МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Генерация случайных чисел

Реферат

студента 1 курса 111 группы
направления 02.03.02 "Фундаментальная информатика и информационные технологии"
факультета КНиИТ
Карасева Арсения Михайловича

Проверил
Программист _____ В. С. Петров

Саратов 2021

СОДЕРЖАНИЕ

1 Какими бывают случайные числа?

Стоит начать с того, какими бывают случайные числа. Существует два вида случайных чисел: истинные случайные числа и псевдослучайные числа.

1.1 Истинные случайные числа

Истинные случайные числа можно получить из непредсказуемых физических явлений. Работа устройств, генерирующих такие числа, основана на использовании надёжных источников энтропии, таких, как тепловой шум, дробовой шум, фотоэлектрический эффект, квантовые явления и т. д. Эти процессы в теории абсолютно непредсказуемы, на практике же получаемые из них случайные числа проверяются с помощью специальных статистических тестов. Например, анализ уровня шумов звуковой карты компьютера. Последние байты этой величины будут являться истинными случайными числами. [1]

1.2 Псевдослучайные числа

У псевдослучайных чисел есть некоторый математический алгоритм, по которому они генерируются. За основу берётся какое-то стартовое число (его называют seed). Соответственно, Генератор псевдослучайных чисел (ГПСЧ) — алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению. Каждое новое число в последовательности ГПСЧ генерируется исходя из предыдущего определенным способом. [2]

Пример самого примитивного ГПСЧ:

```
1 int PRNG()
2 {
3         static unsigned long int seed = 5643;
4         seed = seed * 1103515245 + 12345;
5         return (unsigned int)(seed / 65536) % 32768;
6 }
```

Этот алгоритм не является хорошим, из–за своего распределения вероятностей — очень маленький шанс «выбить» маленькое число. А что же такое распределение?

2 Распределения

2.1 Ключевые термины

— Распределение вероятностей — это закон, описывающий область значений случайной величины и соответствующие вероятности появления этих значений. Грубо говоря, распределения отвечают за вероятность, с которой определённые числа будут генерироваться.

— Плотность вероятности — вещественная функция, характеризующая сравнительную вероятность реализации тех или иных значений случайной переменной.

— Функция распределения в теории вероятностей — функция, характеризующая распределение случайной величины.

— Математическое ожидание (μ) — среднее значение случайной величины. В случае непрерывной случайной величины подразумевается взвешивание по плотности распределения. [3]

— Среднеквадратическое отклонение (σ) — наиболее распространённый показатель рассеивания значений случайной величины относительно её математического ожидания.

2.2 Равномерное распределение

Равномерное распределение в теории вероятностей — распределение случайной вещественной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины, характеризующееся тем, что плотность вероятности на этом промежутке почти всюду постоянна. То есть вероятности появлений всех чисел в диапазоне равны, соответственно, это самое подходящее распределение для ГПСЧ.

Математическое ожидание в этом распределении равно $\mu = (b-a)/2$, то есть для $a = 2$ и $b = 6$ самое ожидаемое число будет 4. Функция rand() в C++ имеет равномерное распределение.

2.3 Нормальное распределение (Гауссово)

Нормальное распределение — распределение вероятностей, которое в одномерном случае задаётся функцией плотности вероятности, совпадающей с функцией Гаусса:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Таким образом, одномерное нормальное распределение является двухпараметрическим семейством распределений, которое принадлежит экспоненциальному классу распределений. [4] Стандартным нормальным распределением называется нормальное распределение с математическим ожиданием $\mu = 0$ и стандартным отклонением $\sigma = 1$. [5]

Я реализовал генерацию случайных чисел на нормальном распределении, всё свелось к анализу графика функции нормального распределения для получения значения в некотором промежутке. Код реализации:

```
#include <iostream>
#include <vector>
#include <math.h>
#include <iomanip>
#include <map>
#define sqrt_2 1.4142135623730
#define sqrt_pi_2 1.2533141373160
#define eps 1e-5


const double values[] = { 0.0,          //reversedFunction(0.5)
                          0.125661,     //reversedFunction(0.55)
                          0.253347,     //reversedFunction(0.6)
                          0.385320,     //reversedFunction(0.65)
                          0.524401,     //reversedFunction(0.7)
                          0.674490,     //reversedFunction(0.75)
                          0.841621,     //reversedFunction(0.8)
                          1.036433,     //reversedFunction(0.85)
                          1.281552,     //reversedFunction(0.9)
                          1.554774,     //reversedFunction(0.94)
                          1.644854,     //reversedFunction(0.95)
                          1.880794,     //reversedFunction(0.97)
                          2.053749,     //reversedFunction(0.98)
                          2.326348 };   //reversedFunction(0.99)

```

```
26
27 double normalFunction(double x)
28 {
29     if (x == 0.5) {
30         return 0.0;
31     }
32     if (x < eps) {
33         return -4.3;
34     }
35     if (x > 1 - eps) {
36         return 4.3;
37     }
38
39
40     int sign = 1;
41     if (x < 0.5) {
42         x = 1 - x;
43         sign = -1;
44     }
45     double xk1 = values[(int)(round(10 * x) - 5)], xk = xk1;
46     double b = 1 - 2 * x;
47     do
48     {
49         xk = xk1;
50         double t = xk / sqrt_2;
51         xk1 = xk - sqrt_pi_2 * exp(t * t) * (erf(t) + b);
52     }
53     while (fabs(xk - xk1) >= eps);
54
55
56     return sign == 1 ? xk1 : -xk1;
57 }
58
59
```

```
60 double getNormRandVal()
61 {
62     return normalFunction((double)rand() / ((double)RAND_MAX));
63 }
64
65
66 double finalResult(double m, double sigma)
67 {
68     std::vector<double> vec;
69     double randNum;
70     randNum = m + sigma * getNormRandVal();
71
72
73     return randNum;
74 }
```

Получается, в данном случае мы «переводим» равномерное распределение в нормальное.

2.4 Экспоненциальное распределение

Экспоненциальное распределение моделирует время между двумя последовательными свершениями одного и того же события. Для подсчёта вероятности нужен один параметр (λ), причём $\lambda > 0$. [5] Тогда плотность вероятности этого распределения имеет вид:

$$f_X(x) = \begin{cases} \lambda\, e^{-\lambda x}, x \geq 0, \\ 0, x < 0. \end{cases}$$

Пример: пусть существует магазин, в который время от времени заходят покупатели. При определённых допущениях время между появлениями двух последовательных покупателей будет случайной величиной с экспоненциальным распределением. Среднее время ожидания нового покупателя («время затухания») = $\frac{1}{\lambda}$, а сама λ — среднее число покупателей за единицу времени.

Также математическим ожиданием для экспоненциального распределения является $\frac{1}{\lambda}$. Соответственно, чем меньше лямбда, тем больше время затухания.

Реализация алгоритма генерации случайных чисел на экспоненциальном распределении:

```
1 double getRandomnumber()
2 {
3         return (double)rand() / (double)RAND_MAX;
4 }
5
6
7 double exponentialDistribution(double ly)
8 {
9         double x, u;
10        u = getRandomnumber();
11        x = log(1 - u) * (1 / (-ly)); // Случайное число, взятое
12        return x;   // из экспоненциального распределения.
13 }
```

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Поиск генераторов истинных случайных чисел[Электронный ресурс]. — 2021. — URL: https://habr.com/ru/company/mailru/blog/408181/ (Дата обращения 24.05.2021) Загл. с экр. яз. рус.

2 Генерация случайных чисел[Электронный ресурс]. — 2021. — URL: https://ravesli.com/urok-71-generatsiya-sluchajnyh-chisel-funktsii-srand-i-rand/ (Дата обращения 24.05.2021) Загл. с экр. яз. рус.

3 *Б. В. Гнеденко,*. Курс теории вероятностей / Б. В. Гнеденко. — Москва, Россия: Едиториал УРСС, 2019. — Pp. 158–175.

4 *В.П. Чистяков,*. Курс теории вероятностей / В.П. Чистяков. — Москва, Россия: Наука, 1987. — Pp. 71–122.

5 Генераторы непрерывно распределенных случайных величин[Электронный ресурс]. — 2021. — URL: https://habr.com/ru/post/263993/ (Дата обращения 24.05.2021) Загл. с экр. яз. рус.