МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МОЭВМ

ОТЧЕТ

по практической работе №1

по дисциплине «Теория принятия решений»

Тема: Принятие решений в матричных играх

Вариант 10

Студентка гр. 8303	________________	Самойлова А.С.
Преподаватель	________________	Попова Е.В.

Санкт-Петербург

2022

Цель работы

Найти решение задач матричных игр с нулевой суммой.

Порядок выполнения

1. С помощью инструментального средства определить границы выигрыша и наличие седловой точки для матрицы С1.
2. Графически и аналитически решить матричную игру 2×2 для матрицы С2.
3. Графически и аналитически решить матричную игру 2×N для матрицы С3.
4. Графически и аналитически решить матричную игру M×2 для матрицы С4.
5. С помошью симплекс-метода решить матричную игру M×N для матрицы С5.
6. Подсчитать относительную погрешность полученных результатов.

Выполнение практической

1. Матрица С1:

4	8	-1	-2
5	9	3	2
5	-7	-2	4

Для определения границ выигрыша и наличия седловой точки была написана программа на языке Python (Приложение А).

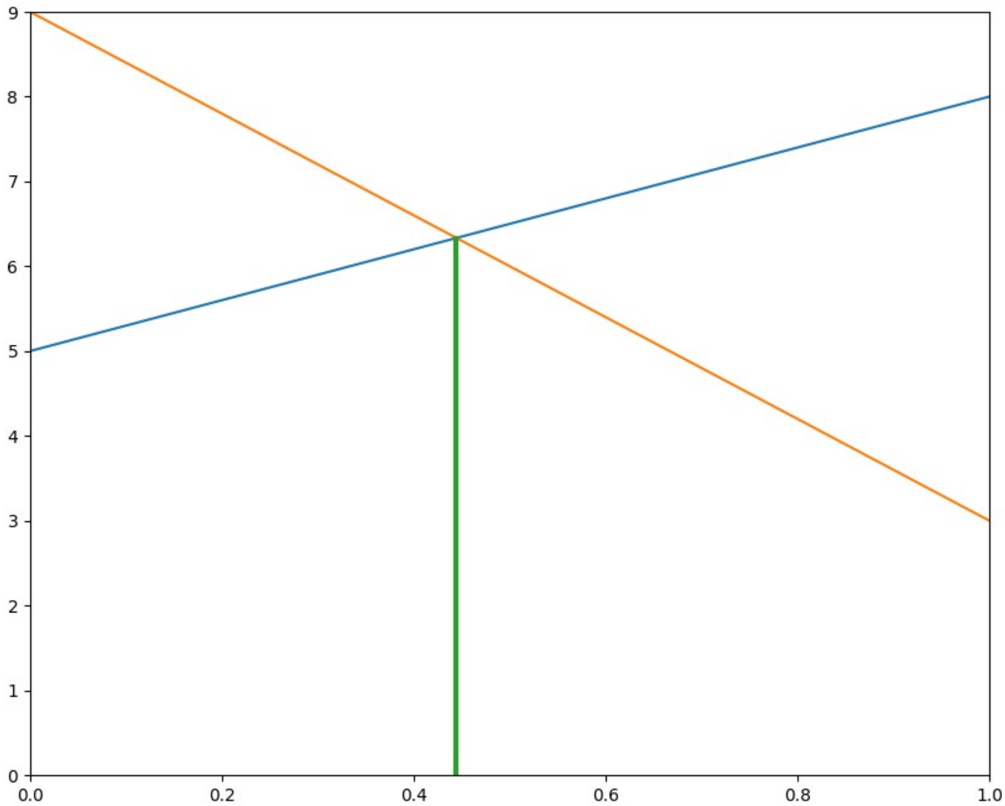
Результат работы программы:

Минимальное в строке 1 : 1
Минимальное в строке 2 : 2
Минимальное в строке 3 : 1
alfa : 2
Максимальное в столбце 1 : 5
Максимальное в столбце 2 : 6
Максимальное в столбце 3 : 5
beta : 5

2. Матрица C2:

5	9
8	3

Графическое решение:



Примерные значения:

- $V = 6.5$
- $p_1 = 0.55$
- $p_2 = 0.45$

Аналитическое решение:

$$C1=\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \quad S_A=\begin{pmatrix} A_1 & A_2 \\ p_1 & p_2 \end{pmatrix} \quad S_B=\begin{pmatrix} B_1 & B_2 \\ q_1 & q_2 \end{pmatrix}$$

$$\begin{cases} c_{11} p_1 + c_{21} p_2 = V \\ c_{12} p_1 + c_{22} p_2 = V \\ p_1 + p_2 = 1 \end{cases}$$

$$p_1 = \frac{c_{22} - c_{21}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{3-8}{5+3-(9+8)} = 0.556$$

$$p_2 = \frac{c_{11} - c_{12}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{5-9}{5+3-(9+8)} = 0.444$$

$$V = \frac{c_{22} c_{11} - c_{12} c_{21}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{2 \cdot 5 - 9 \cdot 8}{5+3-(9+8)} = 6.889$$

$$\begin{cases} c_{11} q_1 + c_{12} q_2 = V \\ c_{21} q_1 + c_{22} q_2 = V \\ q_1 + q_2 = 1 \end{cases}$$

$$q_1 = \frac{c_{22} - c_{12}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{3-9}{5+3-(9+8)} = 0.667$$

$$q_2 = \frac{c_{11} - c_{21}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{5-8}{5+3-(9+8)} = 0.333$$

$$V = \frac{c_{22} c_{11} - c_{12} c_{21}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{2 \cdot 5 - 9 \cdot 8}{5+3-(9+8)} = 6.889$$

Погрешности графического решения:

$$\delta(V) = \frac{|6.889 - 6.5|}{6.889} = 0.056$$

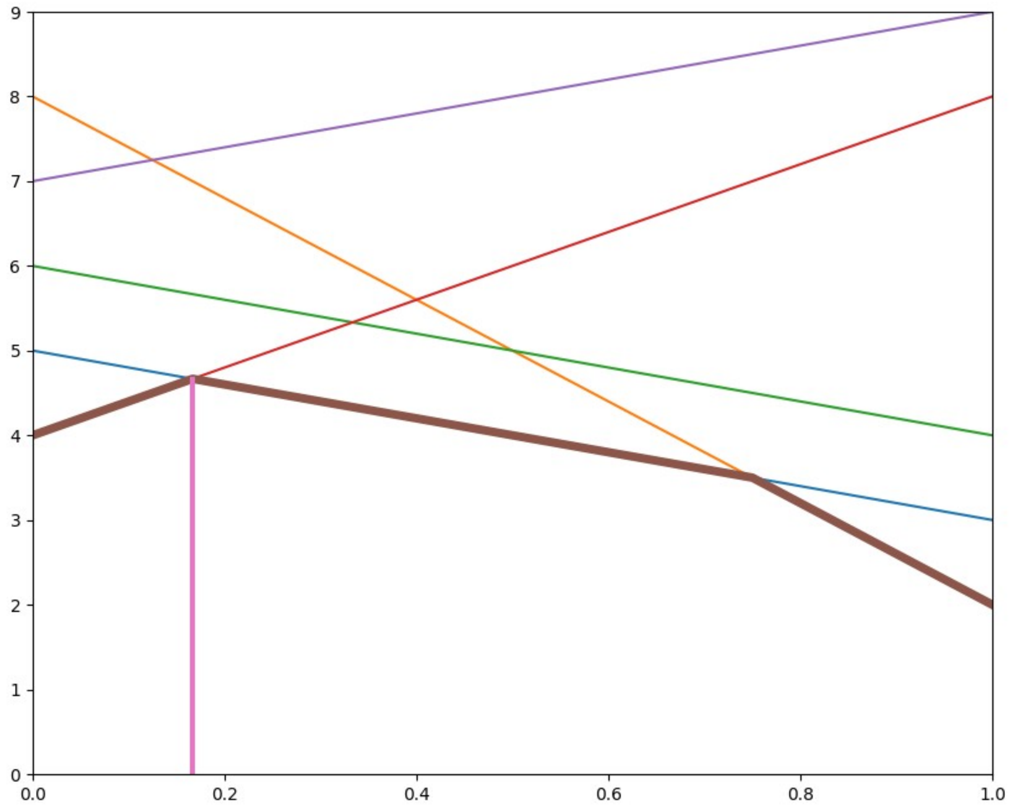
$$\delta(p_1) = \frac{|0.556 - 0.55|}{0.556} = 0.01$$

$$\delta(p_2) = \frac{|0.444 - 0.45|}{0.444} = 0.013$$

3. Матрица С3:

5	8	6	4	7
3	2	4	8	9

Графическое решение:



Примерные значения:

- V = 4.7
- $p_1 = 0.82$
- $p_2 = 0.18$

Аналитическое решение:

$$p_1 = \frac{c_{2k} - c_{2j}}{c_{1j} + c_{2k} - (c_{1k} + c_{2j})} = \frac{8-4}{5+8-(4+3)} = 0.833$$

$$p_2 = \frac{c_{1j} - c_{1k}}{c_{1j} + c_{2k} - (c_{1k} + c_{2j})} = \frac{5-4}{5+8-(4+3)} = 0.167$$

$$V = \frac{c_{2k} c_{1j} - c_{1k} c_{2j}}{c_{1j} + c_{2k} - (c_{1k} + c_{2j})} = \frac{8 \cdot 5 - 4 \cdot 3}{5+8-(4+3)} = 4.667$$

Погрешность графического решения:

$$\delta(V) = \frac{|4.667 - 4.7|}{4.667} = 0.007$$

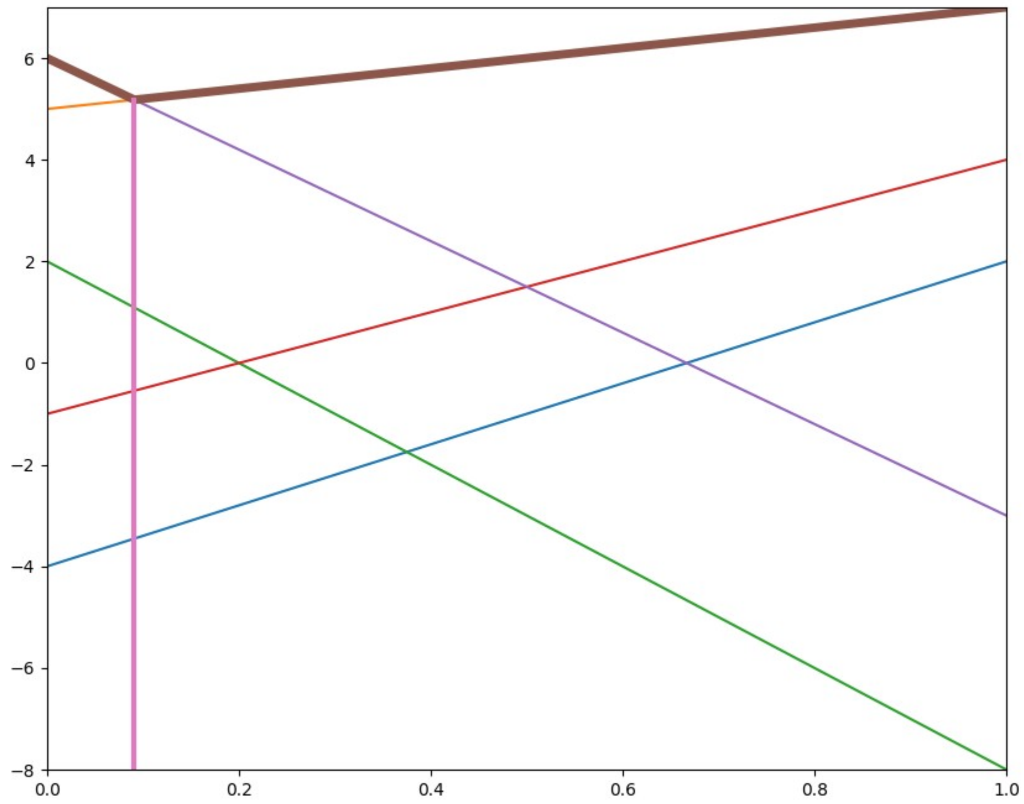
$$\delta(p_1) = \frac{|0.833 - 0.82|}{0.833} = 0.016$$

$$\delta(p_2) = \frac{|0.167 - 0.18|}{0.167} = 0.077$$

4. Матрица C4:

-4	2
5	7
2	-8
-1	4
6	-3

Графическое решение:



Примерные значения:

- $V = 5.2$
- $p_1 = 0.9$
- $p_2 = 0.1$

Аналитическое решение:

$$q_1=\frac{c_{k2}-c_{j2}}{c_{j1}+c_{k2}-(c_{j2}+c_{k1})}=\frac{7+3}{6+7-(-3+5)}=0.909$$

$$q_2=\frac{c_{j1}-c_{k1}}{c_{j1}+c_{k2}-(c_{j2}+c_{k1})}=\frac{6-5}{6+7-(-3+5)}=0.091$$

$$V=\frac{c_{k2}c_{j1}-c_{j2}c_{k1}}{c_{j1}+c_{k2}-(c_{j2}+c_{k1})}=\frac{7\cdot 6+3\cdot 5}{6+7-(-3+5)}=5.182$$

Погрешность графического решения:

$$\delta(V)=\frac{|5.182-5.2|}{5.182}=0.003$$

$$\delta(p_1)=\frac{|0.909-0.9|}{0.909}=0.0099$$

$$\delta(p_2)=\frac{|0.091-0.1|}{0.091}=0.098$$

5. Матрица C5:

2	6	4	5
7	2	3	1
5	3	6	2

$$\begin{cases} 2p_1+6p_2+4p_3+5p_4\geqslant V \\ 7p_1+2p_2+3p_3+1p_4\geqslant V \\ 5p_1+3p_2+6p_3+2p_4\geqslant V \\ 1p_1+1p_2+1p_3+1p_4=1 \end{cases}$$

$$\begin{cases} 2x_1+6x_2+4x_3+5x_4\geqslant 1 \\ 7x_1+2x_2+3x_3+1x_4\geqslant 1 \\ 5x_1+3x_2+6x_3+2x_4\geqslant 1 \\ 1x_1+1x_2+1x_3+1x_4=Z \end{cases}$$, где $x_i=\frac{p_i}{V}$, $Z=\frac{1}{V}$

Для решения задачи линейного программирования была написана программа на языке Python:

```
import numpy as np
from scipy.optimize import linprog

C5 = np.array([[2, 6, 4, 5],
               [7, 2, 3, 1],
               [5, 3, 6, 2]])

c = [1, 1, 1, 1]
A_ub = -C5
b_ub = [-1, -1, -1]
res = linprog(c=c, A_ub=A_ub, b_ub=b_ub, method='simplex')

print('V =', 1/res.fun)
print('p = ', res.x/res.fun)
```

Результат работы программы:

```
V = 4.148148148148149
p =  [0.40740741,  0.48148148,  0.11111111,  0]
```

Приложение А

```
if __name__ == '__main__':
    matrix = [[4,  8, -1, -2],
              [5,  9,  3,  2],
              [5, -7, -2,  4]]
    min_els = []
    for line in matrix:
        min_els.append(min(line))
        print('Минимальное в строке', len(min_els), ': ', min_els[-1])

    alfa = max(min_els)
    print('alfa : ', alfa)

    max_els = []
    for i in range(0, len(matrix[0])):
        max_els.append(matrix[0][i])
        for j in range(0, len(matrix)):
            if matrix[j][i] > max_els[-1]:
                max_els[-1] = matrix[j][i]
        print('Максимальное в столбце', len(max_els), ': ', max_els[-1])

    beta = min(max_els)
    print('beta : ', beta)
```