PRNGs needs to be compared to the level of randomization obtained in the case of random sequence generators. A source of random number sequences is represented by natural phenomena. A category of phenomena that can be analyzed for this purpose is the one based on the study of thermodynamic systems, especially those in the state of gas aggregation. The analysis of the behavior of the gas molecules regarding the spatial reference

system of the enclosure in which the gas it is located, can be a source of experiments from which random number sequences can be generated. The level of randomization in this case can certainly be appreciated as a level of refinement, randomization in this case being pure, natural. In assessing the quality of randomness of PRNGs, it is absolutely necessary that the reference system is certified, so that the evaluation is coherent and correct.

b. PRNGs in simulation modeling

In simulation modeling, one of the following situations might occur:
-The system that must be studied is too expensive to create in real life.
- It is not possible to conduct experiments directly on the system.
- A chance plays a part in the data.
- The system that we need to test does not exist yet.

In all these cases, a model can be simulated and tested for the impact of data change on system behavior. To run a simulation algorithm, it is necessary to use a random number generator.

Using a deterministic algorithm, the computer does not really generate a random number but a list of pseudo-random numbers.

Numerous generators of pseudo-random numbers have been created over time, but we cannot say about one of them, that it is the best.

Most programming languages have built-in random number generators, such as the command rand (1) that generates random numbers between 0 and 1, in MathLab. Also, the function round(x) returns 0 of $x \leq 1/2$ and returns 1 if $x > 1/2$.

Another probability distribution commonly used is the normal distribution. To get values of the normal distribution with a mean of μ and standard deviation of σ^2, is usually used:

$$\sigma(-2\ln(rand(1)))^{1/2}\cos(2\pi rand(1)) + \mu .$$

By far, the most used method of simulation is the Monte Carlo method. In this method are used a large number of random numbers to generate a model. By this method even the most complex systems can be described, but this, also takes a lot of time and memory consumption.

In practice, the objectives for simulation model very often may include the following:

1. Collecting statistics on the long-term behavior of the system.
2. Comparing alternative arrangements of the system, investigating the effects of changing the parameters or the modeling assumptions.
3. Finding the optimal operating conditions for the system.
4. Monitoring how the initial conditions influence the running time. Sometimes certain choices of random numbers can create the 'artificial state' of the system which might halt the calculations before the system transitions into a 'busy state'.

Therefore, we can conclude that, as a pseudo-random number generator to be good in simulation modeling it must meet the following requirements:

- low memory consumption;
- high speed;
- high degree of randomness.

c. PRNGs in statistical sampling

Pseudo-random number generators (PRNGs) are central to the practice of Statistics. They are used to draw random samples, allocate patients to treatments, perform the bootstrap, calibrate

permutation tests, perform MCMC, approximate p-values, partition data into training and test sets, and countless other purposes.

To make a simple random sampling must drawing k objects from a group of n in such a way that all $\binom{n}{k}$ possible subsets are equally likely. In real life truly random sample are difficult to obtain.

For this reason in statistical sampling there are used:

1. Pseudorandom number generators (PRNGs) because these produce sequences of bits, and
2. Sampling algorithms. This algorithms map a sequence of pseudorandom numbers into a subset, and that set is the set of population.

Most of the researchers in the field state that the simple sampling conditions are met through this procedure. However, there are some situations where PRNGs must generate the majority of possible cases, such as games of chance and lottery tickets. In such cases it is desirable to use it the multiple-seed PRNGs [2]. If, using the PRNG there are not generated all possible situations, then for that problem certainly there cannot be applied methods such: bootstrap samples, permutation, Monte Carlo integration. Selecting a simple sample is not that simple just for the reason that some are trying to cheat. Furthermore, randomness is not the only selection criteria for the sample. Recent works are encouraging the use of dices in choosing of the samples and discouraging the use of computers. In [3], for example, it is proposed not to use PRNGs at all, including CSPRNGs in the audit process. With all this skepticism, in [4] it is demonstrated that proper use of CSPRNGs would enhance audit security.

So, we conclude that, in statistical sampling, a PRNG must satisfy the following conditions:

- high degree of randomness;
- low memory consumption;
- long period;
- high speed.

d. PRNGs in cryptography

By far, cryptography has the most need of pseudo-random numbers. Cryptography requires numbers that attackers can't guess. For this reason, most generators have been built for cryptographic purposes and many have analyzed their qualities from various points of view [5, 6].
In cryptography, randomness plays has a key role in multiple applications, like key generation, initialization vectors generation, hiding or masking values. For this reason, security of the cryptographic systems is based on the use of sources that generate evenly distributed bits and with perfect randomness.

The main security requirements of an CSPRNG are that they must pass all statistical randomness tests, and that they should resist serious attacks, even in the case that the attacker has access to their initial or running state.

Using a PRNG in cryptographic applications requires that the generated values have a random and unpredictable character. The randomness of the generated values is conditional upon obtaining, at its output, strings of arbitrary length numbers that do not contain repetitions. Since a PRNG uses a fixed amount of memory, after a certain number of iterations, it will return to a previous state, at which point the states will repeat in the same order in an infinite cycle. Therefore, generators of

aperiodic pseudo-aliquot numbers cannot be built, but their successful use in cryptographic applications implies that the rehearsal period should be as high as possible [7].

The unpredictability of the generated values is the computational impossibility to determine, based on the current state, either a previous state or a subsequent state. Since a PRNG is mainly used to generate secret keys, the degree of unpredictability of a generator indicates the degree of security of a cryptosystem. If a possible attacker determines correlations between the generated numbers, then the security of the cryptosystem can be compromised. In fact, a quite important part of cryptanalysis is based on the exploitation of the low degree of unpredictability of the generated values.

Besides the statistical properties, a good cryptographic PRNG must have high generation speed and high resistance against attacks. If the generator is used in session key generation, speed does not play a fundamental role, a longer period is preferable in this case, but in application like simulations, stream ciphers, speed is compulsory condition.

On the other hand, robustness against attacks it is a requirement that all cryptographic PRNGs must meet. A famous example of a random number generator that is not robust from cryptographic point of view is the PRNG used in SSL (Secure Socket Layer) protocol. This generator is weak primarily because of the inappropriate choice of random sources (current time or process ID). A 128-bit session key produced by this generator only contained at most 47 bits of entropy and, thus, could be broken within minutes [8]. Speed and robustness are often contradictory qualities. For example, LFSRs, which are easy to implement in hardware and software, and which can therefore be used in applications requiring a rapid generation of random number sequences, are not cryptographically secure. In this case, an adversary can determine the seed of a generator only by observing 2n consecutive output bits [9]. There are, of course, generators that have a high degree of cryptographic security but which are slower. However, there exist exceptions like AES, which combines speed with high robustness against attacks.

We conclude that the main features that a CSPRNG must have are:

- high degree of randomness;
- long period;
- low memory consumption;
- high speed;
- high robustness.

Of course, the above areas are not the only ones that require PRNGs. The generation of pseudo-random numbers is an important and common task in computer programming. An area in which systems capable of generating random numbers are successfully used is that of the casinos. Nowadays, whether it's classic or virtual mechanical devices, random number generation systems are found in all of these, being essential for such games. Another area that uses such a system is that of the lottery. Random numbers generators are used in this area to establish winning numbers without involving any kind of human intervention, resulting in greater unpredictability.

Certain graphic artists or designers use this type of system to generate completely randomized images. With the help of specialized drawing and graphics applications, they can create impressive compositions that at first glance seem to be inconsistent with any rule. Often the systems involved can be extensively configured using dots, lines, regular or irregular geometric shapes, or any combination of them. Weaker forms of randomness are used in hash algorithms and in creating amortized searching and sorting algorithms [10, 11].

As a general conclusion, we can say that the main criteria for selecting PRNGs are as follows: large period, randomness, high generating speed, easy hardware or software implementation, and low memory consumption.

3. Building the optimization problem

Like any multi-criteria optimization problem, the problem formulation actually involves constructing the objective function. This function involves the characteristics that must be optimized.
In the previous section we have identified the criteria for selecting a particular PRNG from a list of PRNGs. Let's note these five criteria with x_1, x_2, x_3, x_4 *and* x_5.

Criteria may be contradictory. For example, if the period, randomness and generation speed are maximized, implementation costs and memory consumption should be minimized (in our case, must to be minimize criterion x_4 and x_5). For this reason, there is no single solution to optimize them all. We say that there are several Pareto optimal solutions and we need to find that solution called Pareto Optimal or Pareto Efficient, for which none of the objective functions is improved at the expense of another[12]. Multi-criteria optimization problems are studied from different points of view, therefore, there are different goals and solutions in solving and setting them. In some of these studies it is even introduced a set of pseudo-operators with the purpose of eliminating some of drawbacks which appear in mathematical optimization, like the lack of differentiability and convexity of the objective function [13,14].
To solve these types of problems there are various methods of finding the optimal solution, from the aggregation method or the displacements to a target value to evolutionary algorithms.

Problem formulation:

We consider the optimization problem

$$\max \quad f(x)$$

where $x = (x_1, x_2, x_3, x_4, x_5)$ is the set of characteristics of all available PRNGs, $f : \mathbb{R}^5 \to \mathbb{R}$.
The objective function is build such as:

$$f(x) = \prod_{i=1}^{3} w_i x_i \cdot \prod_{i=4}^{5} \frac{w_i}{x_i}, \; w_i \in (0,1), \; \sum_{i=1}^{5} w_i = 1.$$

To determine the parameters w_i, $i = 1,...,5$ we propose the following procedure:

1. Depending on the number of domains in which the PRNG is going to be used, a relevant number of experts can be selected, whose opinion is conclusive in the PRNG quality analysis study, without the aim of a statistical analysis, the number of domains of interest being finite.
In this paper we propose an analysis model based on the selection of seven experts, although there can be more or fewer. Let's note the seven experts with E_i, with $i = 1,...,7$.
2. Queries of the experts on the weight that each criterion f_i, with $i = 1,...,5$ should have. Let's note these weights with p_{ij}, with $i = 1,...,5$ and $j = 1,...,7$. We specify that these values p_{ij} are subunit values with $\sum_{j=1}^{7} p_{ij} = 1$ and with i = 1, ..., 5.
3. Of the values p_{ij}, with $i = 1,...,5$ the lowest and highest value are removed, with the purpose of obtaining an average value by dropping extreme values. Consider that the remaining values are p_{ij}, cu $i = 1,...,5$ and $j = 1,...,5$.
4. Add the remaining weights. We get the following values:

$$v_1 = \sum_{i=1}^{5} p_{i1}, \; v_2 = \sum_{i=1}^{5} p_{i2}, \; v_3 = \sum_{i=1}^{5} p_{i3}, \; v_4 = \sum_{i=1}^{5} p_{i4}, \; v_5 = \sum_{i=1}^{5} p_{i5} .$$

5. We sum up the five values $v_i \; i=1,...,5$ and we divide every value v_i at $\sum_{i=1}^{5} v_i$. Thus obtaining the values $w_i = \frac{v_i}{\sum_{i=1}^{5} v_i}$. These values have to be the necessary parameters in our optimization problem.

Conclusion:

In this article we have developed a multi-criteria evaluation procedure of pseudo-number generators to weigh them in relation to others and a method of determining the weight of each criterion. The selection of the experts is extremely important. We suggest choosing them from various fields of activity which involve the use of these generators, for example, from industry, research, education, commerce or users area. By this method, there can be analyzed any kind of technical device that needs to be weighed according to several parameters.

We did not intend to evaluate a particular case, this being a matter of detail that requires knowledge of each field. A complete analysis of a set of generators using the proposed methodology may be the subject of a future article.

As well as future research directions, we try to find ways to optimize the characteristics of pseudo-random number generators by classifying them according to their genre. Classifying it in: arithmetic generators, algebraic generators, hardware generators, generators obtained from simulations of various physical phenomena, we can get the optimum from its genre.

The domain of thermodynamics might have a hard word to say in terms of generating purely random numbers. It is well known that in the case of thermodynamic processes such as the combustion process, a relationship must always be fulfilled: $V \cdot P \cdot Temp = ct.$ When one of these parameters varies, the internal energy inside the gas container changes. By modifying internal energy, the behavior of gas molecules becomes chaotic. In this way, by studying the behavior of one or more gas molecules, at variations of temperature or pressure, or both, we could generate purely random numbers. These numbers could be given, for example, by the distance from the molecule to the container walls at different and discreet moments of time.

References

[1] L'ecuyer P 2008 Comparison of point sets and sequences for quasi-montecarlo and for random number generation *SETA 2008* vol. LNCS **5203** p 1–17

[2] Marsaglia G 2003 Seeds for Random Number Generators Commun, *ACM*, **46(5)** p 90–93

[3] Jefferson D, Ginnold E, Midstokke K, Alexander K, Stark P and Lehmkuhl A July 2007 *Evaluation of audit sampling models and options for strengthening California's manual count* Post-election audit standards working group report

[4] Joseph A, Calandrino J, Halderman A and Edward W Felten July 28-29 2008 *In Defense of Pseudorandom Sample Selection* USENIX/ACCURATE Electronic Voting Workshop San Jose CA USA

[5] Gaeini A, Mirghadri A and Jandaghi G 2015 A General Evaluation Pattern for Pseudo Random Number Generators *Trends in Applied Sciences Research*, vol. **10(5)** p 231-244

[6] Knezevic K 2017 *Combinatorial Optimization in Cryptography* 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)

[7] Marsaglia G 1994 Some Portable Very-long Period Random Number Generators *Computers in Physics* vol. **8** p 117-121

[8] Goldberg I Wagner D January 1996 Randomness and the netscape browser *Dr. Dobbs Journal*

[9] Schneier B 1993 *Applied Cryptography* (Wiley New York NY USA)

[10] Rogobete M Răcuciu C 2016 Watermarking Protection for 3D Images *"Mircea cel Batran" Naval Academy Scientific Bulletin* Volume **XIX** – Issue 1, DOI: 10.21279/1454-864X-16-I1-083, ISSN: 2392-8956.

[11] Rogobete M Tărăbuţă O Rogobete AD and Eftimie S 2018 Using Gravity Potential Field and Inertial Navigation System in Real Time Submarine Positioning, *IOP Conference Series: Earth and Environmental Science*, Volume **172**, conference 1, DOI: 10.1088/1755-1315/172/1/012005, ISSN 1755-1315.

[12] Ehrgott M 2005 *Multicriteria Optimization*(second edition, Springer Berlin Heidelberg New York).

[13] Cornaciu V Ileana I 2017 The Avriel-Ben-Tal algebraic operations approach for a short version proof of the Karush-Kuhn-Tucker optimality conditions *Ovidius University annals, Mathematical Series*, **25(2).**

[14] Cornaciu V 2016 $(h,\varphi)_{\varepsilon}-$Optimality conditions for multi-objective fractional semi-infinite programming with $K-(F_b,\rho)-$uniform convexity *"Mircea cel Batran"Naval Academy Scientific Bulletin*, **19(1)**, 359-366.