



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ Информационные технологии и системы управления
КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ ПО КУРСУ «ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТОВ»

Лабораторная работа №4

Студент _____ ИУ7-83Б
(Группа)

_____ Зыкин Д.А.
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

_____ Куров А.В.
(И.О.Фамилия)

2020 г.

Цель работы

Реализация ОЦКП на имитационной модели функционирования СМО для нахождения среднего времени ожидания заявок в очереди.

Условие работы

1. составить матрицу планирования для проведения ОЦКП для СМО с двумя генераторами заявок;
2. рассчитать необходимые величины для ОЦКП (звездное плечо);
3. по результатам ОЦКП вычислить коэффициенты нелинейной регрессионной зависимости;
4. предусмотреть возможность сравнения рассчитанной величины с реальной, полученной по результатам имитационного моделирования.

Теоретическая часть

В данной лабораторной работе интервалы для факторов подбираются таким образом, чтобы загрузка системы лежала в промежутке от 0 до 1. Таким образом обеспечивается функционирование системы в стационарном режиме.

В данной работе реализуется ортогональный центральный композиционный эксперимент(ОЦКП). Используется СМО с двумя генераторами и одним обработчиком. По условию первой лабораторной работы, заявки поступают по закону распределения Рэля, а обслуживаются по нормальному закону распределения.

Для генерации времени обработки на основе заданных промежутков рассчитываются параметры распределений. Для генератора будет определяться сигма по формуле:

$$\sigma = \frac{1}{\sqrt{\frac{\pi}{2} I}} \quad (1)$$

Для обработчика параметры распределения будут определяться по формуле:

$$M = \frac{1}{I} \quad (2)$$

$$D = coeff * M \quad (3)$$

, где coeff – это заданный пользователем коэффициент дисперсии.

Таким образом, эксперименты будут иметь четыре фактора, где:

1. первый фактор – интенсивность первого генератора;
2. второй фактор – интенсивность второго генератора;
3. третий фактор – интенсивность обработчика;
4. четвертый фактор – коэффициент дисперсия обработчика.

Таким образом, уравнение ОЦКП будет иметь вид:

$$y = b_0 x_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_{12} x_1 x_2 + b_{13} x_1 x_3 + b_{14} x_1 x_4 + b_{23} x_2 x_3 + b_{24} x_2 x_4 + b_{34} x_3 x_4 + b_{11} x_1^2 + b_{22} x_2^2 + b_{33} x_3^2 + b_{44} x_4^2 \quad (4)$$

Число экспериментов рассчитывается как:

$$N = 2^m + 2 \cdot m + n_c \quad (5)$$

, где m – число факторов, n_c – число экспериментов в центре плана (с координатами $x_1 = 0$ $x_2 = 0$ $x_3 = 0$ $x_4 = 0$). Таким образом, в данной работе:

$$N = 2^4 + 2 \cdot 4 + 1 = 25 \quad (6)$$

В результате план проведения ОЦКП с учетом кодирования факторов имеет вид:

n\p	x0	x1	x2	x3	x4	x1x2	x1x3	x1x4	x2x3	x2x4	x3x4	x_1^2	x_2^2	x_3^2	x_4^2	y
1	1	-1	-1	-1	-1	1	1	1	1	1	1	1-S	1-S	1-S	1-S	y1
2	1	-1	-1	-1	1	1	1	-1	1	-1	-1	1-S	1-S	1-S	1-S	y2
3	1	-1	-1	1	-1	1	-1	1	-1	1	-1	1-S	1-S	1-S	1-S	y3
4	1	-1	-1	1	1	1	-1	-1	-1	-1	1	1-S	1-S	1-S	1-S	y4
5	1	-1	1	-1	-1	-1	1	1	-1	-1	1	1-S	1-S	1-S	1-S	y5
6	1	-1	1	-1	1	-1	1	-1	-1	1	-1	1-S	1-S	1-S	1-S	y6
7	1	-1	1	1	-1	-1	-1	1	1	-1	-1	1-S	1-S	1-S	1-S	y7
8	1	-1	1	1	1	-1	-1	-1	1	1	1	1-S	1-S	1-S	1-S	y8
9	1	1	-1	-1	-1	-1	-1	-1	1	1	1	1-S	1-S	1-S	1-S	y9
10	1	1	-1	-1	1	-1	-1	1	1	-1	-1	1-S	1-S	1-S	1-S	y10
11	1	1	-1	1	-1	-1	1	-1	-1	1	-1	1-S	1-S	1-S	1-S	y11
12	1	1	-1	1	1	-1	1	1	-1	-1	1	1-S	1-S	1-S	1-S	y12
13	1	1	1	-1	-1	1	-1	-1	-1	-1	1	1-S	1-S	1-S	1-S	y13
14	1	1	1	-1	1	1	-1	1	-1	1	-1	1-S	1-S	1-S	1-S	y14
15	1	1	1	1	-1	1	1	-1	1	-1	-1	1-S	1-S	1-S	1-S	y15
16	1	1	1	1	1	1	1	1	1	1	1	1-S	1-S	1-S	1-S	y16
17	1	$-\alpha$	0	0	0	0	0	0	0	0	0	$\alpha^2 - S$	-S	-S	-S	y17
18	1	$+\alpha$	0	0	0	0	0	0	0	0	0	$\alpha^2 - S$	-S	-S	-S	y18
19	1	0	$-\alpha$	0	0	0	0	0	0	0	0	-S	$\alpha^2 - S$	-S	-S	y19
20	1	0	$+\alpha$	0	0	0	0	0	0	0	0	-S	$\alpha^2 - S$	-S	-S	y20
21	1	0	0	$-\alpha$	0	0	0	0	0	0	0	-S	-S	$\alpha^2 - S$	-S	y21
22	1	0	0	$+\alpha$	0	0	0	0	0	0	0	-S	-S	$\alpha^2 - S$	-S	y22
23	1	0	0	0	$-\alpha$	0	0	0	0	0	0	-S	-S	-S	$\alpha^2 - S$	y23
24	1	0	0	0	$+\alpha$	0	0	0	0	0	0	-S	-S	-S	$\alpha^2 - S$	y24
25	1	0	0	0	0	0	0	0	0	0	0	-S	-S	-S	-S	y25

Для проведения ОЦКП необходимо рассчитать дополнительные значения (звездное плечо). Из условия ортогональности матрицы планирования:

$$\alpha = \sqrt{\frac{n}{2} \cdot \left(\frac{N}{n} - 1\right)} = 2.12 \quad (7)$$

$$S = \sqrt{\frac{n}{N}} = 0.64 \quad (8)$$

Интерфейс

Для проведения эксперимента пользователь должен задать четыре промежутка и четыре значения из каждого промежутка.

MainWindow

Первый генератор

☒ Сигма

10,000

☐ Интенсивность

0,033

Второй генератор

☒ Сигма

10,000

☐ Интенсивность

0,033

Обработчик

Интенсивность

0,170

Дисперсия

0,170

Время моделирования

10000

Моделировать

Получить график

Эксперимент

Интенсивность первого генератора минимальная

0,030

максимальная

0,035

Интенсивность второго генератора минимальная

0,030

максимальная

0,035

Интенсивность обработчика минимальная

0,100

максимальная

0,200

Дисперсия обработчика минимальная

0,100

максимальная

0,200

ОЦКП:
b0 = 1.35
b1 = 0.02
b2 = 0.03
b3 = -0.74
b4 = 0.00
b5 = 0.26
b6 = 0.00
b7 = -0.01
b8 = -0.01
b9 = -0.03
b10 = 0.01
b11 = 0.01
b12 = 0.03
b13 = 0.10
b14 = 0.09

Сравнение результатов
Результат моделирования 0.9673
Результат ОЦКП: 1.0670
Отклонение: 10.31%

Сравнение результатов

Проведем эксперименты для разных значений факторов в интервалах от 0.03 до 0.035 для интенсивностей поступления заявок, от 0.1 до 0.2 для интенсивности обработчика и от 0.1 до 0.2 для коэффициента дисперсии обработчика.

Суммарная нагрузка будет лежать в интервале от 0.35 до 0.7. Режим стационарный.

Проведенные эксперименты:

- 1 I1 = 0.030; I2 = 0.030; I3 = 0.100; D = 0.100
- 2 Ожидаемая нагрузка 0.6000
- 3 Полученная нагрузка 0.6080
- 4 Время ожидания 2.3412
- 5
- 6
- 7 I1 = 0.030; I2 = 0.030; I3 = 0.100; D = 0.200
- 8 Ожидаемая нагрузка 0.6000
- 9 Полученная нагрузка 0.6064
- 10 Время ожидания 2.3228
- 11
- 12
- 13 I1 = 0.030; I2 = 0.030; I3 = 0.200; D = 0.100
- 14 Ожидаемая нагрузка 0.3000
- 15 Полученная нагрузка 0.3027
- 16 Время ожидания 0.6609

17
18
19 $I1 = 0.030; I2 = 0.030; I3 = 0.200; D = 0.200$
20 Ожидаемая загрузка 0.3000
21 Полученная загрузка 0.3000
22 Время ожидания 0.7035
23
24
25 $I1 = 0.030; I2 = 0.030; I3 = 0.200; D = 0.200$
26 Ожидаемая загрузка 0.3000
27 Полученная загрузка 0.3024
28 Время ожидания 0.7501
29
30
31 $I1 = 0.030; I2 = 0.035; I3 = 0.100; D = 0.100$
32 Ожидаемая загрузка 0.6500
33 Полученная загрузка 0.6564
34 Время ожидания 1.6338
35
36
37 $I1 = 0.030; I2 = 0.035; I3 = 0.100; D = 0.100$
38 Ожидаемая загрузка 0.6500
39 Полученная загрузка 0.6601
40 Время ожидания 1.7013
41
42
43 $I1 = 0.030; I2 = 0.035; I3 = 0.200; D = 0.100$
44 Ожидаемая загрузка 0.3250
45 Полученная загрузка 0.3273
46 Время ожидания 0.4565
47
48
49 $I1 = 0.030; I2 = 0.035; I3 = 0.200; D = 0.200$
50 Ожидаемая загрузка 0.3250
51 Полученная загрузка 0.3260
52 Время ожидания 0.4366
53
54
55 $I1 = 0.035; I2 = 0.030; I3 = 0.100; D = 0.100$
56 Ожидаемая загрузка 0.6500
57 Полученная загрузка 0.6566
58 Время ожидания 1.6782
59
60
61 $I1 = 0.030; I2 = 0.035; I3 = 0.100; D = 0.200$
62 Ожидаемая загрузка 0.6500
63 Полученная загрузка 0.6574

64 Время ожидания 1.7147
65
66
67 $I1 = 0.030$; $I2 = 0.035$; $I3 = 0.200$; $D = 0.100$
68 Ожидаемая загрузка 0.3250
69 Полученная загрузка 0.3265
70 Время ожидания 0.4238
71
72
73 $I1 = 0.030$; $I2 = 0.035$; $I3 = 0.200$; $D = 0.200$
74 Ожидаемая загрузка 0.3250
75 Полученная загрузка 0.3283
76 Время ожидания 0.4608
77
78
79 $I1 = 0.035$; $I2 = 0.030$; $I3 = 0.100$; $D = 0.100$
80 Ожидаемая загрузка 0.6500
81 Полученная загрузка 0.6564
82 Время ожидания 1.6643
83
84
85 $I1 = 0.035$; $I2 = 0.030$; $I3 = 0.100$; $D = 0.200$
86 Ожидаемая загрузка 0.6500
87 Полученная загрузка 0.6593
88 Время ожидания 1.7090
89
90
91 $I1 = 0.035$; $I2 = 0.030$; $I3 = 0.200$; $D = 0.100$
92 Ожидаемая загрузка 0.3250
93 Полученная загрузка 0.3263
94 Время ожидания 0.4550
95
96
97 $I1 = 0.035$; $I2 = 0.030$; $I3 = 0.200$; $D = 0.200$
98 Ожидаемая загрузка 0.3250
99 Полученная загрузка 0.3263
100 Время ожидания 0.4914
101
102 $I1 = 0.035$; $I2 = 0.030$; $I3 = 0.200$; $D = 0.200$
103 Ожидаемая загрузка 0.3250
104 Полученная загрузка 0.3272
105 Время ожидания 0.4632
106
107
108 $I1 = 0.035$; $I2 = 0.035$; $I3 = 0.100$; $D = 0.100$
109 Ожидаемая загрузка 0.7000
110 Полученная загрузка 0.7100

111	Время ожидания 2.2834
112	
113	I1 = 0.035; I2 = 0.035; I3 = 0.100; D = 0.100
114	Ожидаемая загрузка 0.7000
115	Полученная загрузка 0.7067
116	Время ожидания 2.5057
117	
118	
119	I1 = 0.035; I2 = 0.035; I3 = 0.100; D = 0.200
120	Ожидаемая загрузка 0.7000
121	Полученная загрузка 0.7053
122	Время ожидания 2.3534
123	
124	
125	I1 = 0.035; I2 = 0.035; I3 = 0.200; D = 0.100
126	Ожидаемая загрузка 0.3500
127	Полученная загрузка 0.3544
128	Время ожидания 0.7749
129	
130	
131	I1 = 0.035; I2 = 0.035; I3 = 0.200; D = 0.200
132	Ожидаемая загрузка 0.3500
133	Полученная загрузка 0.3532
134	Время ожидания 0.7794
135	
136	I1 = 0.033; I2 = 0.033; I3 = 0.170; D = 0.170
137	Ожидаемая загрузка 0.3882
138	Полученная загрузка 0.3900
139	Время ожидания 0.9238

При этом коэффициенты уравнения равны:

1	b0 = 1.35
2	b1 = 0.02
3	b2 = 0.03
4	b3 = -0.74
5	b4 = 0.00
6	b5 = 0.26
7	b6 = 0.00
8	b7 = -0.01
9	b8 = -0.01
10	b9 = -0.03
11	b10 = 0.01
12	b11 = 0.01
13	b12 = 0.03
14	b13 = 0.10
15	b14 = 0.09

Значения, полученные ОЦКП дали результаты близкие к получаемым моделированием. В среднем получено отличие на 13.2%.

Реализованные классы

Для хранения информации об экспериментах ОЦКП был реализован класс

`OExperiment`

```
1  using System;
2  namespace ExperimentsLibrary
3  {
4      public class OExperiment
5      {
6          public double FirstGenMinIntense { get; }
7          public double FirstGenMaxIntense { get; }
8          public double SecondGenMinIntense { get; }
9          public double SecondGenMaxIntense { get; }
10         public double ProcMinIntense { get; }
11         public double ProcMaxIntense { get; }
12         public double ProcMinD { get; }
13         public double ProcMaxD { get; }
14         public double FristGenIntense { get; }
15         public double SecondGenIntense { get; }
16         public double ProcIntense { get; }
17         public double ProcD { get; }
18         public double[] Results { get => results; set => results
= value; }
19         public double[] Coeffs { get => coeffs; set => coeffs =
value; }
20         public double[] ZCenterList { get => zCenterList; set =>
zCenterList = value; }
21         public double[] ZDeltaList { get => zDeltaList; set =>
zDeltaList = value; }
22         public double[] X { get => x; set => x = value; }
23         public double[] ExperimentResults { get; set; }
24
25         double[] coeffs;
26         double[] results;
27         double[] zCenterList;
28         double[] zDeltaList;
29         double[] x;
30
31         double linExpRes = 0;
32         double nonLinExpRes = 0;
33
34         int factorNumber = 0;
```

```

35         int paramsAmount = 4;
36         int coeffsAmount = 16;
37         int experimentsAmount = 25;
38         double S = 0.64;
39         double alpha = 2.12;
40
41         public OExperiment(double firstGenMinIntense, double
firstGenMaxIntense,
42                             double secondGenMinIntense,
double secondGenMaxIntense,
43                             double procMinIntense, double
procMaxIntense,
44                             double procMinD, double
procMaxD,
45                             double fristGenIntense, double
secondGenIntense, double procIntense, double procD,
46                             double alpha, double s)
47         {
48             FirstGenMinIntense = firstGenMinIntense;
49             FirstGenMaxIntense = firstGenMaxIntense;
50             SecondGenMinIntense = secondGenMinIntense;
51             SecondGenMaxIntense = secondGenMaxIntense;
52             ProcMinIntense = procMinIntense;
53             ProcMaxIntense = procMaxIntense;
54             ProcMinD = procMinD;
55             ProcMaxD = procMaxD;
56             FristGenIntense = fristGenIntense;
57             SecondGenIntense = secondGenIntense;
58             ProcIntense = procIntense;
59             ProcD = procD;
60
61             Coeffs = new double[coeffsAmount];
62             Results = new double[coeffsAmount];
63             ZCenterList = new double[paramsAmount];
64             ZDeltaList = new double[paramsAmount];
65             X = new double[paramsAmount];
66         }
67
68
69         public void GetCoeffs()
70         {
71             for (int i = 0; i < ExperimentResults.Length; i++)
72             {
73                 double sum = 0;
74                 for (int j = 0; j < coeffs.Length; j++)
75                 {
76                     double x = coeffs[j];

```

```

77         if (i > factorNumber + 1 + factorNumber *
(factorNumber - 1) / 2)
78             x -= S;
79             sum += ExperimentResults[j] * x;
80         }
81         if (i == 0)
82             coeffs[i] = sum / ExperimentResults.Length;
83         else if (i < factorNumber + 1)
84             coeffs[i] = sum / (Math.Pow(2, factorNumber)
+ 2 * alpha * alpha);
85         else if (i < factorNumber + 1 + factorNumber *
(factorNumber - 1) / 2)
86             coeffs[i] = sum / Math.Pow(2, factorNumber);
87         else
88             coeffs[i] = sum / (2 * Math.Pow(alpha, 4));
89     }
90 }
91 }
92 }

```

Для проведения экспериментов используется класс `PlotGenerator`

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using TimeElementsLibrary;
5
6  namespace ExperimentsLibrary
7  {
8      public static class PlotGenerator
9      {
10         static double modellingTime = 1000;
11         static double load_step = 0.05;
12         static double repeats = 30;
13
14         public static double GetExperimentResult(double
firstGenIntense, double secondGenIntense,
15                                                     double
procIntense, double procD,
16                                                     double
expectedLoad = 0)
17         {
18             double avgTime = 0;
19             double avgLoad = 0;
20             for (int i = 0; i < repeats; i++)
21             {

```

```

22         ModellingController controller = new
ModellingController(firstGenIntense, secondGenIntense,
procIntense, procD, modellingTime);
23         Report report = controller.StartModelling();
24         avgTime += report.GetAvgTime();
25         avgLoad += report.Load();
26     }
27     avgLoad /= repeats;
28     avgTime /= repeats;
29
30     Console.WriteLine($"I1 = {firstGenIntense:F3}; I2 =
{secondGenIntense:F3}; I3 = {procIntense:F3}; D = {procD:F3}");
31     Console.WriteLine($"Ожидаемая загрузка
{(firstGenIntense + secondGenIntense) /
procIntense:F4}\nПолученная загрузка {avgLoad:F4}\nВремя ожидания
{avgTime:F4}\n\n");
32     return avgTime;
33 }
34
35 public static string GetPlotPoints()
36 {
37     double load = 0.1;
38     double procIntense = 10;
39     double procD = 0.3;
40     List<double> y = new List<double>();
41     List<double> x = new List<double>();
42
43     while (load < 1)
44     {
45         double genIntense = (procIntense * load) / 2;
46
47         x.Add(load);
48         y.Add(GetExperimentResult(genIntense, genIntense,
procIntense, procD, load));
49         load += load_step;
50     }
51
52     StringBuilder builder = new StringBuilder();
53     for (int i = 0; i < x.Count; i++)
54     {
55         builder.Append($"{x[i]};{y[i]};\n");
56     }
57
58     return builder.ToString();
59 }
60 }
61 }

```

Защита

ОЦКП - это ортогональное центральное композиционное планирование.

Ортогональное - скалярное произведение любых 2 столбцов плана дает 0.

Центральное - все точки симметричны относительно центра плана.

Композиционное - состоит из нескольких частей, а именно: ядро плана, "звездные" точки и центр плана.

Для "звездного" плеча сначала высчитывается параметр S , пользуясь свойством симметрии, а потом, зная S , высчитывается "звездное" плечо на основе свойства ортогональности.

Факторы варьируются на пяти уровнях: $+1$, -1 , $+\alpha$, $-\alpha$, 0 .

Чтобы получить нелинейную зависимость нужно минимум 3 уровня.

Провести ПФЭ с 3 уровнями варьирования факторов.

В качестве ядра можно использовать ПФЭ или ДФЭ с разной степенью дробности.