

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Моделирование

ОТЧЕТ  
по лабораторной работе № 3  
на тему  
ПОСТРОЕНИЕ И ИССЛЕДОВАНИЕ ИМИТАЦИОННОЙ МОДЕЛИ  
ДИСКРЕТНО-СТОХАСТИЧЕСКОЙ СМО  
ВАРИАНТ № 4

Студент:

П.В. Сякачѐв

Проверила:

Ю.О. Герман

МИНСК 2022

## 1. Цель работы

Изучить методы имитационного моделирования поведения дискретно-стохастической СМО.

## 2. Краткие теоретические сведения

### 2.1 Линейный конгруэнтный алгоритм

Данный алгоритм может быть выражен следующей формулой:

$$X(i) = (a * X(i - 1) + c) \bmod m$$

Новое случайное число формируется из суммы константы  $c$  и произведения предыдущего случайного числа и константы  $a$ . Обе константы должны быть простыми числами.

Для получения чисел в определённом диапазоне результат предыдущей операции делится по модулю на константу  $m$ , которая в случае определения вероятностей должна быть равна 101, чтобы получить числа в диапазоне от 0 до 100 включительно.

Методом подбора установлено, что константы  $a$  и  $c$  равные 911 и 733 соответственно и начальное число 32 позволяют добиться близкого к равномерному распределению.

Просчитаем несколько шагов генерации псевдослучайных чисел с заданными выше параметрами.

$$X(0) = (911 * 32 + 733) \bmod 101 = 90$$

Теперь, получив первое число последовательности, подставим его в формулу для получения второго, такое действие означает, что формула является рекуррентной, то такой, в которой для выражения следующего члена последовательности используются предыдущие члены этой же последовательности. Продолжим последовательность:

$$X(1) = (911 * 90 + 733) \bmod 101 = 4$$

$$X(2) = (911 * 4 + 733) \bmod 101 = 34$$

$$X(3) = (911 * 34 + 733) \bmod 101 = 94$$

### 2.2 Представление участков переходов состояний на единичном отрезке

Рассмотрим матрицу переходных состояний 3.1 из третьего раздела. Пусть система находится в состоянии  $S_0$ , тогда она имеет вероятности

соответственно 0.4, 0.2 и 0.4 для отсутствия перехода в другое состояние, перехода в состояние  $S1$  и перехода в состояние  $S2$  соответственно, тогда можно представить в виде участков отрезка от нуля до единицы, где части равны вероятностям переходов, то есть поровый отрезок будет находится в пределах  $[0; 0.4]$ , второй отрезок –  $(0.4; 0.6]$ , а последний –  $(0.6; 1]$ .

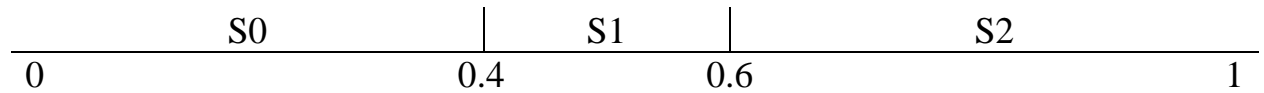


Рисунок 2.2.1 – единичный отрезок с обозначением участков переходов из состояния  $S0$

Проведём аналогичную операцию для состояний  $S1$  и  $S2$  и изобразим участки на единичных отрезках.

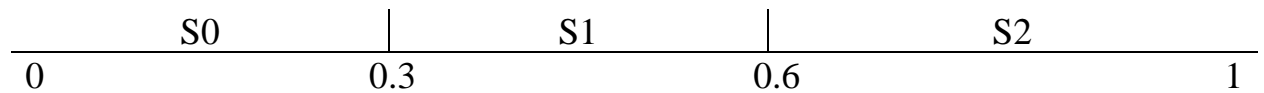


Рисунок 2.2.2 – единичный отрезок с обозначением участков переходов из состояния  $S1$

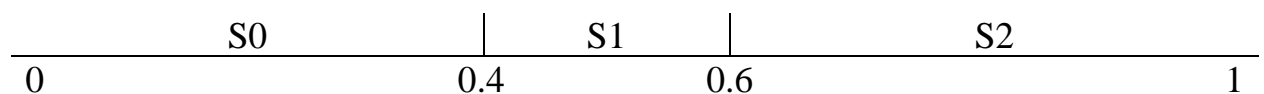


Рисунок 2.2.3 – единичный отрезок с обозначением участков переходов из состояния  $S2$

## 2.3 Пример последовательности переходов

Используем последовательность псевдослучайных чисел из раздела 2.1, приведём её к вероятностям, то есть поделим на 100, получим: 0.9, 0.04, 0.34, 0.94, 0.12, 0.5, 0.25, 0.76.

Возьмём за начальное состояние  $S0$ , тогда имея вероятность 0.9 мы попадаем в участок перехода к  $S2$ .

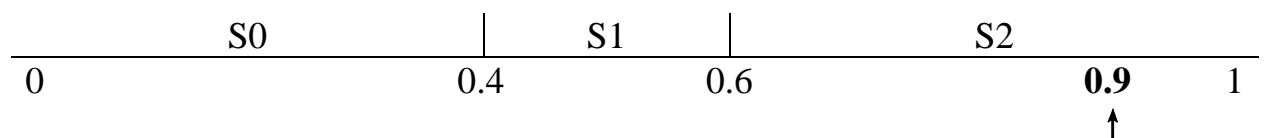


Рисунок 2.3.1 – первый шаг примера последовательности переходов

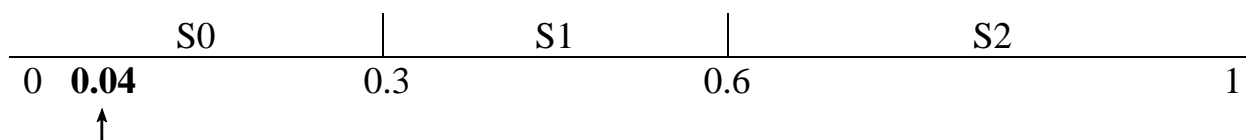


Рисунок 2.3.2 – второй шаг примера последовательности переходов

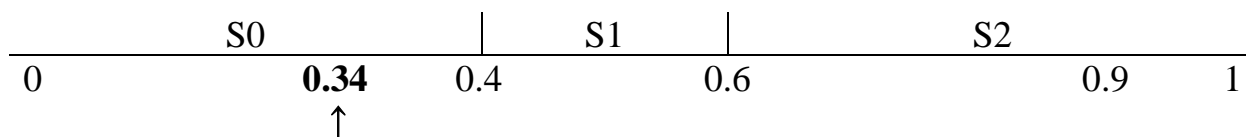


Рисунок 2.3.3 – третий шаг примера последовательности переходов

Подобным образом продолжим проводить данный алгоритм до тех пор, пока количество итераций не достигнет минимум 100.

### 3. Задание

Для всех вариантов в задании необходимо выполнить не менее 100 итераций.

Пусть матрица переходных вероятностей  $P$ :

Таблица 3.1 – матрица переходных вероятностей

	S0	S1	S2
S0	0.4	0.2	0.4
S1	0.3	0.3	0.4
S2	0.4	0.2	0.4

3.1 Найти установившиеся вероятности состояний системы:  $P_0$ ,  $P_1$ ,  $P_2$  методом имитационного моделирования.

3.2 Рассчитать на основе моделирования число шагов до попадания поглощающее состояние для матрицы вероятностей переходов.

Таблица 3.2 – матрица вероятностей переходов

	S0	S1	S2
S0	0.4	0.2	0.4
S1	0.3	0.3	0.4
S2	0	0	1.0

## 4. Ход работы

Для выполнения работы написана консольная программа на языке C#. Для получения псевдослучайных чисел при помощи линейного конгруэнтного алгоритма был написан метод Random:

```
const int a = 911;
const int c = 733;
const int m = 101;
static int Xprev = 32;

static double Random()
{
    Xprev = (a * Xprev + c) % m;
    return (double)Xprev / 100;
}
```

Для нахождения следующего состояния системы был написан метод NextStep:

```
static int NextStep(int s, double[,] matrix)
{
    double rand = Random();
    if (rand < matrix[s, 0])
        return 0;
    else if (rand >= (matrix[s, 0] + matrix[s, 1]))
        return 2;
    else return 1;
}
```

### 4.1 Найти установившиеся вероятности состояний системы: P0, P1, P2 методом имитационного моделирования.

Для нахождения вероятности состояний системы необходимо воспользоваться методом из разделов 2.2 и 2.3, алгоритм приведён ниже:

```
while (true)
{
    k++;
    for (int i = 0; i < 100; i++)
    {
        cond = NextStep(cond, matrix1);
        conds[cond]++;
    }
    for (int i = 0; i < 3; i++)
        if (probability[i] == (double)conds[i] / (100 * k))
        {
            flag = true;
        }
        else
        {
            flag = false;
            probability[i] = (double)conds[i] / (100 * k);
        }

    if (flag) break;
}
```

## 4.2 Рассчитать на основе моделирования число шагов до попадания поглощающее состояние для матрицы вероятностей переходов.

Для расчёта среднего числа шагов до попадания системы в поглощающее состояние нужно сначала вычислить, какое из состояний является поглощающим:

```
int abs = 0;
for (int i = 0; i < 3; i++)
    for (int j = 0; j < 3; j++)
        if (matrix2[i, j] == 1)
        {
            abs = i;
            break;
        }
```

После этого используется следующий алгоритм: для каждого из непоглощающих состояний для одинаковой последовательности псевдослучайных чисел проводится процесс переходов до попадания в поглощающее состояние, в случае попадания в поглощающее состояние система возвращается к изначальному состоянию и проводит цикл заново, но последовательность псевдослучайных чисел при этом не сбрасывается, цикл продолжается пока не будет проведено 100 итераций, после чего аналогичная операция проводится для следующего непоглощающего состояния.

При каждом попадании в поглощающее состояние количество шагов записывается в массив, после завершения цикла вычисляется среднее число шагов для начала из каждого непоглощающего состояния.

Код алгоритма приведён ниже:

```
for (int j = 0; j < 3; j++)
{
    if (j == abs) break;
    Xprev = x;

    for (int i = 0; i < 100; i++)
    {
        cond = j;
        while (true)
        {
            steps[j]++;
            cond = NextStep(cond, matrix2);
            if (cond == abs)
                break;
        }
    }
}
```

## 5. Вывод

Изучены методы имитационного моделирования поведения дискретно-стохастической СМО.