



Проект по теории информации

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a6e9e59e-1986-413e-b76a-cdb18033c7b9/ПРОЕКТ_ТИ.docx

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d95a0b3d-5589-4d5e-ac4f-b71a58b91134/Листинг_программы_расчетов.docx

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/eca5cc80-f37f-4cdc-9b0f-8634c93d8be5/2012-uch-posob-kutuzov-tatarnikova.pdf>

http://e-biblio.ru/book/bib/06_management/teor_mass_obslug/158.9.21.html

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9035d9dd-b3e4-41db-8d5c-5a4db6de79cf/itmo354.pdf>

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/26217097-c06a-461e-9ec9-16cc43e37d13/Лабораторная_работа_6.doc

Цель работы:

Изучение моделей экспоненциальных сетей массового обслуживания и методов расчета их локальных и интегрированных показателей.

Задачи:

Для экспоненциальной разомкнутой сети массового обслуживания требуется рассчитать:

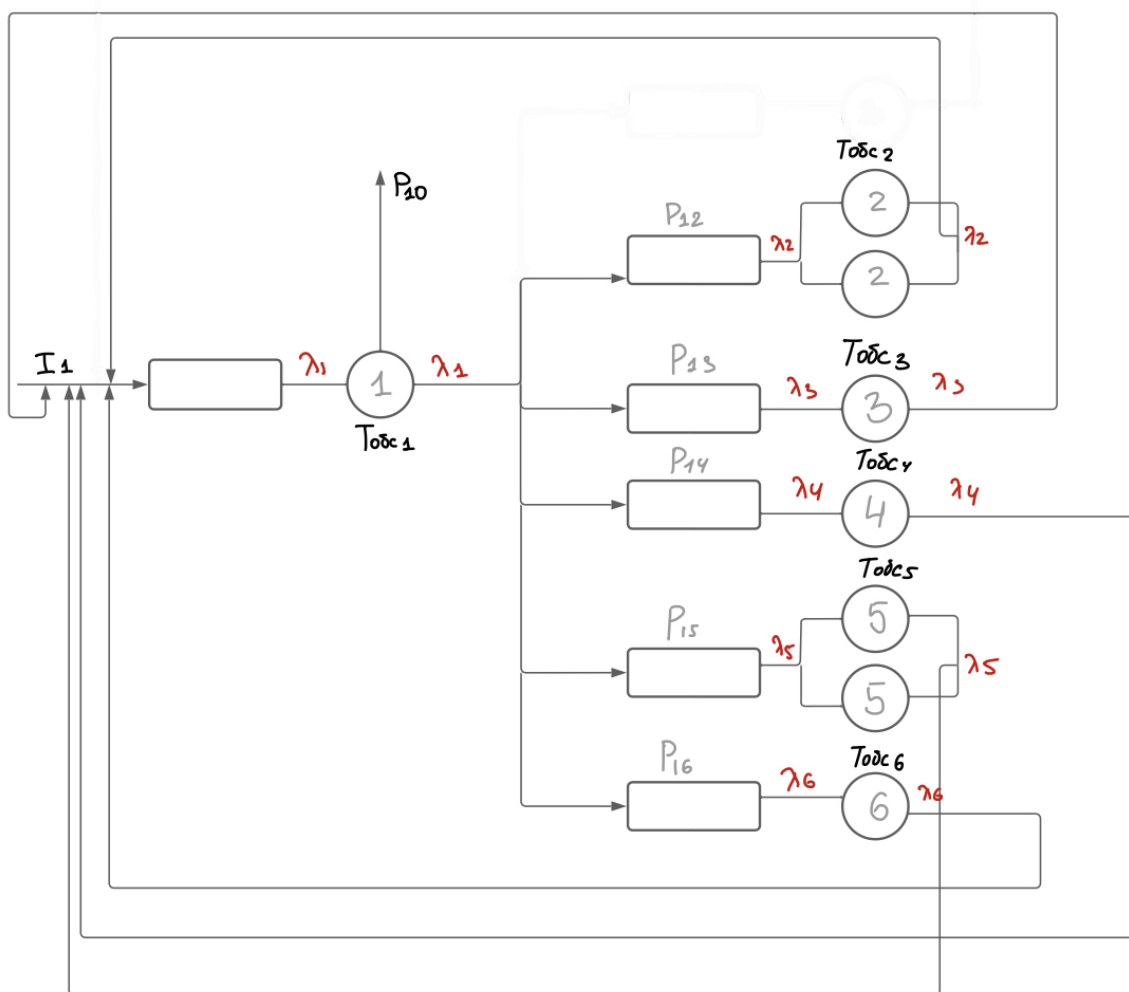
1. Частные характеристики узлов СеМО.
2. Среднее время пребывания заявки в сети. Для узлов сети с интенсивностями входного потока $\lambda_i > 0$.
3. Среднее входное время пребывания заявки в сети.
4. Абсолютную пропускную способность.
5. Условную пропускную способность.
6. Запас по пропускной способности.

Выполнение:

Сеть массового обслуживания (Семо) представляет собой совокупность конечного числа взаимосвязанных узлов обслуживания, в которой циркулируют заявки, переходящие в соответствии с маршрутной матрицей с выхода одного узла на вход другого. Каждый отдельный узел является разомкнутой СМО и отображает функционально самостоятельную часть реальной системы.

Семо используются для определения таких важных характеристик моделируемых систем как:

- производительность;
- время доставки заявок (сообщений, пакетов и пр.);
- вероятность потери заявки;
- вероятность блокировки узла;
- допустимые значения нагрузки, при которых обеспечивается требуемое качество обслуживания и др.



Для выполнения работы была построена Сеть массового обслуживания

Число узлов $N = 6$

Число каналов $K_1=K_3=K_4=K_6 = 1; K_2=K_5=2$

матрица вероятности передач:

	0	1	2	3	4	5	6
1	0,1	0	0,2	0,1	0,3	0,05	0,25
2	0	1	0	0	0	0	0
3	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0
5	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0

Интенсивность входного потока заявок $I_1=0$

$I_2=I_3=I_4=I_5=I_6=0$

Средние времена обслуживания заявок:

$T_{обс1}=0.07$

$T_{обс2}=0.05$

$T_{обс3}=0.03$

$T_{обс4}=0.01$

$T_{обс5}=0.06$

$T_{обс6}=0.1$

1) Расчет баланса интенсивностей сети

$$\lambda = I_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6$$

$$I = P_{10} \cdot \lambda_1$$

$$\lambda_2 = P_{12} \cdot \lambda_1$$

$$\lambda_3 = P_{13} \cdot \lambda_1$$

$$\lambda_4 = P_{14} \cdot \lambda_1$$

$$\lambda_5 = P_{15} \cdot \lambda_1$$

$$\lambda_6 = P_{16} \cdot \lambda_1$$

Отсюда

$$\lambda_1 = \frac{I}{P_{10}} = 10$$

$$\lambda_2 = \lambda_1 \cdot P_{12} = 2$$

$$\lambda_3 = \lambda_1 \cdot P_{13} = 1$$

$$\lambda_4 = \lambda_1 \cdot P_{14} = 3$$

$$\lambda_5 = \lambda_1 \cdot P_{15} = 0.5$$

$$\lambda_6 = \lambda_1 \cdot P_{16} = 2.5$$

2) Проверка стационарности

Стационарность сети означает, что среднее число заявок в любом ее фрагменте неизменно, т.е., суммарная интенсивность входящих в эту часть потоков равна суммарной интенсивности выходящих. Сеть стационарна, если стационарны все ее узлы, т.е. если

$$\rho_j \leq 1, \quad \overline{j = 1, N} \quad (2)$$

$$\rho_1 = \lambda_1 \cdot T_{\text{обс1}} = 0.7 < 1$$

$$\rho_2 = \lambda_2 \cdot T_{\text{обс2}} \div 2 = 0.05 < 1$$

$$\rho_3 = \lambda_3 \cdot T_{\text{обс3}} = 0.03 < 1$$

$$\rho_4 = \lambda_4 \cdot T_{\text{обс4}} = 0.03 < 1$$

$$\rho_5 = \lambda_5 \cdot T_{\text{обс5}} = 0.015 < 1$$

$$\rho_6 = \lambda_6 \cdot T_{\text{обс6}} = 0.25 < 1$$

3) Расчет локальных характеристик семо

3.1 Средняя длина очереди в узле

$$L_i = \frac{\rho_i^2}{1 - \rho_i}$$

$$L_1 = 1.633$$

$$L_2 = 0.00263$$

$$L_3 = 0.000928$$

$$L_4 = 0.000928$$

$$L_5 = 0.000228$$

$$L_6 = 0.0833$$

3.2 Среднее число заявок в узле

$$M_i = \frac{\rho_i}{1 - \rho_i}$$

$$M_1 = 2.333$$

$$M_2 = 0.0526$$

$$M_3 = 0.0309$$

$$M_4 = 0.0309$$

$$M_5 = 0.01523$$

M6 = 0.333

3.3 Средняя продолжительность пребывания заявки в очереди на обработку в узле

$$T_{i_{\text{очереди}}} = \frac{T_{i_{\text{обс}}} \cdot \rho_i}{1 - \rho_i}$$

Точ1=0.1633

Точ2=0.00263

Точ3=0.0009278

Точ4=0.0003

Точ5=0.0009

Точ6=0.0333

3.4 Среднее время пребывания заявки в СМО

$$T_{i_{\text{СМО}}} = \frac{T_{i_{\text{обс}}}}{1 - \rho_i}$$

Тсм01=0.2333

Тсм02=0.0526

Тсм03=0.0309278

Тсм04=0.010309

Тсм05=0.06091

Тсм06=0.1333

3.5 Нагрузка узла семо

$$B_i = T_{i_{\text{СМО}}} \cdot \lambda_i$$

b1 = 2.333

b2 = 0.10526

b3 = 0.0309

b4 =0.0309

b5 =.0304568

b6= 0.3333

3.6 Коэффициент простоя

$$\eta_i = 1 - \rho_i$$

n1 =0.299

n2 =0.95

n3 =0.97

n4=0.97

n5=0.985

n6=0.75

4. Суммарна нагрузка

$$B = \sum b_i = 2.86424$$

$$B = 2.86424$$

5. Среднее число заявок в сети

$$M = \sum M_i = 2.79638$$

$$M = 2.79638$$

6. Среднее число заявок на обслуживание в сети

$$L = \sum L_i = 1.7213$$

$$L = 1.7213$$

7. Среднее время ожидания в сети

$$T_{\text{оч}} = \frac{1}{I} \cdot \sum \lambda_i \cdot T_{\text{оч}_i} = 1.7242423$$

$$T_{\text{оч}} = 1.7242423$$

8. Среднее время пребывания в сети

$$T_{\text{семо}} = \frac{1}{I} \cdot \sum \lambda_i \cdot T_{\text{семо}_i} = 2.8642423$$

$$T_{\text{семо}} = 2.8642423$$

9. Передаточные коэффициенты

$$\alpha_{ij}, i, j = \overline{1, N}$$

Пусть заявка входит в сеть из i -го входного потока. Её маршрут в сети случаен, поэтому случайно и число приходов в j -ю СМО за время пребывания в сети. Среднее значение этого числа приходов будем называть передаточным коэффициентом. Он однозначно определяется для любых i, j , матрицей P вероятностей передач.

$$\left\{ \begin{array}{l} \lambda_1 = \alpha_{11} I_1 + \alpha_{21} I_2 + \dots + \alpha_{N1} I_N \\ \lambda_2 = \alpha_{12} I_1 + \alpha_{22} I_2 + \dots + \alpha_{N2} I_N \\ \cdot \\ \lambda_N = \alpha_{1N} I_1 + \alpha_{2N} I_2 + \dots + \alpha_{NN} I_N \end{array} \right. \quad (4)$$

Алгоритм вычисления матрицы $\|\alpha_{ij}\|$.

1) Составить уравнения баланса сети, включающие интенсивности I_1, \dots, I_N в буквенном виде.

2) Положить $k=1$.

3) Решить уравнения баланса для случая, когда $l_k=1$, остальные $l_i=0$. Полученные значения $\lambda_1, \dots, \lambda_N$ записать в k -ю строку матрицы передаточных коэффициентов.

4) Положить $k=k+1$.

5) Если $k < N$, перейти к шагу 3, иначе к шагу 6.

6) Конец.

Матрица $\|\alpha_{ij}\|$

$$\begin{bmatrix} 10 & 2 & 1 & 3 & 0,5 & 2,5 \\ 10 & 3 & 1 & 3 & 0,5 & 2,5 \\ 10 & 2 & 2 & 3 & 0,5 & 2,5 \\ 10 & 2 & 1 & 4 & 0,5 & 2,5 \\ 10 & 2 & 1 & 3 & 1,5 & 2,5 \\ 10 & 2 & 1 & 3 & 0,5 & 3,5 \end{bmatrix}$$

10. Средние входовые времена пребывания в сети.

Входовые средние времена F_1, \dots, F_N пребывания в сети. Величина F_j определяется как среднее время пребывания в сети заявки, поступающей из j -го входного потока

$$\begin{aligned} F_1 &= \bar{T}_{\text{np1}} + p_{12}F_2 + p_{13}F_3, \\ F_2 &= \bar{T}_{\text{np2}} + F_1, \\ F_3 &= \bar{T}_{\text{np3}} + F_1. \end{aligned} \quad (12)$$

$$F_i = \sum_{j=1}^N \alpha_{ij} \bar{T}_{\text{npj}}. \quad (13)$$

$$F_1 = 2.845$$

$$F_2 = 2.897$$

$$F_3 = 2.876$$

$$F_4 = 2.855$$

$$F_5 = 2.906$$

$$F_6 = 2.978$$

11. Абсолютные пропускные способности

Абсолютную пропускную способность по i -му входу A_i можно найти непосредственно по ее определению.

$$\begin{cases} A_1 \leq \frac{n_1}{t_{s_1} \alpha_{11}} \\ A_2 \leq \frac{n_2}{t_{s_2} \alpha_{12}} \\ \dots\dots\dots \\ A_N \leq \frac{n_N}{t_{s_N} \alpha_{1N}} \end{cases}$$

Величина A_i определится как минимум значений, стоящих в правых частях неравенств

$$A_i = \min_j \frac{n_j}{t_{s_j} \alpha_{ij}}$$

$$A_1 = 1.428$$

$$A_2 = 20$$

$$A_3 = 33$$

$$A_4 = 33$$

$$A_5 = 66$$

$$A_6 = 4$$

$$A_i = 1.428$$

12. Производительность Семо

$$\lambda_0 = \frac{M}{U}$$

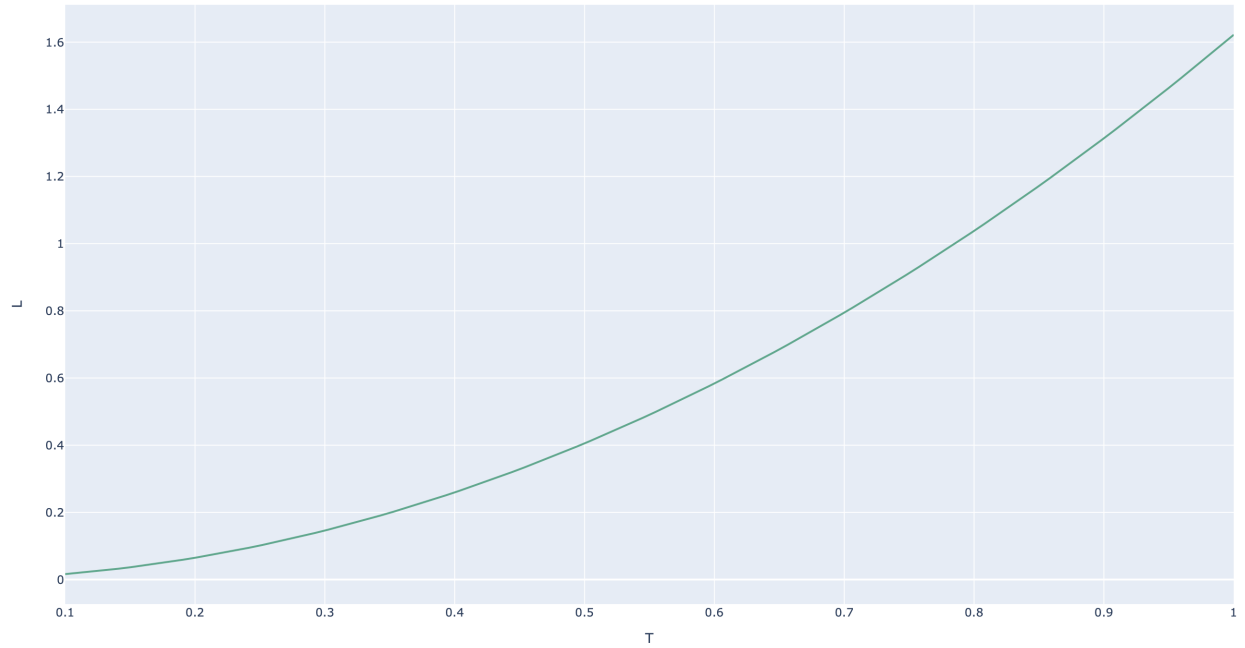
$$\lambda_0 = 1.6218$$

Графики

1. График зависимости среднего числа заявок на обслуживания в сети от времени ожидания

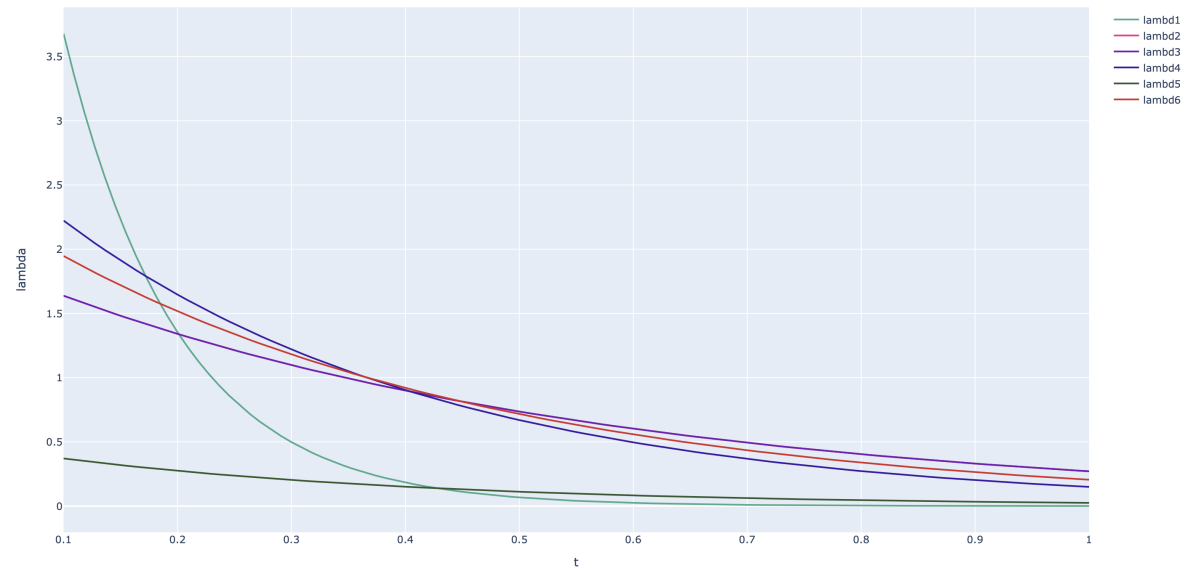
$$L = T_{оч} \cdot \lambda_0$$

График зависимости среднего числа заявок на обслуживания в сети от времени ожидания



2. График зависимости плотности распределение интенсивностей от времени

График зависимости плотности распределение интенсивностей от времени



3. График зависимости числа заявок от времени пребывания заявок

График зависимости числа заявок от времени пребывания заявок в сети

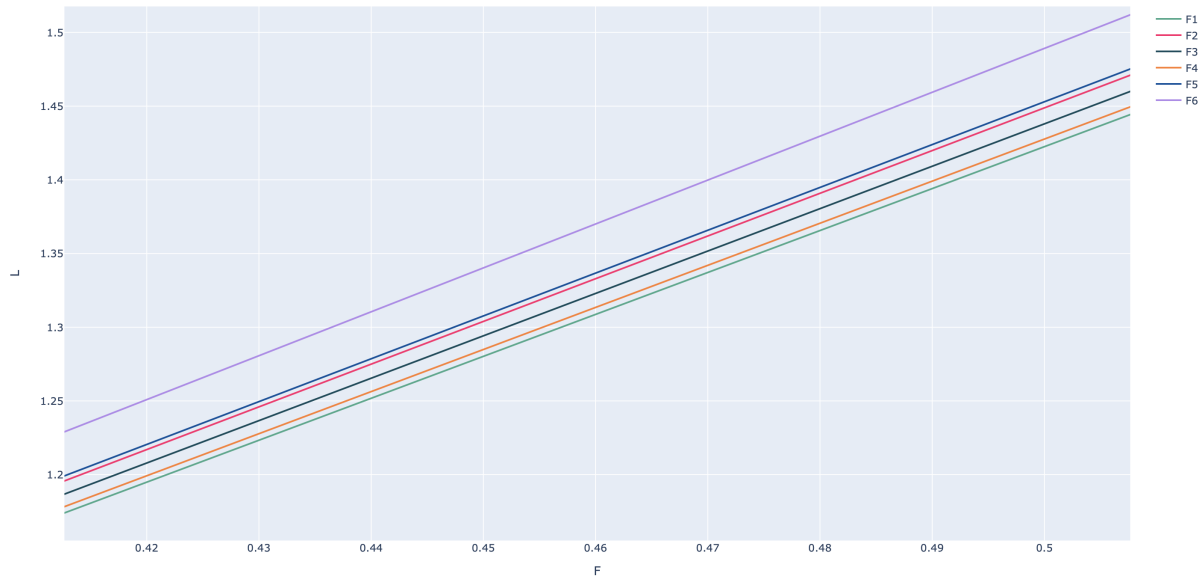
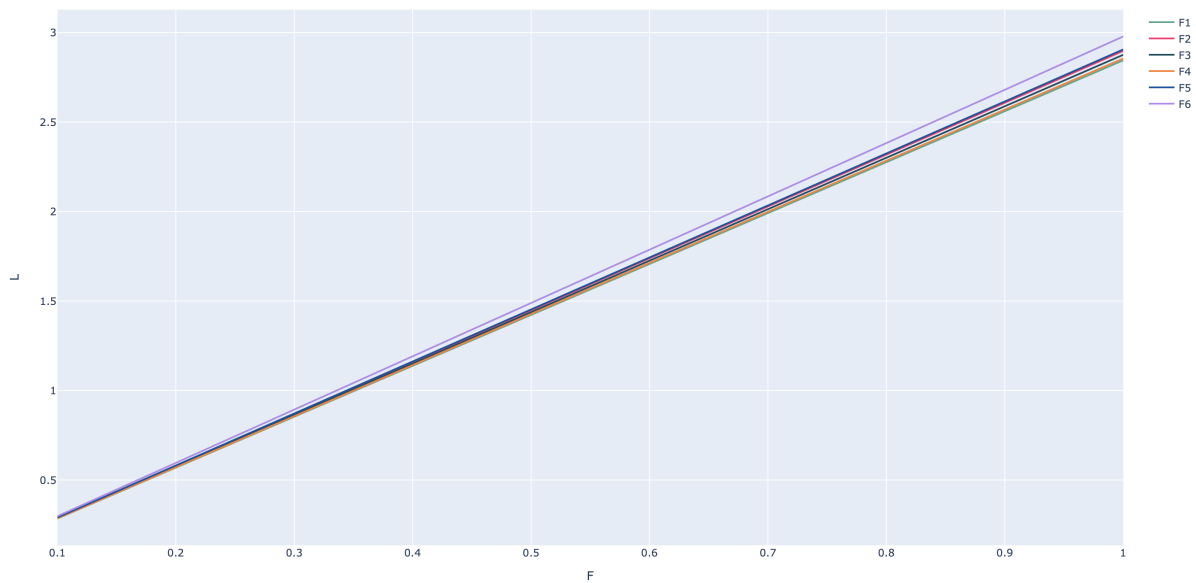
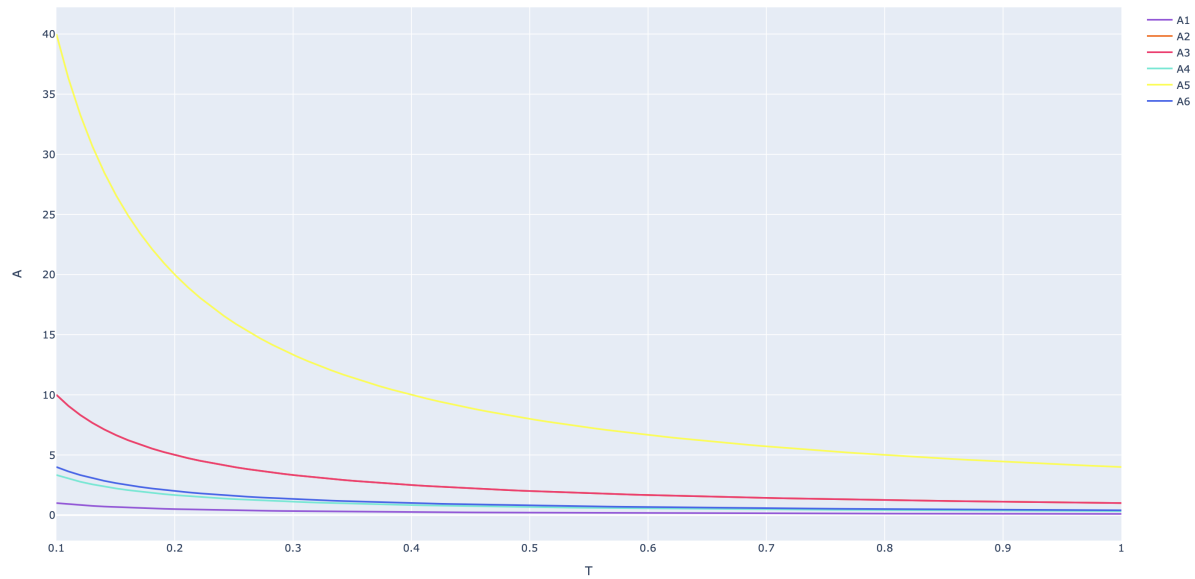


График зависимости числа заявок от времени пребывания заявок в сети



4. График зависимости абсолютной пропускной способности от среднего времени обслуживания заявок

График зависимости абсолютной пропускной способности от среднего времени обслуживания заявок



▼ Код

```
import numpy as np
import math
import plotly.graph_objects as go

# входные данные
N = 6
K1 = K3 = K4 = K6 = 1
K2 = K5 = 2
P10 = 0.1
P12 = 0.2
P13 = 0.1
P14 = 0.3
P15 = 0.05
P16 = 0.25
P21 = 1
P31 = 1
P41 = 1
P51 = 1
P61 = 1
print(P10+P12+P13+P14+P15+P16)
I = 1
Tobs1 = 0.07
Tobs2 = 0.05
Tobs3 = 0.03
Tobs4 = 0.01
Tobs5 = 0.06
Tobs6 = 0.1

# расчет баланса интенсивности
lambda1 = I / P10
lambda2 = P12 * lambda1
lambda3 = P13 * lambda1
lambda4 = P14 * lambda1
lambda5 = P15 * lambda1
lambda6 = P16 * lambda1

print("lambda1: ", lambda1)
print(I + lambda2+ lambda3+ lambda4+ lambda5+lambda6)
print("lambda: ", lambda2, lambda3, lambda4, lambda5, lambda6)
```

```

# проверка стационарности
ro1 = lambda1 * Tobs1
ro2 = (lambda2 * Tobs2)/2
ro3 = lambda3 * Tobs3
ro4 = lambda4 * Tobs4
ro5 = (lambda5 * Tobs5)/2
ro6 = lambda6 * Tobs6

print("ro: ", ro1, ro2, ro3, ro4, ro5, ro6)

# расчет локальных х-р семо
# средняя длина очереди в узле
L1 = ro1**2 / (1-ro1)
L2 = ro2**2 / (1-ro2)
L3 = ro3**2 / (1-ro3)
L4 = ro4**2 / (1-ro4)
L5 = ro5**2 / (1-ro5)
L6 = ro6**2 / (1-ro6)
print("L: ", L1, L2, L3, L4, L5, L6)
# среднее число заявок в узле
M1 = ro1 / (1-ro1)
M2 = ro2 / (1-ro2)
M3 = ro3 / (1-ro3)
M4 = ro4 / (1-ro4)
M5 = ro5 / (1-ro5)
M6 = ro6 / (1-ro6)
print("M: ", M1, M2, M3, M4, M5, M6)
# Средняя продолжительность пребывания заявки в очереди на обработку в узле
Toch1 = (Tobs1*ro1)/(1-ro1)
Toch2 = (Tobs2*ro2)/(1-ro2)
Toch3 = (Tobs3*ro3)/(1-ro3)
Toch4 = (Tobs4*ro4)/(1-ro4)
Toch5 = (Tobs5*ro5)/(1-ro5)
Toch6 = (Tobs6*ro6)/(1-ro6)
print("Toch: ", Toch1, Toch2, Toch3, Toch4, Toch5, Toch6)
# Среднее время пребывания заявки в СМО
Tsmo1 = Tobs1/(1-ro1)
Tsmo2 = Tobs2/(1-ro2)
Tsmo3 = Tobs3/(1-ro3)
Tsmo4 = Tobs4/(1-ro4)
Tsmo5 = Tobs5/(1-ro5)
Tsmo6 = Tobs6/(1-ro6)
print("Tsmo: ", Tsmo1, Tsmo2, Tsmo3, Tsmo4, Tsmo5, Tsmo6)
# нагрузка узла
b1 = Tsmo1*lambda1
b2 = Tsmo2*lambda2
b3 = Tsmo3*lambda3
b4 = Tsmo4*lambda4
b5 = Tsmo5*lambda5
b6 = Tsmo6*lambda6
print("b: ", b1, b2, b3, b4, b5, b6)
# коэффициент простоя
n1 = 1 - ro1
n2 = 1 - ro2
n3 = 1 - ro3
n4 = 1 - ro4
n5 = 1 - ro5
n6 = 1 - ro6
print("n: ", n1, n2, n3, n4, n5, n6)

# расчет х-р Семо
# суммарная нагрузка
B = b1+ b2+ b3+b4+b5+b6
print("B: ", B)
# среднее число заявок в сети
M = M1+M2+M3+M4+M5+ M6
print("M: ", M)
# среднее число заявок на обслуживание в сети
L = L1+L2+L3+L4+ L5+L6
print("L: ", L)
# среднее время ожидания в сети
Tocheredi = (1*(lambda1 * Toch1 + lambda2 * Toch2 + lambda3 * Toch3 + lambda4 * Toch4 + lambda5 * Toch5 + lambda6 * Toch6))/I
print("Tocheredi: ", Tocheredi)
# среднее время прибытие в сети

```

```

Tsemo = (1*(lambd1 * Tsmo1 + lambd2 * Tsmo2 + lambd3 * Tsmo3 + lambd4 * Tsmo4 + lambd5 * Tsmo5 + lambd6 * Tsmo6))/I
print("Tsemo: ", Tsemo)

# передаточные коэффициенты
Aarr= [[10, 2, 1, 3, 0.5, 2.5],
        [10, 3, 1, 3, 0.5, 2.5],
        [10, 2, 2, 3, 0.5, 2.5],
        [10, 2, 1, 4, 0.5, 2.5],
        [10, 2, 1, 3, 1.5, 2.5],
        [10, 2, 1, 3, 0.5, 3.5]]

# Средние входовые времена пребывания в сети.
F1 = 2.845
F2 = Tsmo2 + F1
F3 = Tsmo3 + F1
F4 = Tsmo4 + F1
F5 = Tsmo5 + F1
F6 = Tsmo6 + F1
print("F: ", F1, F2, F3, F4, F5, F6)

# Абсолютные пропускные способности
Absolut1 = K1/(Tobs1*10)
Absolut2 = K2/(Tobs2*2)
Absolut3 = K3/(Tobs3*1)
Absolut4 = K4/(Tobs4*3)
Absolut5 = K5/(Tobs5*0.5)
Absolut6 = K6/(Tobs6*2.5)
Ai = min(Absolut1, Absolut2, Absolut3, Absolut4, Absolut5, Absolut6)
print("Absolut: ", Absolut1, Absolut2, Absolut3, Absolut4, Absolut5, Absolut6)
print("Ai: ", Ai)

# производительность семо
Lambd0 = M/Tocheredi
print("Lambd0: ", Lambd0)

def line_1(x):
    return x * Lambd0

def plot_line_1():
    fig = go.Figure()
    x = np.linspace(0.1, 1, 90)

    fig.add_trace(go.Scatter( # name="",
        x=x,
        y=x * line_1(x),
        mode='lines',
        line_color='#43aa8b'))

    fig.update_layout(title='График зависимости среднего числа заявок на обслуживания в сети от времени ожидания')
    fig.update_traces(hoverinfo='x+y',
        hovertemplate="{y} ... при {x} ...")
    fig.update_yaxes(title="L")
    fig.update_xaxes(title="T")
    fig.update_layout(title_font={'size': 24, 'color': "#f94144", 'family': 'Sans-Serif'},
        showlegend=True)

    fig.show()

plot_line_1()

def line_2(a):
    return lambd1*np.exp(-lambd1*a)
def line_3(b):
    return lambd2*np.exp(-lambd2*b)
def line_4(c):
    return lambd3*np.exp(-lambd3*c)
def line_4(d):
    return lambd4*np.exp(-lambd4*d)
def line_5(e):
    return lambd5*np.exp(-lambd4*e)
def line_6(f):
    return lambd6*np.exp(-lambd6*f)
def plot_line_2():
    fig = go.Figure()

```

```

a = np.linspace(0.1, 1, 100)
b = np.linspace(0.1, 1, 100)
c = np.linspace(0.1, 1, 100)
d = np.linspace(0.1, 1, 100)
e = np.linspace(0.1, 1, 100)
f = np.linspace(0.1, 1, 100)

fig.add_trace(go.Scatter( name="lambd1",
    x=a,
    y=line_2(a),
    mode='lines',
    line_color='#43aa8b'))
fig.add_trace(go.Scatter( name="lambd2",
    x=b,
    y=line_3(b),
    mode='lines',
    line_color='#f72585'))

fig.add_trace(go.Scatter(name="lambd3",
    x=c,
    y=line_3(c),
    mode='lines',
    line_color='#7209b7'))
fig.add_trace(go.Scatter(name="lambd4",
    x=d,
    y=line_4(d),
    mode='lines',
    line_color='#3a0ca3'))
fig.add_trace(go.Scatter(name="lambd5",
    x=e,
    y=line_5(e),
    mode='lines',
    line_color='#31572c'))
fig.add_trace(go.Scatter(name="lambd6",
    x=f,
    y=line_6(f),
    mode='lines',
    line_color='#d62828'))

fig.update_layout(title='График зависимости плотности распределение интенсивностей от времени')
fig.update_traces(hoverinfo='x+y',
    hovertemplate="{y} ... при {x} ...")
fig.update_yaxes(title="lambda")
fig.update_xaxes(title="t")
fig.update_layout(title_font={'size': 24, 'color': "#f94144", 'family': 'Sans-Serif'},
    showlegend=True)

fig.show()

plot_line_2()

def line_7(g):
    return g * F1
def line_8(g1):
    return g1 * F2
def line_9(g2):
    return g2 * F3
def line_10(g3):
    return g3 * F4
def line_11(g4):
    return g4 * F5
def line_12(g5):
    return g5 * F6

def plot_line_3():
    fig = go.Figure()
    g = g1=g2=g3=g4=g5=np.linspace(0.1, 1,10)

    fig.add_trace(go.Scatter( name="F1",
        x=g,
        y=line_7(g),
        mode='lines',
        line_color='#43aa8b'))
    fig.add_trace(go.Scatter( name="F2",

```

```

        x=g1,
        y=line_8(g1),
        mode='lines',
        line_color='#ff206e'))
fig.add_trace(go.Scatter( name="F3",
        x=g2,
        y=line_9(g2),
        mode='lines',
        line_color='#0f4c5c'))
fig.add_trace(go.Scatter( name="F4",
        x=g3,
        y=line_10(g3),
        mode='lines',
        line_color='#fe7f2d'))
fig.add_trace(go.Scatter( name="F5",
        x=g4,
        y=line_11(g4),
        mode='lines',
        line_color='#00509d'))
fig.add_trace(go.Scatter( name="F6",
        x=g5,
        y=line_12(g5),
        mode='lines',
        line_color='#b388eb'))

fig.update_layout(title='График зависимости числа заявок от времени пребывания заявок в сети')
fig.update_traces(hoverinfo='x+y',
        hovertemplate="{y} ... при {x} ...")
fig.update_yaxes(title="L")
fig.update_xaxes(title="F")
fig.update_layout(title_font={'size': 24, 'color': "#f94144", 'family': 'Sans-Serif'},
        showlegend=True)

fig.show()

plot_line_3()

def line_13(h1):
    return K1/(h1*10)
def line_14(h2):
    return K2/(h2*2)
def line_15(h3):
    return K3/(h3*1)
def line_16(h4):
    return K4/(h4*3)
def line_17(h5):
    return K5/(h5*0.5)
def line_18(h6):
    return K6/(h6*2.5)
def plot_line_4():
    fig = go.Figure()
    h1=h2=h3=h4=h5=h6 = np.linspace(0.1, 1, 90)

    fig.add_trace(go.Scatter( name="A1",
        x=h1,
        y=line_13(h1),
        mode='lines',
        line_color='#9d4edd'))
    fig.add_trace(go.Scatter( name="A2",
        x=h2,
        y=line_14(h2),
        mode='lines',
        line_color='#ff6d00'))
    fig.add_trace(go.Scatter( name="A3",
        x=h3,
        y=line_15(h3),
        mode='lines',
        line_color='#ff206e'))
    fig.add_trace(go.Scatter( name="A4",
        x=h4,
        y=line_16(h4),
        mode='lines',
        line_color='#41ead4'))

```

```

fig.add_trace(go.Scatter( name="A5",
    x=h5,
    y=line_17(h5),
    mode='lines',
    line_color='#fbff12'))
fig.add_trace(go.Scatter( name="A6",
    x=h6,
    y=line_18(h6),
    mode='lines',
    line_color='#4361ee'))

fig.update_layout(title='График зависимости абсолютной пропускной способности от среднего времени обслуживания заявок')
fig.update_traces(hoverinfo='x+y',
    hovertemplate="%{y} ... при  %{x} ...")
fig.update_yaxes(title="A")
fig.update_xaxes(title="T")
fig.update_layout(title_font={'size': 24, 'color': "#f94144", 'family': 'Sans-Serif'},
    showlegend=True)

fig.show()

plot_line_4()

```