

Практическое занятие № 7

Тема: Расчет разомкнутых сетей массового обслуживания (СеМО) с многоканальными узлами и приоритетами.

Цель: Приобретение практических навыков расчета показателей оперативности обработки данных с помощью разомкнутых сетей массового обслуживания с многоканальными узлами и приоритетами.

Язык программирования, ПО и библиотеки: python 3.x, установленный пакет библиотек Anaconda или отдельно следующие: pandas, numpy, matplotlib. Среда разработки – PyCharm, Spyder 3 или Jupyter.

Порядок выполнения практического занятия:

1. Создайте копию файла pz7.py на локальном ПК.
2. Добавьте в папку с проектом файлы, находящиеся в каталоге ПЗ№7 (smo_im_prty.py, rand_distribution.py и др.).
3. Откройте файл с pz7.py. Ниже приведен его листинг. Запустите программу на выполнение

```
import rand_distribution as rd
import numpy as np
import matplotlib.pyplot as plt
import network_calc
from network_im_prty import NetworkPrty

def print_results(n, loads, k_num, v_im, v_ch):
    """
    Вывод результатов расчета
    nn:
    loads:
    k_num:
    v_im:
    v_ch:
    """
    ...

k_num = 3 # кол-во классов заявок
n_num = 5 # кол-во узлов сети
n = [3, 2, 3, 4, 3] # кол-во каналов обслуживания в узлах

coe_v_fix = 0.7 # коэфф вариации времени обслуживания в узлах (при построении зависимости от ro)
ro_fix = 0.7 # фиксированный коэфф загрузки в каждом узле (при построении зависимости от coe_v)

is_np = True # False - абсолютный приоритет, True - относительный

L = [0.9 / k_num] * k_num # интенсивности поступления заявок в сеть

jobs_num = 30000 # кол-во требуемых к обслуживанию заявок в ИМ сети

# списки для накопления времен пребывания в сети
```

```

vs_ch = [] # для численного расчета
vs_im = [] # для ИМ

for j in range(k_num):
    vs_ch.append([])
    vs_im.append([])

# коэфф загрузки узлов. Для построения зависимости от коэфф загрузки узлов в сети
roes = np.linspace(0.1, 0.9, 15)

# Матрицы интенсивностей переходов
R = []

for i in range(k_num):
    R.append(np.matrix([
        [1, 0, 0, 0, 0, 0],
        [0, 0.4, 0.6, 0, 0, 0],
        [0, 0, 0, 0.6, 0.4, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1]
    ]))

# зависимость от коэфф загрузки сети
for ro in roes:

    b = [] # список начальных моментов обслуживания в каждом из узлов по классам
    # [k][node][j], k - номер класса, node - номер узла, j - номер начального момента

    serv_params = [] # параметры распределения обслуживания [node][k]{ 'type': type, 'params': params}
    gamma_params = [] # параметры Гамма-распределения [node][mu, alpha]
    nodes_prty = [] # приоритеты по узлам сети [node][номера приоритетов по классам]

    for m in range(n_num):
        nodes_prty.append([])
        for j in range(k_num):
            nodes_prty[m].append(j)
            # if m == 4:
            #     nodes_prty[m].append(k_num - j - 1)

        b1 = ro * n[m] / sum(L)
        gamma_params.append(rd.Gamma.get_mu_alpha_by_mean_and_coev(b1, coev_fix))

        serv_params.append([])
        for i in range(k_num):
            serv_params[m].append({'type': 'Gamma', 'params': gamma_params[m]})

    for k in range(k_num):
        b.append([])
        for m in range(n_num):
            b[k].append(rd.Gamma.calc_theory_moments(*gamma_params[m], 4))

    # Задаем тип приоритета по узлам. NP - относительный, PR - абсолютный
    # В нашем случае тип приоритета в каждом узле одинаковый
    if is_np:
        prty = ['NP'] * n_num
    else:
        prty = ['PR'] * n_num

    # Задаем ИМ сети
    semo_im = NetworkPrty(k_num, L, R, n, prty, serv_params, nodes_prty)
    # Запускаем на расчет

```

```

semo_im.run(jobs_num)
# Получаем результаты - начальные моменты времени пребывания в сети
v_im = semo_im.v_semo

# Численный расчет сети
semo_calc = network_calc.network_prty_calc(R, b, n, L, prty, nodes_prty)
v_ch = semo_calc['v']
loads = semo_calc['loads']

for j in range(k_num):
    vs_im[j].append(v_im[j][0])
    vs_ch[j].append(v_ch[j][0])

# Вывод результатов расчета
print_results(n, loads, k_num, v_im, v_ch)

# строим график от го
fig, ax = plt.subplots()

for i in range(k_num):
    ax.plot(roes, vs_im[i], label="ИМ класс {0:d}".format(i + 1))
    ax.plot(roes, vs_ch[i], label="Числ класс {0:d}".format(i + 1))
ax.set_xlabel(r"$\rho$")
ax.set_ylabel(r"$\epsilon_{1}$")

if is_np:
    ax.set_title("Относительный приоритет")
else:
    ax.set_title("Абсолютный приоритет")

plt.legend()
plt.show()

coevs = np.linspace(0.3, 3.0, 15)

for coev in coevs:
    pass

# # строим график от coevs
...

```

4. После окончания имитационного моделирования (ИМ) и выполнения расчетов должен получиться график, похожий на представленный на рисунке 1.

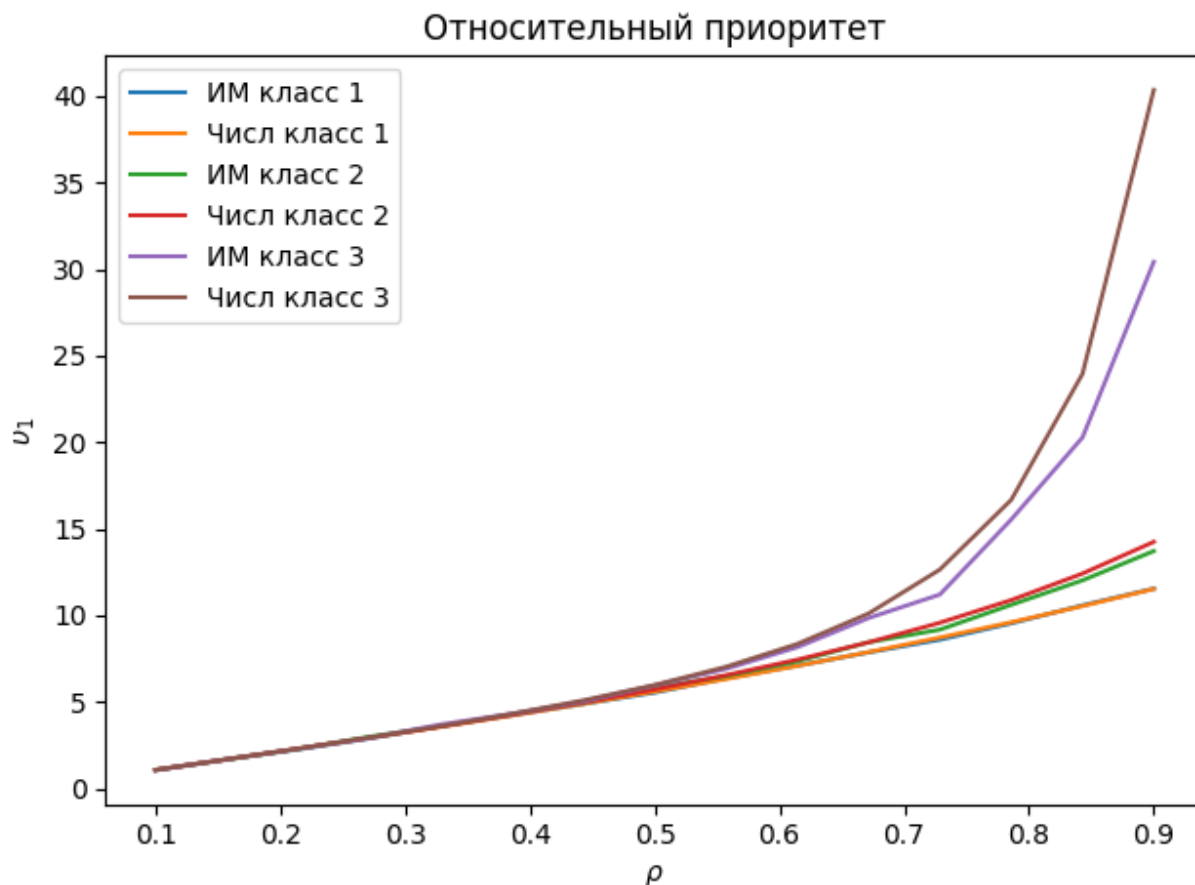


Рисунок 1. Результат ИМ и расчетов СеМО

На графике представлена зависимость средних времен пребывания для СеМО с тремя классами заявок. Данные зависимости получены в результате численных расчетов и ИМ, таким образом на рисунке шесть графиков. Структура данной сети (разомкнутой СеМО с относительным приоритетом) представлена на рисунке 2.

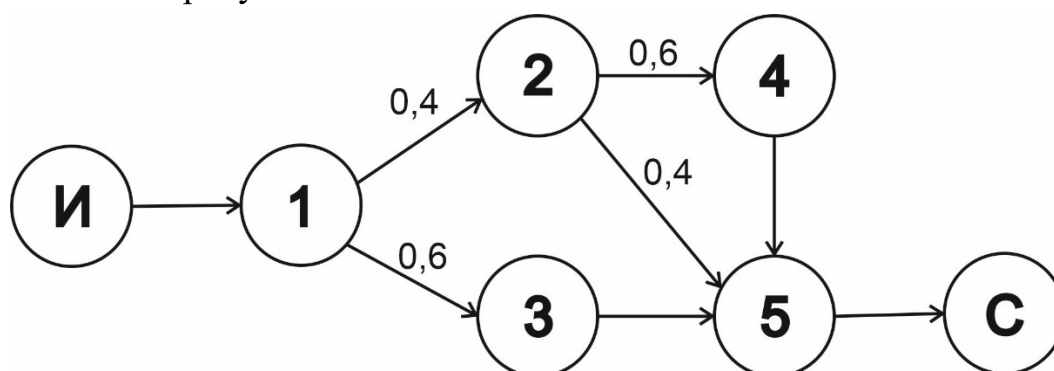


Рисунок 2. Структура СеМО

Узлы 0 и 6 являются истоком и стоком заявок. Узлы 1-5 представляют собой СМО с относительным приоритетом с 3, 2, 3, 4, 3 каналами

соответственно. Неподписанным ребрам графа соответствуют вероятности переходов, равные 1.0.

Структура СеМО задается матрицами переходов \mathbf{R}_k , где k – номер класса.

```
for i in range(k_num):
    R.append(np.matrix([
        [1, 0, 0, 0, 0, 0],
        [0, 0.4, 0.6, 0, 0, 0],
        [0, 0, 0, 0.6, 0.4, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1]
    ]))
```

Данная матрица представляет собой вероятности переходов заявок между соответствующими узлами. Например, $\mathbf{R}[1,1]=0.4$ и $\mathbf{R}[1,2]=0.6$ соответствуют вероятностям 0.4 и 0.6 переходов из узла 1 в узлы 2 и 3. Первая строка матрицы соответствует переходам из истока в узлы. Последний столбец матрицы – переходы в сток.

Считается, что в сети отсутствуют переходы в исток, поэтому первому столбцу матрицы соответствуют переходы в первый узел сети. Таким образом при количестве узлов сети равном M , размерность матрицы \mathbf{R} равна $(M+1, M+1)$.

Для каждого класса заявок матрица переходов может быть своя. В данном случае матрицы переходов для всех классов заявок одинаковые.

В каждом из узлов может быть задан любой произвольный тип и порядок приоритетов с помощью двумерного массива (списка)

```
nodes_prtty [номер узла][номер приоритета для класса]
```

По умолчанию задан порядок приоритетов, при котором меньшему номеру класса соответствует больший приоритет. Здесь n_num – число узлов СеМО, k_num – число классов заявок:

```
nodes_prtty = []

for m in range(n_num):
    nodes_prtty.append([])
    for j in range(k_num):
        nodes_prtty[m].append(j)
        # if m == 4:
        #     nodes_prtty[m].append(k_num - j - 1)
```

Если раскомментировать последние две строки, то 5-му узлу будет задан обратный порядок приоритетов. Сделайте это и попробуйте запустить расчет СеМО и ИМ. Как изменится среднее время пребывания заявок в СеМО? После окончания расчетов закомментируйте строки.

5. Постройте аналогичную зависимость для СеМО с абсолютным приоритетом. Для этого установите значение переменной *is_np=False* вверху текста программы.

```
is_np = True # True - относительный приоритет, False - абсолютный
```

6. Постройте зависимость среднего времени пребывания в СеМО с абсолютным и относительным приоритетом от коэффициента вариации. Для этого скопируйте код, расположенный после строки начала цикла по массиву коэффициентов загрузки

```
# зависимость от ro
for ro in roes:
    # код внутри цикла
```

внутри цикла по коэффициентам вариации вместо команды *pass*

```
for coev in coevs:
    pass
```

и измените содержимое так, чтобы ИМ и расчет осуществлялись для текущего значения *coev* при фиксированном коэффициенте загрузки *ro_fix*.

Значения *ro_fix* и *coev_fix*, а также число классов заявок следует определить из таблицы. Также необходимо изменить структуру сети (матрицы переходов и количество каналов в узлах). Структуры сети представлены на рисунках 3-7. Номер варианта соответствует номеру по журналу.

Вариант	ro_fix	coev_fix	Количество классов заявок	Номер варианта СеМО
1	0.7	0.7	2	1
2	0.75	0.9	3	2
3	0.8	1.1	4	3
4	0.85	1.3	2	4
5	0.9	1.4	3	5
6	0.7	2.1	4	1
7	0.75	0.75	2	2
8	0.8	1.8	3	3
9	0.85	0.85	4	4
10	0.9	1.9	3	5
11	0.72	0.76	2	1
12	0.79	0.92	3	2
13	0.81	1.2	4	3
14	0.86	1.35	2	4
15	0.93	1.42	3	5
16	0.74	2.14	4	1
17	0.78	0.76	2	2

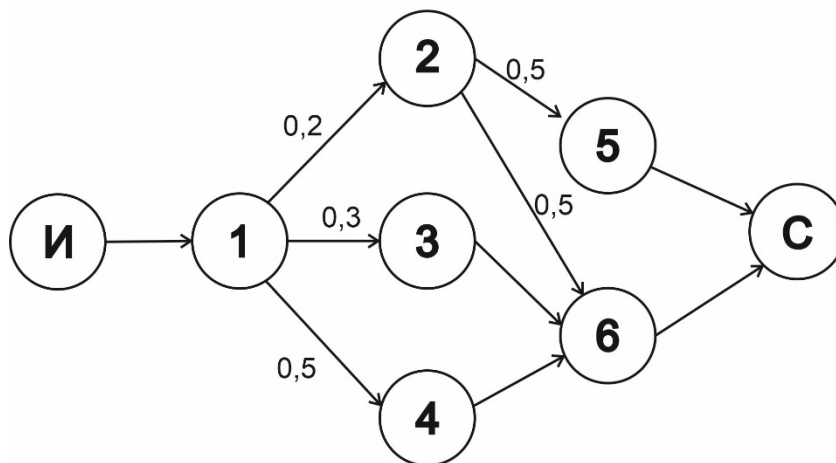


Рисунок 3. Вариант СеМО № 1.
Число каналов по узлам сети – 3, 2, 3, 4, 3, 3

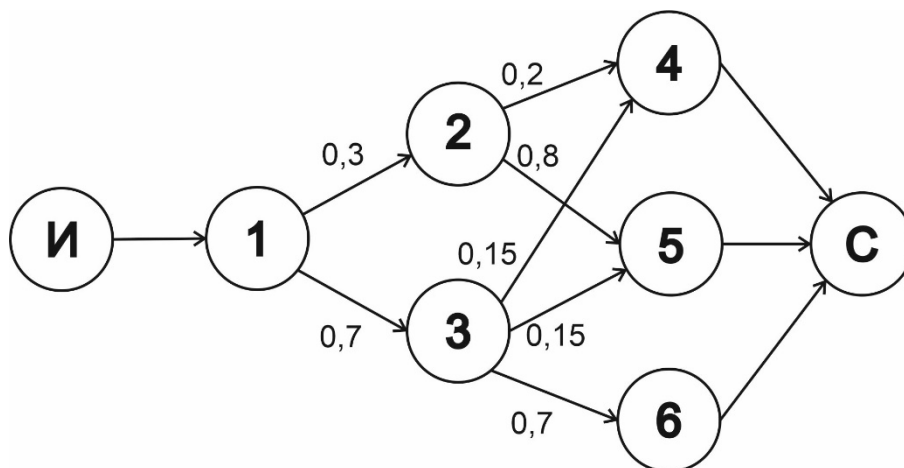


Рисунок 4. Вариант СеМО № 2.
Число каналов по узлам сети – 3, 2, 3, 2, 2, 3

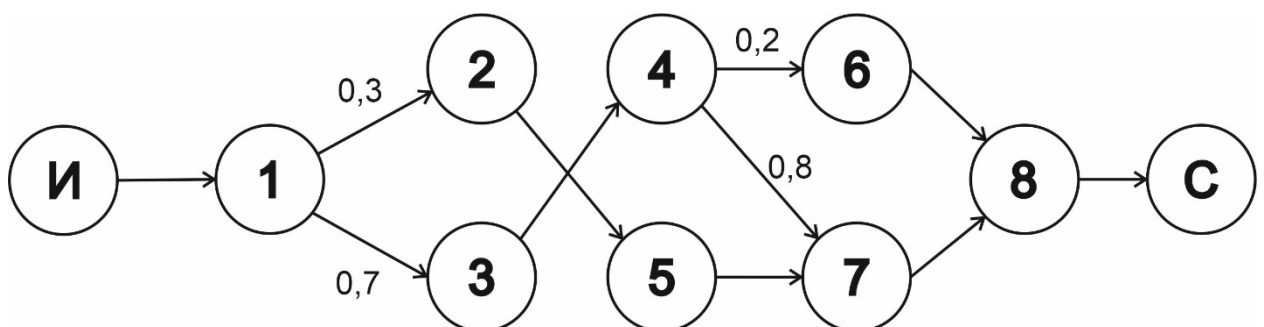


Рисунок 5. Вариант СеМО № 3.
Число каналов по узлам сети – 3, 2, 3, 3, 2, 3, 2, 3

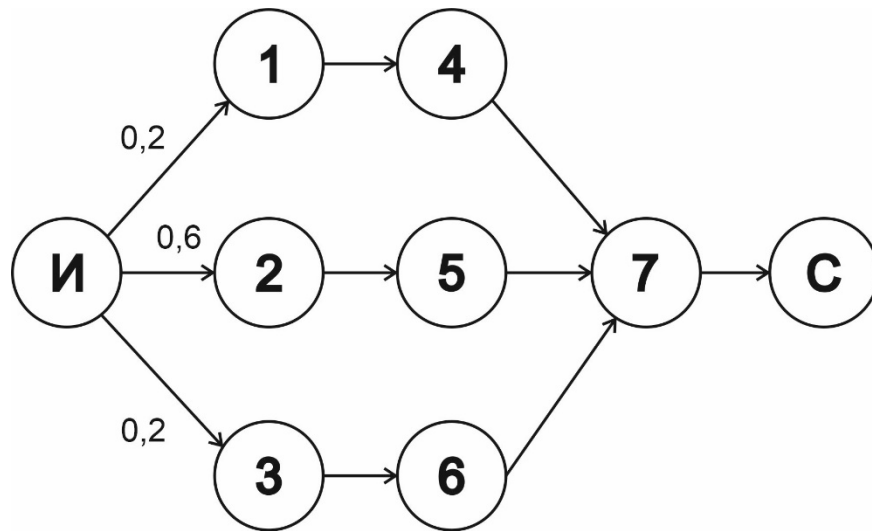


Рисунок 6. Вариант СеМО № 4.
Число каналов по узлам сети – 1, 1, 1, 1, 1, 1, 3

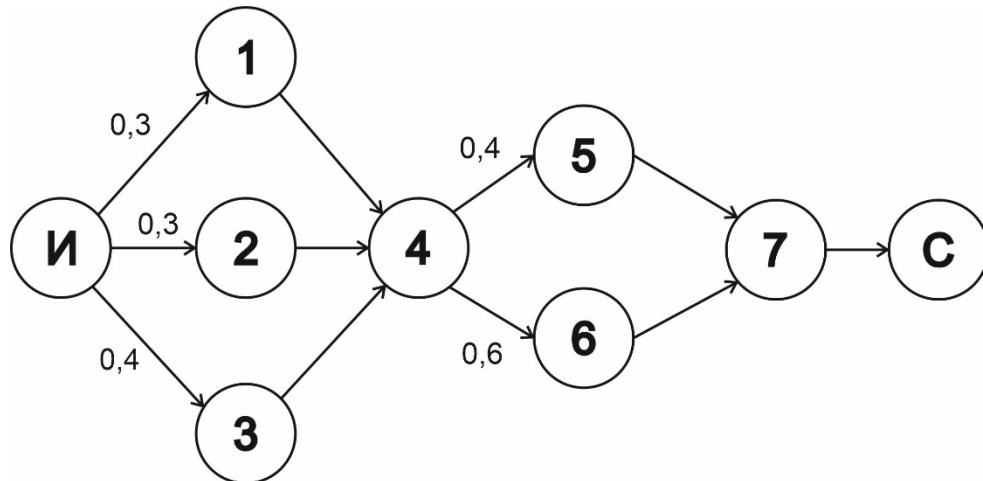


Рисунок 7. Вариант СеМО № 5.
Число каналов по узлам сети – 1, 2, 1, 3, 1, 2, 3

7. В ходе практического занятия необходимо построить зависимости среднего времени пребывания для заявок в СеМО от коэффициента вариации и коэффициента загрузки при относительном и абсолютном приоритете заявок в узлах. Итоговый отчет должен быть представлен в виде программы и графиков с результатами моделирования.

Будьте готовы ответить на **контрольные вопросы** по практическому занятию:

- 1) Каким образом устроена нотация Кендалла?
- 2) Какую модель СеМО вы исследовали? Чем отличаются разомкнутые и замкнутые СеМО?
- 3) Как зависит среднее время пребывания заявок в сети от коэффициента загрузки?

- 4) Как зависит среднее время пребывания заявок в сети от коэффициента вариации времени обслуживания?
- 5) Как зависит точность оценок, полученных с помощью ИМ, от числа обработанных заявок?
- 6) Какие виды приоритетных дисциплин обслуживания вы знаете? В чем их различие?
- 7) Каким образом введение приоритетов сказывается на показатели оперативности (среднем времени пребывания) в сети?
- 8) При каких значениях коэффициентов загрузки введение приоритетов имеет смысл?
- 9) Какие подходы к расчету СеМО вы знаете? В чем их достоинства и недостатки?
- 10) В каких случаях введение приоритетов является оправданным? Приведите примеры приоритетных дисциплин обслуживания из жизни, в военной деятельности.

Литература

1. Рыжиков Ю.И. Алгоритмический подход к задачам массового обслуживания: монография / Ю.И. Рыжиков. СПб.: ВКА им. А.Ф. Можайского, 2013. 496 с