Федеральное агентство связи (Россвязь)
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра ВС

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к нулевой лабораторной работе
«Оценка качества генератора псевдослучайных чисел»
по дисциплине «Моделирование»

Выполнил:
студент гр. ИВ-621 ______________________________ /Д.А. Гурулев/
подпись

Проверил:
ассистент кафедры ВС
______________________________ /Я.В. Петухова /
ОЦЕНКА, подпись

Новосибирск, 2020

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ

В рамках лабораторной работы необходимо выбрать три генератора псевдослучайных чисел, оценить равномерность последовательностей псевдослучайных чисел на заданном отрезке с помощью критерий согласия Пирсона, оценить статистическую независимость генерируемых последовательностей с помощью автокорреляционной функции.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Критерий согласия Пирсона (χ^2)

Критерий проверки гипотезы о предполагаемом законе неизвестного распределения называется критерием согласия.

Для проверки гипотезы поступают следующим образом. Область значений с. в. X разбивается на k интервалов δ_i и подсчитывают вероятности p_i ($i = 1, 2, \ldots, k$) попадания в δ_i интервал, используя формулу $P\{\alpha \le X \le \beta\} = F_0(\beta) - F_0(\alpha)$. Тогда число значений с. в. X, попавших δ_i интервал, можно рассчитать по формуле $N * p_i$.

Таким образом, имеем статистический ряд распределения с. в. X и теоретический ряд распределения. Чтобы оценить степень расхождения эмпирических частот n_i и теоретических частот $\grave{n_i}$ К. Пирсон предложил такую величину:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - N * p_i)^2}{N * p_i} = \frac{1}{N} \sum_{i=1}^{k} (\frac{{n_i}^2}{p_i}) - N$$

Доказано, что при $n \to \infty$ закон распределения случайной величины независимо от того, по какому закону распределена генеральная совокупность, стремится к закону χ^2 (хи-квадрат) с k степенями свободы.

Число степеней свободы находят по равенству $k = S - 1 - r$, где S − число групп интервалов, r − число параметров.

1) равномерное распределение: r=2, k=S−3

2) нормальное распределение: r=2, k=S−3

3) показательное распределение: r=1, k=S−2.

Проверка гипотезы по критерию Пирсона:

1. Для проверки гипотезы вычисляют теоретические частоты и находят $\chi^2_{набл}$.

2. По таблице критических точек распределения χ^2 по заданному уровню значимости α и числу степеней свободы k находят $\chi^2_{кр}(\alpha, k)$.

3. Если $\chi^2_{набл}<\chi^2_{кр}$, то нет оснований отвергать гипотезу, если не выполняется данное условие - то отвергают.

Необходимым условием применения критерия Пирсона является наличие в каждом интервале не менее 5 наблюдений.

Автокорреляция

Корреляция, корреляционная зависимость - взаимозависимость двух или нескольких случайных величин. При расчете корреляций пытаются определить, существует ли статистически достоверная связь между двумя или несколькими переменными в одной или нескольких выборках.

Автокорреляция — это статистическая взаимосвязь между последовательностями величин одного ряда, взятыми со сдвигом. Определяется по формуле:

$$\hat{\alpha}(\tau) = \frac{1}{(N-\tau) * S^2} \sum_{i=1}^{N-\tau} (X_i - \bar{X}) * (X_{i+\tau} - \bar{X})$$

, где X_i – множество псевдослучайных чисел, S^2– выборочная дисперсия, N – кол-во чисел, τ –сдвиг последовательности.

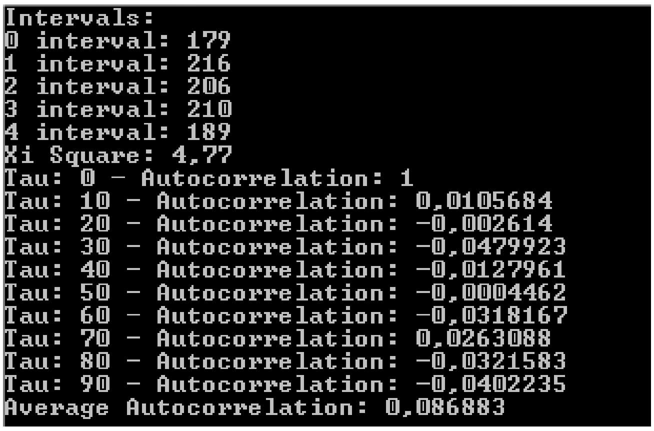
ГЕНЕРАТОРЫ

Генератор псевдослучайных чисел (ГСЧ) — алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению.

Для проектирования были выбраны следующие генераторы случайных чисел:

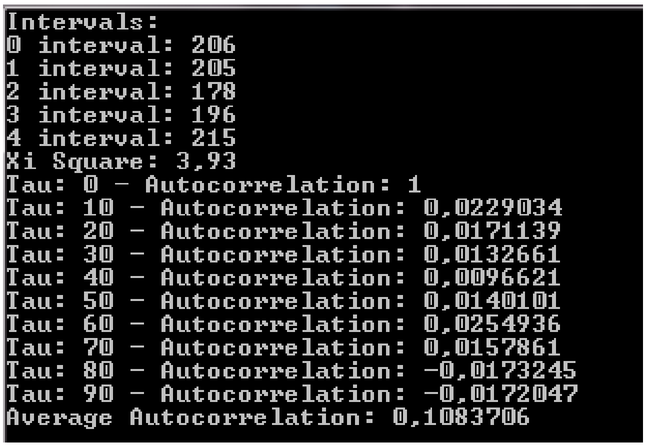
1. Измененная версия алгоритма генератора псевдослучайных чисел с вычитанием Дональда Е. Кнута, реализованная в .Net в виде класса Random.
2. Mersenne Twister — генератор псевдослучайных чисел, разработанный в 1997 году японскими учёными Макото Мацумото и Такудзи Нисимура. Вихрь Мерсенна основывается на свойствах простых чисел Мерсенна (отсюда название) и обеспечивает быструю генерацию высококачественных по критерию случайности псевдослучайных чисел. Псевдослучайная последовательность, порождаемая вихрем Мерсенна, имеет очень большой период, равный числу Мерсенна, что более чем достаточно для многих практических приложений.
3. Генератор случайных чисел Xorshift, также называемый генератором регистра сдвига, представляет собой класс генераторов псевдослучайных чисел, которые были обнаружены Джорджем Марсаглией. Он генерирует следующее число в своей последовательности путем многократного взятия исключительного числа или числа со сдвинутым битом. Это делает его чрезвычайно быстрым на современных компьютерных архитектурах.

РЕЗУЛЬТАТЫ ЭКСПЕРЕМНТОВ



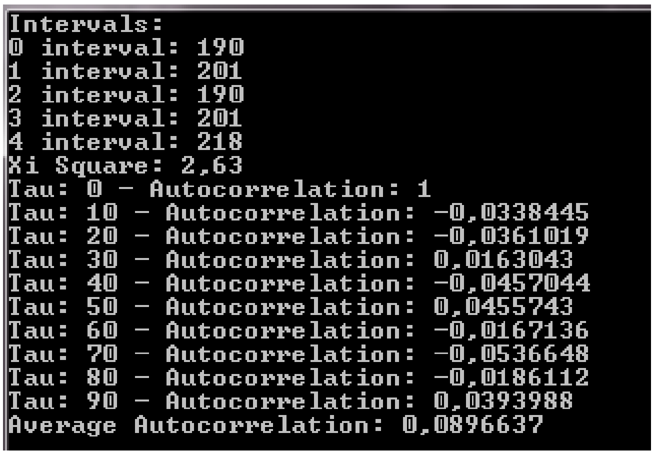
```
Intervals:
0 interval: 179
1 interval: 216
2 interval: 206
3 interval: 210
4 interval: 189
Xi Square: 4,77
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: 0,0105684
Tau: 20 - Autocorrelation: -0,002614
Tau: 30 - Autocorrelation: -0,0479923
Tau: 40 - Autocorrelation: -0,0127961
Tau: 50 - Autocorrelation: -0,0004462
Tau: 60 - Autocorrelation: -0,0318167
Tau: 70 - Autocorrelation: 0,0263088
Tau: 80 - Autocorrelation: -0,0321583
Tau: 90 - Autocorrelation: -0,0402235
Average Autocorrelation: 0,086883
```

Рисунок 1. Результат проверки ГСЧ с вычитанием Д.Кнута с параметрами N=1000, k=5



```
Intervals:
0 interval: 206
1 interval: 205
2 interval: 178
3 interval: 196
4 interval: 215
Xi Square: 3,93
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: 0,0229034
Tau: 20 - Autocorrelation: 0,0171139
Tau: 30 - Autocorrelation: 0,0132661
Tau: 40 - Autocorrelation: 0,0096621
Tau: 50 - Autocorrelation: 0,0140101
Tau: 60 - Autocorrelation: 0,0254936
Tau: 70 - Autocorrelation: 0,0157861
Tau: 80 - Autocorrelation: -0,0173245
Tau: 90 - Autocorrelation: -0,0172047
Average Autocorrelation: 0,1083706
```

Рисунок 2. Результат проверки ГСЧ Mersenne Twister с параметрами N=1000, k=5



```
Intervals:
0 interval: 190
1 interval: 201
2 interval: 190
3 interval: 201
4 interval: 218
Xi Square: 2,63
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: -0,0338445
Tau: 20 - Autocorrelation: -0,0361019
Tau: 30 - Autocorrelation: 0,0163043
Tau: 40 - Autocorrelation: -0,0457044
Tau: 50 - Autocorrelation: 0,0455743
Tau: 60 - Autocorrelation: -0,0167136
Tau: 70 - Autocorrelation: -0,0536648
Tau: 80 - Autocorrelation: -0,0186112
Tau: 90 - Autocorrelation: 0,0393988
Average Autocorrelation: 0,0896637
```

Рисунок 3. Результат проверки ГСЧ Xorshift с параметрами N=1000, k=5

```
Intervals:
0 interval: 1047
1 interval: 966
2 interval: 1016
3 interval: 1033
4 interval: 1010
5 interval: 973
6 interval: 999
7 interval: 964
8 interval: 984
9 interval: 1008
Xi Square: 7,156
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: -0,007668
Tau: 20 - Autocorrelation: -0,0079235
Tau: 30 - Autocorrelation: 0,0042541
Tau: 40 - Autocorrelation: -0,0052093
Tau: 50 - Autocorrelation: -0,0168563
Tau: 60 - Autocorrelation: 0,0007122
Tau: 70 - Autocorrelation: 0,001234
Tau: 80 - Autocorrelation: 0,0046653
Tau: 90 - Autocorrelation: 0,0042306
Average Autocorrelation: 0,0977439
```

Рисунок 4. Результат проверки ГСЧ с вычитанием Д.Кнута с параметрами N=10000, k=10

```
Intervals:
0 interval: 1011
1 interval: 966
2 interval: 979
3 interval: 1006
4 interval: 1001
5 interval: 995
6 interval: 972
7 interval: 1062
8 interval: 1008
9 interval: 1000
Xi Square: 6,472
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: 0,0178571
Tau: 20 - Autocorrelation: 0,0038234
Tau: 30 - Autocorrelation: -0,0032238
Tau: 40 - Autocorrelation: 0,0064979
Tau: 50 - Autocorrelation: 0,0204474
Tau: 60 - Autocorrelation: -0,0013592
Tau: 70 - Autocorrelation: -1,42E-05
Tau: 80 - Autocorrelation: -0,0089258
Tau: 90 - Autocorrelation: 0,0051369
Average Autocorrelation: 0,104024
```

Рисунок 5. Результат проверки ГСЧ Mersenne Twister с параметрами N=10000, k=10

```
Intervals:
0 interval: 1029
1 interval: 1033
2 interval: 996
3 interval: 970
4 interval: 999
5 interval: 1007
6 interval: 1005
7 interval: 1016
8 interval: 962
9 interval: 983
Xi Square: 4,91
Tau: 0 - Autocorrelation: 1
Tau: 10 - Autocorrelation: -0,013969
Tau: 20 - Autocorrelation: -0,0096379
Tau: 30 - Autocorrelation: 0,013953
Tau: 40 - Autocorrelation: -0,0396626
Tau: 50 - Autocorrelation: -0,0065652
Tau: 60 - Autocorrelation: 0,0024798
Tau: 70 - Autocorrelation: 0,0218315
Tau: 80 - Autocorrelation: -0,0056223
Tau: 90 - Autocorrelation: 0,0069643
Average Autocorrelation: 0,0969772
```

Рисунок 5. Результат проверки ГСЧ Xorshift с параметрами N=10000, k=10

ЗАКЛЮЧЕНИЕ

Для проведения экспериментов была написана программа на языке программирования C#. В ходе работы были изучены методы тестирования качества работы генератора псевдослучайных чисел: критерий согласия Пирсона (χ^2) и автокорреляция. С помощью данных методов тестирования были исследованы: измененная версия алгоритма генератора псевдослучайных чисел с вычитанием Д.Кнута, Mersenne Twister и Xorshift. Реализации используемых ГСЧ есть в свободно распространяемых пакетах для .Net Framework.

Для анализа критерия согласия Пирсона при уровне значимости α = 0,05 и числе степеней свободы k=7 по таблице выбрано значение квантиля $\chi^2_{\text{кр}} = \chi^2_{\alpha=0,05,\ k=7} = 14.1$. Из полученных результатов по критерию согласия Пирсона можно сделать следующие выводы: у генератора Xorshift среднее отклонение эмпирических частот от теоретических больше, чем у остальных генераторов, следовательно, данный ГСЧ выдал более равномерное распределение, чем другие. При увеличении количества интервалов и точек, значение χ^2 увеличивается у всех ГСЧ, но не превышает критического значения. Результаты подтверждают, что все выбранные генераторы случайных чисел генерируют последовательность чисел, подчиняющуюся равномерному распределению.

Из результатов проектирования можно судить о том, что автокорреляционная функция для всех трех ГСЧ колеблется около 0, принимая положительные и отрицательные значения близкие нулю. Значит статистическая взаимосвязь между исходной и сдвинутой последовательностью пренебрежимо мала и можно сделать вывод о том, что исследуемые генераторы выдают независимые случайные величины.

ПРИЛОЖЕНИЕ

Листинг программы

```
using System;
using System.Collections.Generic;
using CenterCLR.XorRandomGenerator;
using MersenneTwister;

namespace Modeling
{
    class Program
    {
        const int MAX = 10000;
        static double get_rand()
        {
            Random rnd = new Random();
            double value = rnd.NextDouble();
            return value;
        }
        static double get_mt()
        {
            double value = Randoms.NextDouble();
            return value;
        }
        static double get_xor()
        {
            var r = new XorRandom();
            double value = r.NextDouble();
            return value;
        }
        public void xi_sq(int N, double k)
        {
            List<double> numbers = new List<double>();
            List<int> intervals = new List<int>();
            numbers.Add(0.0);
            double delta = 1.0 / k;
            double n = delta;
            for (double i = 0; i < k; i++)
            {
                numbers.Add(n);
                n += delta;
            }
            for (int i = 0; i < k; i++)
            {
                intervals.Add(0);
            }
            for (int i = 0; i < N; i++)
            {
               double chance = get_rand();
               for (int j = 0; j < k; j++) {
                    if (chance.CompareTo(numbers[j])>=0 &&
chance.CompareTo(numbers[j+1])<0)
                    {
                        intervals[j] += 1;
                    }
                }
            }
            Console.WriteLine("Intervals:");
            for (int i = 0; i < k; i++) {
```

```
                Console.WriteLine($"{i} interval: {intervals[i]}");
            }
            double p = 1.0 / k;
            double sum = 0;
            for (int i = 0; i < k; i++)
            {
                sum += Math.Pow(intervals[i], 2) / p;
            }
            sum /= N;
            sum -= N;
            Console.WriteLine($"Xi Square: {Math.Round(sum, 3)}");
        }
        public void autocorr(int N, int k)
        {
            List<double> X = new List<double>(N);
            double exp = 0.0;
            double exp_sq = 0.0;
            for (int i = 0; i < N; i++)
            {
                X.Add(get_rand());
                exp += X[i];
                exp_sq += Math.Pow(X[i], 2);
            }
            exp /= N;
            exp_sq /= N;
            double sq_exp = Math.Pow(exp, 2);
            double disp = exp_sq - sq_exp;
            List<double> R_a = new List<double>();
            for (int offset = 0; offset < 100; offset += 10)
            {
                double R = 0.0;
                for (int i = 0; i < N; i++)
                {
                    R += (X[i] - exp) * (X[(i + offset) % N] - exp);
                }
                R /= disp * N;
                R_a.Add(R);
                Console.WriteLine($"Tau: {offset} - Autocorrelation:
{Math.Round(R, 7)}");
            }
            double av = 0.0;
            for (int i = 0; i < R_a.Count; i++)
            {
                av += R_a[i];
            }
            Console.WriteLine($"Average Autocorrelation:
{Math.Round(av/R_a.Count, 7)}");
        }
        static void Main()
        {
            var a = new Program();
            a.xi_sq(MAX, 10);
            a.autocorr(MAX, 10);
            Console.ReadLine();
        }
    }
}
```