Ссылка на статью:
// Ученые записки УлГУ. Сер. Математика и информационные технологии. 2023, № 2, с. 1-10.

Поступила: 01.12.2023
Окончательный вариант: 18.12.2023

УДК 004.415.53, 004.421.5

Тестирование генераторов псевдослучайных чисел

Вдовина О.Н.

ovdovina33@gmail.com
УлГУ, Ульяновск, Россия

В статье изложены возможные методы тестирования генераторов псевдослучайных чисел. Рассмотрен вопрос истиной случайности генерируемых последовательностей. Проведены обзор и анализ существующих в настоящее время генераторов псевдослучайных чисел (ГСПЧ) и выявлены причины необходимости в их тестировании. Проведено собственное тестирование популярных ГПСЧ. Рассмотрено применение таких программ, как rngtest, и их эффективность при тестировании алгоритмов генерации псевдослучайных последовательностей.

***Ключевые слова**: генератор псевдослучайных чисел, статические и графические тесты, набор статических тестов NIST, тестирование генераторов псевдослучайных чисел*

Введение

Случайные последовательности являются полезным инструментом компьютерного моделирования [1]. Например, в имитационном моделировании, которое помогает в решении задачи оптимизации производственных процессов, исследует режимы загрузки оборудования, проводит анализ финансово-экономического состояния предприятия, прогнозирует результаты определенных решений и т. д. Сходство с реальными явлениями моделям придают псевдослучайные числа. Для реализации случайных факторов используются датчики базовой случайной величины. Псевдослучайные числа являются хорошим источником данных для проверки различных тестов, поэтому находят применение в криптографии, компьютерной графике, в тестировании компьютерных алгоритмов.

Псевдослучайная последовательность – это последовательность чисел, вычисляемая с помощью некоторого математического алгоритма и не имеющая, на первый взгляд, закономерностей. Такие последовательности генерируются псевдослучайными генераторами [2].

Качественный генератор псевдослучайных чисел (ГПСЧ) должен удовлетворять следующим условиям [3]:

- числа в генерируемой последовательности должны быть равномерно распределены и независимы;
- период последовательности должен иметь возможно большую длину.

Также современный генератор псевдослучайных чисел должен удовлетворять следующим требованиям:

- эффективность;
- невозможность повторной генерации;
- мультиплатформенность;
- простота программной реализации.

Для определения качества генерируемой последовательности существуют различные тесты, позволяющие выявить преимущества и недостатки алгоритма. Тестирование является необходимым этапом при создании генератора псевдослучайных чисел.

Цель данной работы заключается в исследовании методов тестирования генераторов псевдослучайных чисел.

1. Статистические и графические тесты

Тестирование псевдослучайных последовательностей бывает статистическим или визуальным. В тестах первого вида статистические свойства последовательности определяются числовыми характеристиками. В данном тестировании оценочные критерии, полученные при исследовании свойств анализируемой и истинно случайной последовательностей, служат фундаментом, на котором строится заключение о прохождении теста.

Визуальное тестирование проще, в нем свойства генерируемой последовательности отображаются графически, что позволяет визуально отследить поведение исследуемой последовательности.

Однако, интерпретируемые пользователем результаты предполагают возможные различные трактовки этих результатов, что делает визуальное тестирование не самым точным. Статистические тесты в свою очередь выдают численную характеристику, что однозначно дает понять, пройден тест или нет [4].

2. Тестирование псевдослучайных генераторов чисел

Практическая оценка случайности опирается на эмпирические тесты, оценивающие гипотезы или теории посредством обращения к фактам и данным, полученным опытным путем. Большинство эмпирических тестов основаны на проверке статистических гипотез. Каждый такой тест исследует случайность определенным образом, проверяет отдельные статистические характеристики. Статистические тесты группируются в наборы для лучшего анализа случайностей [4].

Такие наборы должны быть достаточно большими для получения надежных результатов. Тесты могут быть реализованы для разных целей. Например, в работе [5] были рассмотрены некоторые виды генераторов псевдослучайных чисел, один из них – линейно

конгруэнтный генератор, разработанный Д.Х. Лехмером в 1949 г. и Blum BlumShub, предложенный Л. Блюм, М. Блюм и М. Шубом в 1986 г. Благодаря проведенным тестам, а именно: тест на самую длинную серию единиц в блоке, универсальный статистический тест Маурера, тест совокупных сумм, тест дискретного преобразования Фурье и т.д., удалось проанализировать данные алгоритмы генерации псевдослучайных чисел, что позволило прийти к выводу об их простоте и легкости реализации, а также удалось подтвердить случайность и непредсказуемость сгенерированных последовательностей.

Существует ряд пакетов статистических тестов для проверки случайности. Сборник тестов случайности Кнута, включающий в себя 11 статистических тестов, CRYPT-X включает 8 тестов, DIEHARD включает 15 тестов, DIEHARDER включает 26 статистических тестов, TESTU01 включает 16 тестов, набор тестов NIST изначально включает в себя 16 тестов.

Работа [4] содержит описание некоторых из них:

- Тесты Д. Кнута, предложенные им в 1969 г., основанные на статистическом критерии, сравнивают вычисляемое значение с табличными результатами, и в зависимости от вероятности появления такой статистики делается вывод о ее качестве. Достоинство этих тестов – небольшое количество и быстрый алгоритм выполнения. Недостаток – неопределенность в трактовке результатов.
- Тесты DIEHARD – набор статистических тестов для измерения качества набора случайных чисел. Они были разработаны Д. Марсалья [6] и считаются одними из наиболее строгих наборов тестов.
- Набор статистических тестов CRYPT-X является коммерческим пакетом программного обеспечения. Они применяются в зависимости от типа алгоритма генератора. В данный набор входят такие тесты, как частотный, на последовательность одинаковых битов, на линейную сложность, на сложность последовательности и т.д.
- В [4] был разобран новый тест «Стопка книг», разработанный Б.Я. Рябко и А.И. Пестуновым. Результаты, полученные с помощью этого теста, показали его эффективность в сравнении с тестами NIST STS. Пакет статистических тестов NIST STS используется в основном для задач криптологии. Набор тестов применяются при исследовании битовых последовательностей. Тесты считаются в достаточной мере независимыми, а пакет NIST STS является эффективным по затратам машинного времени и строгим в оценке свойств ГПСЧ [7].

3. Качество генератора псевдослучайных чисел

Есть два вида генераторов случайных чисел: генераторы псевдослучайных чисел, рассмотренные выше, стремящиеся математически вывести последовательность, «похожую» на истинно случайную, и физические, генерирующие действительно случайные последовательности. Последние используют в качестве источника энтропии (меры хаоса в какой-либо системе) измерения физических величин, что делает последовательность не только

равномерной, но и непредсказуемой. Однако такие генераторы очень трудоемки в реализации за счет высокой стоимости и сложности получения физических измерений и заметно уступают в скорости и производительности псевдослучайным генераторам чисел.

Тестирование алгоритмов проверяет выполнение необходимых свойств, характерных для чисто случайной последовательности чисел. Но является ли тестирование гарантией качества генератора? В [8] рассмотрен пакет с набором тестом NIST STS. Набор состоит из 15 эмпирических тестов, разработанных для анализа двоичных последовательностей. Посредством тестирования проверяется случайность данных в соответствии с различной статистикой битов или статистикой блоков битов. Появление пройденной теста последовательности равна дополнению уровня значимости 1–a. В исследовании утверждается, что доля проходящих последовательностей должна попадать в определенный интервал около (1–a) с большой вероятностью. NIST рассчитывает интервал допустимых пропорций (равенство двух соотношений) с помощью выражения

$$(1-a) \pm 3 \frac{\sqrt{a(1-a)}}{k}.$$

В [7] данная формула описана подробнее, рассмотрено ее применение в тестировании криптостойких генераторов псевдослучайных чисел.

Вернемся к статье [8], в которой исследуют вопрос ненадежности известных генераторов псевдослучайных чисел. Чтобы оценить случайность данных, была измерена вероятность того, что случайная последовательность окажется неудачной. Вычисленные вероятности используются для вычисления значения p для всего набора теста. Это значение является вероятностью того, что идеальный генератор случайных чисел будет генерировать последовательности с результатами хуже, чем результаты, вычисленные для данной последовательности.

Авторы продемонстрировали важность интерпретации результатов тестирования. NIST предлагает считать данные случайными, если абсолютно все тесты пройдены, однако ими было доказано, что по-настоящему случайные данные показывают высокую вероятность, а именно, 80%, провалить хотя бы один тест NIST STS. Рекомендуется провести анализ дополнительных образцов, если данные не соответствуют некоторым тестам NIST.

В [9] авторы провели тестирование тринадцати известных генераторов. Ими был использован простой визуальный критерий для оценки случайности чисел последовательности. Была рассмотрена зависимость от магических чисел N, то есть промежуточных значений, с помощью которых вычисляется последовательность Соболя [10], и проведенные расчеты полученных результатов показали, что даже при очень больших N неравномерность заполнения остается значительной. Такие выводы свидетельствуют о не слишком высоком качестве генераторов.

4. Тестирование генератора

Не всегда к генератору выдвигаются серьезные требования, иногда необходимо, чтобы генератор псевдослучайных чисел выдавал более или менее случайную последова-

тельность. Например, такую, чтобы генерировать случайные объекты в игре. Для проверки таких последовательностей достаточно использовать тесты попроще. Один из таких тестов Rngtest, который проверяет случайность данных с помощью набора тестов FIPS 140-2. Для проведения тестирования в данной работе был использован бесплатный хостинг-провайдер OnWorks [11], где посредством команды rngtest были протестированы четыре генератора псевдослучайных чисел. Генератор проходит тестирование, если на выходе выдается статус 0.

Приведенный ниже код линейно-конгруэнтного генератора псевдослучайных чисел определен посредством rngtest как генерирующий хорошие псевдослучайные последовательности.

```
#include <iostream>
using namespace std;
double congruent(int&);
int main(int argc, char* argv[])
{
    const int num = 25;
    int x0 = 2;
    cout << "\random numbers:   ";
    for (int i = 0; i <= num; i++)
        cout << congruent(x0) << " ";
    cout << "\n";
    return 0;
}
double congruent(int& x) {
    const int m = 100,
        a = 8,
        inc = 65;
    x = ((a * x) + inc) % m;
    return (x / double(m));
}
```

Об этом свидетельствуют представленные в Таблице 1.1 результаты.

Таблица 1.1 – Результаты тестирования линейно-конгруэнтного генератора

Тесты	Значение	Скорость		
		Min	avg	max
bits received from input:	-	-	-	-
FIPS 140-2 successes	0	-	-	-
FIPS 140-2 failures	0	-	-	-

FIPS 140-2(2001-10-10) Monobit	0	-	-	-
140-2(2001-10-10) Poker	0	-	-	-
FIPS 140-2(2001-10-10) Runs	0	-	-	-
FIPS 140-2(2001-10-10) Long run	0	-	-	-
FIPS 140-2(2001-10-10) Continuous run	0	-	-	-
input channel speed	-	5.201(Gibits/s)	5.201(Gibits/s)	5.201(Gibits/s)
FIPS tests speed	-	204.427(Mibits/s)	204.427(Mibits/s)	204.427(Mibits/s)
Program run time	5338 microseconds	-	-	-

Прочерки в ячейках таблицы означают, что при тестировании соответствующие параметры не рассчитывались.

Нули, записанные в колонку «Значения», указывают на прохождение данным генератором тестов rngtest.

В Таблице 2.1 приведены результаты тестирования другого генератора псевдослучайных чисел, а именно вихрь Мексона [12].

Таблица 2.1 – Результаты тестирования генератора вихрь Мексона

Тесты	Значение	Скорость		
		min	avg	max
bits received from input:	-	-	-	-
FIPS 140-2 successes	0	-	-	-
FIPS 140-2 failures	0	-	-	-
FIPS 140-2(2001-10-10) Monobit	0	-	-	-
140-2(2001-10-10)	0	-	-	-

Poker				
FIPS 140-2(2001-10-10) Runs	0	-	-	-
FIPS 140-2(2001-10-10) Long run	0	-	-	-
FIPS 140-2(2001-10-10) Continuous run	0	-	-	-
input channel speed	-	7.321(Gibits/s)	7.321(Gibits/s)	7.321(Gibits/s)
FIPS tests speed	-	237.425(Mibits/s)	237.425(Mibits/s)	237.425(Mibits/s)
Program run time	4763 microseconds	-	-	-

По результатам тестирования можно сделать вывод о том, что выбранный генератор псевдослучайных чисел показал отличный результат. Также были проведены тесты для таких генераторов случайных чисел, как стандартный генератор rand C++ и генератор Фибоначчи с запаздыванием.

Для сравнения был протестирован “плохой” генератор псевдослучайных чисел, код которого представлен ниже:

```
#include <iostream>
class BadRandomNumberGenerator {
private:
    unsigned int seed;
public:
    BadRandomNumberGenerator(unsigned int s) : seed(s) {}
    int next() {
        seed = (seed + 1234567) % 100;
        return seed;
    }
};
int main() {
    BadRandomNumberGenerator rng(0);
    for (int i = 0; i < 10; i++) {
        std::cout << rng.next() << std::endl;
    }
    return 0;
}
```

Данный генератор не является криптографически стойким или особо хорошим генератором чисел по ряду причин, таких как:

- предсказуемые изменения, так как просто добавление фиксированной константы к начальному значению означает, что разница между последовательными “случайны-

ми” числами всегда одинакова, а значит, создается очень предсказуемая последовательность.

- нет изменений при вводе одного и того же начального значения, это в свою очередь позволяет предсказать все будущие выходные данные.
- последовательность не случайна по любым хорошим меркам. Она не проходит статические тесты на случайность, о чем свидетельствуют результаты, представленные в Таблице 3.1.

Таблица 3.1 – Результаты тестирования не криптографически стойкого генератора

Тесты	Значение	Скорость		
		min	avg	max
bits received from input:	-	-	-	-
FIPS 140-2 successes	0	-	-	-
FIPS 140-2 failures	0	-	-	-
FIPS 140-2(2001-10-10) Monobit	1	-	-	-
140-2(2001-10-10) Poker	1	-	-	-
FIPS 140-2(2001-10-10) Runs	1	-	-	-
FIPS 140-2(2001-10-10) Long run	1	-	-	-
FIPS 140-2(2001-10-10) Continuous run	0	-	-	-
input channel speed	-	3.825 ms(Gibits/s)	3.825ms(Gibits/s)	3.825 ms(Gibits/s)
FIPS tests speed	-	192.025(Mibits/s)	192.025(Mibits/s)	192.025(Mibits/s)
Program run time	1.1562 microseconds	-	-	-

Rngtest обнаружил, что данный генератор потерпит неудачу в монобитном тесте (об этом свидетельствует единица в соответствующей колонке), старшие биты будут равны нулю, искажая баланс. Так же Гпсч не пройдет тест Poker и приведет к сбою теста Runs. Long run – еще один гарантированный сбой, который произойдет из-за низкого периода.

Такой инструмент, как rngtest, практически всегда указывает на то, что проверяемая последовательность проходит тест на случайность. Тогда возникает ожидаемый вопрос – для чего нужны такие тесты? Это необходимо для того, чтобы быть уверенным, что разрабатываемый алгоритм, генерирующий псевдослучайные последовательности, не оказался слишком слабым как, например, в приведенном выше примере “плохого” генератора. Встречаются генераторы псевдослучайных чисел, у которых существуют недостатки подобия малого диапазона генерации, частые повторения одних и тех же блоков и т. д. Rngtest обратит на это внимание, что и делает его полезным инструментом для случаев, когда генерируемые последовательности должны быть более или менее случайными.

Чтобы правильно протестировать генератор псевдослучайных чисел, необходимо ясно представлять, для каких целей будут использоваться генерируемые последовательности, и насколько случайными они должны быть. Поэтому оценка качества генераторов псевдослучайных чисел опирается на область их использования.

Заключение

Проблема объективной оценки качества алгоритмов генерации псевдослучайных последовательностей все еще остается актуальным вопросом. Нет общепринятого метода оценивания близости псевдослучайных последовательностей к истинно случайным. Также не существует такого алгоритма построения псевдослучайных чисел, который бы мог использоваться для расчета любых случайных процессов, ведь количественные оценки качества генераторов не охватывают всех факторов, необходимых для решения конкретной задачи. Однако тестирование является важнейшим этапом при создании и выборе генераторов псевдослучайных чисел. Оно выявляет сильные и слабые стороны генерируемых последовательностей, ориентирует пользователя в оценке качества определенных характеристик генератора и позволяет по максимуму использовать тестируемые алгоритмы.

По результатам проведенного исследования можно сделать вывод о том, что при работе с псевдослучайными последовательностями проведение тестирования является необходимым. Насколько строгими должны быть тесты, зависит от области использования генерируемых генератором чисел. Для проверки ГПСЧ, генерирующего, например, случайные имена пользователя, достаточно будет проверки такой программой, как rngtest. В случае же, когда случайные числа используются в области криптологии, для проверки псевдослучайных последовательностей необходимы такие наборы тестов, как Тесты DIEHARD, NIST и т. д. Выбор метода тестирования ГПСЧ всегда зависит от цели использования генерируемых последовательностей.

Список литературы

1. Кнут Д.Э. *Искусство программирования. Т.2. Получисленные алгоритмы* / М.: Изд. дом «Вильяме», 2000.
2. Харин Ю.С. *Математические и компьютерные основы криптологии*: учеб. пособие / Минск: Новое знание, 1999.

3. Колесова Н.А. Оценка качества генераторов последовательностей случайных чисел // *Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика*. 2011, №1, с. 119–123.
4. Григорьев А.Ю. Методы тестирование генераторов случайных и псевдослучайных последовательностей // *Ученые записки УлГУ. Сер. математика и информационные технологии.* 2017, № 1, с. 22–28.
5. Priyanka, Hussain I, Khalique A. Random Number Generators and their Applications: A Review // *International Journal of Research in Electronics and Computer Engineering*. 2019, v. 7, p. 1777–1781.
6. Вильданов Р.Р., Мещеряков Р.В., Бондарчук С.С. Тесты псевдослучайных последовательностей и реализующее их программное средство // *Доклады ТУСУРа*. 2012, № 1 (25), часть 2, с. 108–111.
7. Будько М.Б., Будько М.Ю., Гирик А.В., Грозов В.А. Метод оценки качества криптостойких генераторов псевдослучайных последовательностей // Вопросы кибербезопасности. 2018, № 4 (28), с. 29–37.
8. Marek Sys, Vashek Matyas On the interpretation of from the NIST statistical test suite // Romanian Information Journal of Science and Technology. 2015, v/ 18, p. 18-32
9. Белов А.А., Калиткин Н.Н., Тинтул М.А. Ненадежность известных генераторов псевдослучайных чисел // *Журнал вычислительной математики и математической физики*. 2020. Т. 60, с. 807–814.
10. Соболь И.М. О распределении точек в кубе и сетках интегрирования // *Успехи матем. наук*. 1966. Т. 21, № 5. С. 271–272.
11. *Хостинг-провайдер OnWorks, реализующий выполнение программы Rngtest*. Режим доступа: https://www.onworks.net/ru/programs/rngtest-online (дата обращения 25.11.2023).
12. Git репозиторий с программной реализацией стороннего генератора псевдослучайных чисел, опубликованный Erik Garrison. Режим доступа: https://github.com/root-https://github.com/ekg/mutatrix/blob/master/mt19937ar.h (дата обращения 26.11.2023).

Testing pseudorandom number generators

Vdovina O.N.

ovdovina33@gmail.com

Ulyanovsk State University, Russia

The paper describes possible methods for testing pseudorandom number generators. The question of true randomness of generated sequences is considered. A review and analysis of currently available pseudorandom number generators (PRNGs) is performed, and the reasons for the need to test them are revealed. The authors have performed their own tests on popular PRNGs. The use of programs such as rngtest and their effectiveness in testing pseudorandom sequence generation algorithms are considered.

Keywords*: pseudorandom number generator, static and graphical tests, NIST static test suite, testing of pseudorandom number generators*