Учреждение образования

Белорусский государственный университет информатики

и радиоэлектроники

Факультет компьютерных технологий ИИТ

Лабораторная работа №3

Моделирование

Разработка, отладка и исследование  
 программной модели дискретно-стохастической СМО

Выполнил: студенты гр.180511

Жданко М.И.

Гриневич А.А..

Проверил: преподаватель Лашкевич Е.М.

Минск, 2014

**Задание**

Для СМО заданной конфигурации построить имитационную модель и исследовать ее (разработать алгоритм и написать имитирующую программу, предусматривающую сбор и статистическую обработку данных для получения оценок заданных характеристик СМО). Распределение интервалов времени между заявками во входном потоке и интервалов времени обслуживания – геометрическое с соответствующим параметром (ρ, π1, π2). Если ρ не задано, то входной поток – регулярный (с указанным в обозначении источника числом тактов между заявками).

Ротк – вероятность отказа;

Рбл – вероятность блокировки;

Lоч – средняя длина очереди; Q – относительная пропускная способность;

А – абсолютная пропускная способность.

**Процедура выполнения**

1) Исходные данные



|  |  |  |  |
| --- | --- | --- | --- |
| *ρ* | *π*1 | *π*2 | Цель исследования |
| - | 0,6 | 0,5 | Lоч |

2) Анализ задания

Схема содержит источник с отбрасыванием заявки и фиксированным временем ожидания выдачи заявки (2 такта до выдачи), накопитель на 2 заявки, два канала с вероятностями просеивания (не обслуживания заявки) π1 и π2.

Граф состояний кодируется четырехкомпонентным вектором *TN1К2К3,* где

*T* – время до выдачи очередной заявки источником, *T*={2,1,0}

2 – два такта до выдачи заявки

1 – один такт до выдачи заявки (по окончании такта заявка поступит в канал обслуживания)

*N1* – количество заявок, находящихся в накопителе (длина очереди), *N1*={0,1,2}

0 – заявок в очереди на обслуживание нет

1 – одна заявка в очереди;

2 – две заявки в очереди.

*К2* и *К3* – состояние каналов обслуживания, *К2* (*К3*)={0,1}

0 – канал свободен

1 – канал занят обслуживанием заявки.

3) Рассмотрев все возможные состояния системы, строим матрицу переходов, в виде таблицы

Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 2000 | 1000 | 2010 | 1010 | 2011 | 1001 | 1011 | 2111 | 1111 | 2211 | 1211 |
| 1 | 2000 | 0 | P1,2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1000 | 0 | 0 | P2,3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2010 | 0 | P3,2 | 0 | P3,4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1010 | 0 | 0 | P4,3 | 0 | P4,5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 2011 | 0 | P5,2 | 0 | P5,4 | 0 | P5,6 | P5,7 | 0 | 0 | 0 | 0 |
| 6 | 1001 | 0 | 0 | P6,3 | 0 | P6,5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1011 | 0 | 0 | P7,3 | 0 | P7,5 | 0 | 0 | P7,8 | 0 | 0 | 0 |
| 8 | 2111 | 0 | 0 | 0 | P8,4 | 0 | 0 | P8,7 | 0 | P8,9 | 0 | 0 |
| 9 | 1111 | 0 | 0 | 0 | 0 | P9,5 | 0 | 0 | P9,8 | 0 | P9,10 | 0 |
| 10 | 2211 | 0 | 0 | 0 | 0 | 0 | 0 | P10,7 | 0 | P10,9 | 0 | P10,11 |
| 11 | 1211 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P11,8 | 0 | P11,10 | 0 |

Таблица 2.а

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | **∑** |
| 2000 | 1000 | 2010 | 1010 | 2011 | 1001 | 1011 | 2111 | 1111 | 2211 | 1211 |
| 1 | 2000 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 1000 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 2010 | 0 | π1 | 0 | 1-π1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1010 | 0 | 0 | 1-π1 | 0 | π1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 5 | 2011 | 0 | 1-π1 1-π2 | 0 | π1 1-π2 | 0 | 1-π1 π2 | π1 π2 | 0 | 0 | 0 | 0 | 1 |
| 6 | 1001 | 0 | 0 | 1-π2 | 0 | π2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 7 | 1011 | 0 | 0 | 1-π1 1-π2 | 0 | (1-π1 π2) + (π1 1-π2) | 0 | 0 | π1 π2 | 0 | 0 | 0 | 1 |
| 8 | 2111 | 0 | 0 | 0 | 1-π1 1-π2 | 0 | 0 | (1-π1 π2) + (π1 1-π2) | 0 | π1 π2 | 0 | 0 | 1 |
| 9 | 1111 | 0 | 0 | 0 | 0 | 1-π1 1-π2 | 0 | 0 | (1-π1 π2) + (π1 1-π2) | 0 | π1 π2 | 0 | 1 |
| 10 | 2211 | 0 | 0 | 0 | 0 | 0 | 0 | 1-π1 1-π2 | 0 | (1-π1 π2) + (π1 1-π2) | 0 | π1 π2 | 1 |
| 11 | 1211 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1-π1 1-π2 | 0 | (1-π1 π2) + (π1 1-π2) +  (π1 π2) | 0 | 1 |

В последнем столбце для самопроверки просуммируем построчно вероятности переходов.

Подставим числовые значения

Таблица 2.б

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | **∑** |
| 2000 | 1000 | 2010 | 1010 | 2011 | 1001 | 1011 | 2111 | 1111 | 2211 | 1211 |
| 1 | 2000 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 1000 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 2010 | 0 | 0,6 | 0 | 0,4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1010 | 0 | 0 | 0,4 | 0 | 0,6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 5 | 2011 | 0 | 0,2 | 0 | 0,3 | 0 | 0,2 | 0,3 | 0 | 0 | 0 | 0 | 1 |
| 6 | 1001 | 0 | 0 | 0,5 | 0 | 0,5 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 7 | 1011 | 0 | 0 | 0,2 | 0 | 0,5 | 0 | 0 | 0,3 | 0 | 0 | 0 | 1 |
| 8 | 2111 | 0 | 0 | 0 | 0,2 | 0 | 0 | 0,5 | 0 | 0,3 | 0 | 0 | 1 |
| 9 | 1111 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0 | 0,5 | 0 | 0,3 | 0 | 1 |
| 10 | 2211 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,5 | 0 | 0,3 | 1 |
| 11 | 1211 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0,2 | 0 | 0,8 | 0 | 1 |

1

1

π1

1-π1

π1

π1π2

1-π1

(1-π1) π2

π1(1- π2)

(1- π1) (1- π2)

π1

π1

1-π1

π1π2

(1- π1) π2+ π1(1-π2)

(1- π1) (1- π2)

π1π2

(1- π1) π2+ π1(1-π2)

(1- π1) (1- π2)

π1π2

(1- π1) π2+ π1(1-π2)

(1- π1) (1- π2)

(1- π1) (1- π2)

(1- π1) π2+ π1(1-π2)

π1π2

(1- π1) (1- π2)

(1- π1) π2+ π1(1-π2)+ π1 π2

Рисунок 1 - Граф состояний

6) Расчет параметров данной системы с использованием построенной модели.

**а.** Исходя из графа состояний, построим систему уравнений для нахождения **вероятностей состояний**, воспользовавшись выражением:



дополнив систему нормировочным уравнением 

**б.** Исключим из системы уравнение , подставим значения *π*1=0,6 и *π*2=0,5 и приведем к каноническому виду:

**в.** Решив данную систему, получим значения вероятностей состояний:

Просуммировав для самопроверки вероятности состояний получим единицу.

P2000=0

P1000=0,147

P2010=0,269

P1010 =0,227

P2011 =0,199

P1011 =0,075

P1001 =0,039

P2111 =0,028

P1111 =0,011

P2211 =0,004

P1211=0,001

**г.** Определим значения величин, являющихся целью исследования:

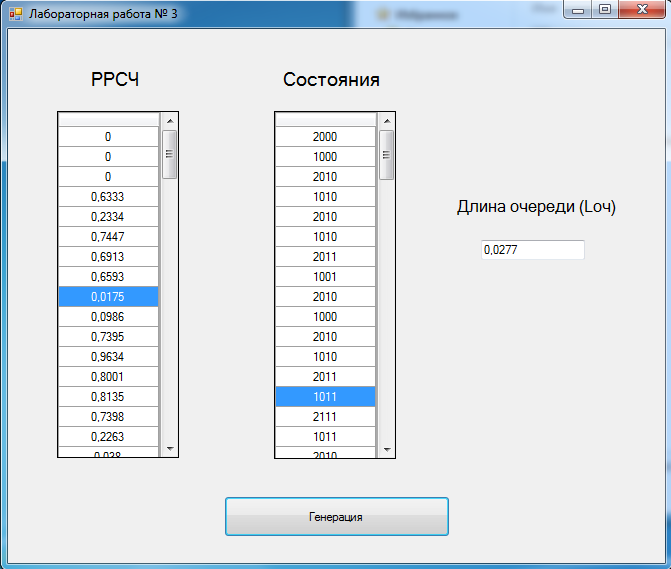
– средняя длина очереди *Lоч*:

,

где *i* – номер состояния, *j* - число заявок в очереди в *i*-том состоянии, s – число состояний системы

1\*P2111+1\*P1111+2\*P2211+ 2\*P1211=0,028+0,011+2\*0,004+2\*0,001=0,049

Скриншот программы



Листинг программы

namespace Lab3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

dataGridView1.RowCount = 100;

dataGridView2.RowCount = 100;

dataGridView1.Rows[0].Cells[0].Value = 0;

dataGridView1.Rows[1].Cells[0].Value = 0;

dataGridView1.Rows[2].Cells[0].Value = 0;

dataGridView2.Rows[0].Cells[0].Value = 2000;

dataGridView2.Rows[1].Cells[0].Value = 1000;

dataGridView2.Rows[2].Cells[0].Value = 2010;

}

Random rand = new Random();

string sost = "2010";

double gen = 0;

int row1 = 2;

int row2 = 2;

double dlina = 0;

double[] sost2010 = { 0, 0.6, 1 };

double[] sost1010 = { 0, 0.4, 1 };

double[] sost2011 = { 0, 0.2, 0.5, 0.7, 1 };

double[] sost1001 = { 0, 0.5, 1 };

double[] sost1011 = { 0, 0.2, 0.7, 1 };

double[] sost2111 = { 0, 0.2, 0.7, 1 };

double[] sost1111 = { 0, 0.2, 0.7, 1 };

double[] sost2211 = { 0, 0.2, 0.7, 1 };

double[] sost1211 = { 0, 0.2, 1 };

string Perehod(double r,string sost)

{

if (sost == "1000")

{

return "2010";

}

if(sost=="2010")

{

if (r > sost2010[0] && r < sost2010[1])

return "1000";

else return "1010";

}

if (sost == "1010")

{

if (r > sost1010[0] && r < sost1010[1])

return "2010";

else return "2011";

}

if (sost == "2011")

{

if (r > sost2011[0] && r < sost2011[1])

return "1000";

if (r >= sost2011[1] && r < sost2011[2])

return "1010";

if (r >= sost2011[2] && r < sost2011[3])

return "1001";

else return "1011";

}

if (sost == "1001")

{

if (r > sost1010[0] && r < sost1010[1])

return "2010";

else return "2011";

}

if (sost == "1011")

{

if (r > sost2011[0] && r < sost2011[1])

return "2010";

if (r >= sost2011[1] && r < sost2011[2])

return "2011";

else return "2111";

}

if (sost == "2111")

{

dlina += 0.0277;

if (r > sost2011[0] && r < sost2011[1])

return "1010";

if (r >= sost2011[1] && r < sost2011[2])

return "1011";

else return "1111";

}

if (sost == "1111")

{

dlina += 0.0103;

if (r > sost2011[0] && r < sost2011[1])

return "2011";

if (r >= sost2011[1] && r < sost2011[2])

return "2111";

else return "2211";

}

if (sost == "2211")

{

dlina += 2\*0.004;

if (r > sost2011[0] && r < sost2011[1])

return "1011";

if (r >= sost2011[1] && r < sost2011[2])

return "1111";

else return "1211";

}

if (sost == "1211")

{

dlina += 2\*0.001;

if (r > sost2011[0] && r < sost2011[1])

return "2111";

else return "2211";

}

return "error";

}

private void button1\_Click(object sender, EventArgs e)

{

gen = rand.NextDouble();

row1++;

row2++;

dataGridView1.Rows[row1].Cells[0].Value = Math.Round(gen,4);

sost = Perehod(gen, sost);

dataGridView2.Rows[row2].Cells[0].Value = sost;

textBox1.Text = dlina.ToString();

}

}

}