Type equation here.

Министерство образования Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет: Информатика и системы управления

Кафедра: Информационная безопасность (ИУ8)

Методы оптимизации

Лабораторная работа №6 на тему: «Матричные игры с нулевой суммой. Смешанные стратегии»

Вариант 5

Преподаватель:
Коннова Н.С.

Студент:
Девяткин Е.Д.

Группа:
ИУ8-34

Репозиторий работы: https://github.com/ledibonibell/MO-lab06

Москва 2023

Цель работы

Изучить постановку антагонистической игры двух лиц в нормальной форме; получить навыки нахождения решения игры в смешанных стратегиях (стратегическую седловую точку) за обоих игроков.

Постановка задачи

В общем случае игра двух игроков A и B с нулевой суммой записывается в виде матрицы стратегий:

Стратегии	b_1	b_2	$\dots$	b_n
a_1	c_{11}	c_{12}	$\dots$	c_{1n}
a_2	c_{21}	c_{22}	$\dots$	c_{2n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
a_n	c_{m1}	c_{m2}	$\dots$	c_{mn}

Минимальной гарантированный выигрыш игрока A называют нижней ценой игры. Максимально возможный проигрыш игрока B называют его верхней ценой игры

Ход работы

Рассмотрим нашу матрицу стратегий (таблица 1):

Стратегии	b_1	b_2	b_3	b_4
a_1	8	12	4	17
a_2	1	6	19	19
a_3	17	11	11	6
a_4	8	10	15	17
a_5	1	16	2	16

Таблица 1.

Игрок A

Найдем смешанные стратегии для игрока A. Для этого составим систему уравнений:

$$\begin{cases} 8x_1 + x_2 + 17x_3 + 8x_4 + x_5 \geq g \\ 12x_1 + 6x_2 + 11x_3 + 10x_4 + 16x_5 \geq g \\ 4x_1 + 19x_2 + 11x_3 + 15x_4 + 2x_5 \geq g \\ 17x_1 + 19x_2 + 6x_3 + 17x_4 + 16x_5 \geq g \end{cases}$$

$$x_1 + x_2 + x_3 + x_4 + x_5 = 1$$

где g - минимальный выигрыш игрока A

Разделим систему на функцию g:

$$\begin{cases} 8u_1 + u_2 + 17u_3 + 8u_4 + u_5 \geq 1 \\ 12u_1 + 6u_2 + 11u_3 + 10u_4 + 16u_5 \geq 1 \\ 4u_1 + 19u_2 + 11u_3 + 15u_4 + 2u_5 \geq 1 \\ 17u_1 + 19u_2 + 6u_3 + 17u_4 + 16u_5 \geq 1 \end{cases}$$

$$u_1 + u_2 + u_3 + u_4 + u_5 = {}^{1}/_{g}$$

Сформулируем задачу для решения симплекс-методом:

$$W = u_1 + u_2 + u_3 + u_4 + u_5 \rightarrow \min$$

$$\begin{cases} 8u_1 + u_2 + 17u_3 + 8u_4 + u_5 \geq 1 \\ 12u_1 + 6u_2 + 11u_3 + 10u_4 + 16u_5 \geq 1 \\ 4u_1 + 19u_2 + 11u_3 + 15u_4 + 2u_5 \geq 1 \\ 17u_1 + 19u_2 + 6u_3 + 17u_4 + 16u_5 \geq 1 \end{cases}$$

$$u_i \geq 0, i = 1,2,3,4,5$$

Рассмотрим решение симплекс методом (таблицы 1-4):

-8.00	-1.00	-17.00	-8.00	-1.00	1.00	0.00	0.00	0.00	-1.00
-12.00	-6.00	-11.00	-10.00	-16.00	0.00	1.00	0.00	0.00	-1.00
-4.00	-19.00	-11.00	-15.00	-2.00	0.00	0.00	1.00	0.00	-1.00
-17.00	-19.00	-6.00	-17.00	-16.00	0.00	0.00	0.00	1.00	-1.00
-1.00	-1.00	-1.00	-1.00	-1.00	0.00	0.00	0.00	0.00	0.00

Таблица 1.

Шаг 1:

0.47	0.06	1.00	0.47	0.06	-0.06	-0.00	-0.00	-0.00	0.06
-6.82	-5.35	0.00	-4.82	-15.35	-0.65	1.00	0.00	0.00	-0.35
1.18	-18.35	0.00	-9.82	-1.35	-0.65	0.00	1.00	0.00	-0.35
-14.18	-18.65	0.00	-14.18	-15.65	-0.35	0.00	0.00	1.00	-0.65
-0.53	-0.94	0.00	-0.53	-0.94	-0.06	0.00	0.00	0.00	0.06

Таблица 2.

Шаг 2:

0.43	0.00	1.00	0.43	0.01	-0.06	0.00	0.00	0.00	0.06
-2.75	0.00	0.00	-0.75	-10.86	-0.55	1.00	0.00	-0.29	-0.17
15.13	0.00	0.00	4.13	14.05	-0.30	0.00	1.00	-0.98	0.28
0.76	1.00	-0.00	0.76	0.84	0.02	-0.00	-0.00	-0.05	0.03
0.19	0.00	0.00	0.19	-0.15	-0.04	0.00	0.00	-0.05	0.09

Таблица 3.

Шаг 3:

0.42	0.00	1.00	0.43	0.00	-0.06	0.00	0.00	0.00	0.06
0.25	-0.00	-0.00	0.07	1.00	0.05	-0.09	-0.00	0.03	0.02
11.57	0.00	0.00	3.15	0.00	-1.01	1.29	1.00	-1.36	0.07
0.55	1.00	0.00	0.70	0.00	-0.02	0.08	0.00	-0.08	0.02
0.22	0.00	0.00	0.20	0.00	-0.03	-0.01	0.00	-0.05	0.09

Таблица 4.

Отсюда получим:

$$\begin{cases} u_1 = 0 \\ u_2 = 0{,}0218 \\ u_3 = 0{,}0566 \\ u_4 = 0 \\ u_5 = 0{,}0154 \end{cases} \quad \begin{cases} W = 0{,}0938 \\ g = 10{,}6594 \\ x_1 = 0 \\ x_2 = 0{,}2322 \\ x_3 = 0{,}6037 \\ x_4 = 0 \\ x_5 = 0{,}1641 \end{cases}$$

Таким образом, оптимальная смешанная стратегия игрока А имеет вид:

$$(0; 0{,}2322; 0{,}6037; 0; 0{,}1641)$$

Игрок В

Для нахождения смешанной стратегии игрока В составим систему уравнений:

$$\begin{cases} 8y_1 + 12y_2 + 4y_3 + 17y_4 \le h \\ y_1 + 6y_2 + 19y_3 + 19y_4 \le h \\ 17y_1 + 11y_2 + 11y_3 + 6y_4 \le h \\ 8y_1 + 10y_2 + 15y_3 + 17y_4 \le h \\ y_1 + 16y_2 + 2y_3 + 16y_4 \le h \end{cases}$$

$$y_1 + y_2 + y_3 + y_4 = 1$$

где h - максимальный проигрыш игрока В

Разделим систему на функцию h:

$$\begin{cases} 8v_1 + 12v_2 + 4v_3 + 17v_4 \le 1 \\ v_1 + 6v_2 + 19v_3 + 19v_4 \le 1 \\ 17v_1 + 11v_2 + 11v_3 + 6v_4 \le 1 \\ 8v_1 + 10v_2 + 15v_3 + 17v_4 \le 1 \\ v_1 + 16v_2 + 2v_3 + 16v_4 \le 1 \end{cases}$$

$$v_1 + v_2 + v_3 + v_4 = {}^{1}/_{h}$$

Сформулируем задачу для решения симплекс-методом:

$$Z = v_1 + v_2 + v_3 + v_4 \rightarrow \max$$

$$\begin{cases} 8v_1 + 12v_2 + 4v_3 + 17v_4 \le 1 \\ v_1 + 6v_2 + 19v_3 + 19v_4 \le 1 \\ 17v_1 + 11v_2 + 11v_3 + 6v_4 \le 1 \\ 8v_1 + 10v_2 + 15v_3 + 17v_4 \le 1 \\ v_1 + 16v_2 + 2v_3 + 16v_4 \le 1 \end{cases}$$

$$v_i \ge 0, i = 1,2,3,4$$

Рассмотрим решение симплекс методом (таблицы 5-8):

8.00	12.00	4.00	17.00	1.00	0.00	0.00	0.00	0.00	1.00
1.00	6.00	19.00	19.00	0.00	1.00	0.00	0.00	0.00	1.00
17.00	11.00	11.00	6.00	0.00	0.00	1.00	0.00	0.00	1.00
8.00	10.00	15.00	17.00	0.00	0.00	0.00	1.00	0.00	1.00
1.00	16.00	2.00	16.00	0.00	0.00	0.00	0.00	1.00	1.00
1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00

Таблица 5.

Шаг 1:

0.00	6.82	-1.18	14.18	1.00	0.00	-0.47	0.00	0.00	0.53
0.00	5.35	18.35	18.65	0.00	1.00	-0.06	0.00	0.00	0.94
1.00	0.65	0.65	0.35	0.00	0.00	0.06	0.00	0.00	0.06
0.00	4.82	9.82	14.18	0.00	0.00	-0.47	1.00	0.00	0.53
0.00	15.35	1.35	15.65	0.00	0.00	-0.06	0.00	1.00	0.94
0.00	0.35	0.35	0.65	0.00	0.00	-0.06	0.00	0.00	-0.06

Таблица 6.

Шаг 2:

0.00	2.75	-15.13	0.00	1.00	-0.76	-0.43	0.00	0.00	-0.19
0.00	0.29	0.98	1.00	0.00	0.05	-0.00	0.00	0.00	0.05
1.00	0.55	0.30	0.00	0.00	-0.02	0.06	0.00	0.00	0.04
0.00	0.75	-4.13	0.00	0.00	-0.76	-0.43	1.00	0.00	-0.19
0.00	10.86	-14.05	0.00	0.00	-0.84	-0.01	0.00	1.00	0.15
0.00	0.17	-0.28	0.00	0.00	-0.03	-0.06	0.00	0.00	-0.09

Таблица 7.

Шаг 3:

0.00	0.00	-11.57	0.00	1.00	-0.55	-0.42	0.00	-0.25	-0.22
0.00	0.00	1.36	1.00	0.00	0.08	-0.00	0.00	-0.03	0.05
1.00	0.00	1.01	0.00	0.00	0.02	0.06	0.00	-0.05	0.03
0.00	0.00	-3.15	0.00	0.00	-0.70	-0.43	1.00	-0.07	-0.20
0.00	1.00	-1.29	0.00	0.00	-0.08	-0.00	0.00	0.09	0.01
0.00	0.00	-0.07	0.00	0.00	-0.02	-0.06	0.00	-0.02	-0.09

Таблица 8.

Отсюда получим:

$$\begin{cases} v_1 = 0{,}0334 \\ v_2 = 0{,}0139 \\ v_3 = 0 \\ v_4 = 0{,}0465 \end{cases} \quad \begin{cases} Z = 0{,}0938 \\ h = 10{,}6594 \\ y_1 = 0{,}356 \\ y_2 = 0{,}1486 \\ y_3 = 0 \\ y_4 = 0{,}4954 \end{cases}$$

Таким образом, оптимальная смешанная стратегия игрока В имеет вид:

$$(0{,}356; 0{,}1486; 0; 0{,}4954)$$

Суммируя полученные вероятности получим единицы

Теперь рассмотрим цену игры и математическое ожидание:

Цена игры будет равна:

$$E(x^*, y^*) = \frac{1}{W} = \frac{1}{Z} = 10{,}6594$$

Математическое ожидание:

$$E(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_i y_j = 10{,}6594$$

И так получим конечный ответ:

$$x = (0; 0{,}2322; 0{,}6037; 0; 0{,}1641)$$

$$y = (0{,}356; 0{,}1486; 0; 0{,}4954)$$

$$E(x^*, y^*) = E(x, y) = 10{,}6594$$

Вывод

В ходе выполнения работы были проделаны расчеты с смешанных игр (матричных игр), в результате которых были получены оптимальные решения или вероятности стратегий для каждого игрока.

Суммируя эти вероятности мы получим единицу, что соответствует определению смешанной стратегии

Приложение А

Файл 'Main.py':

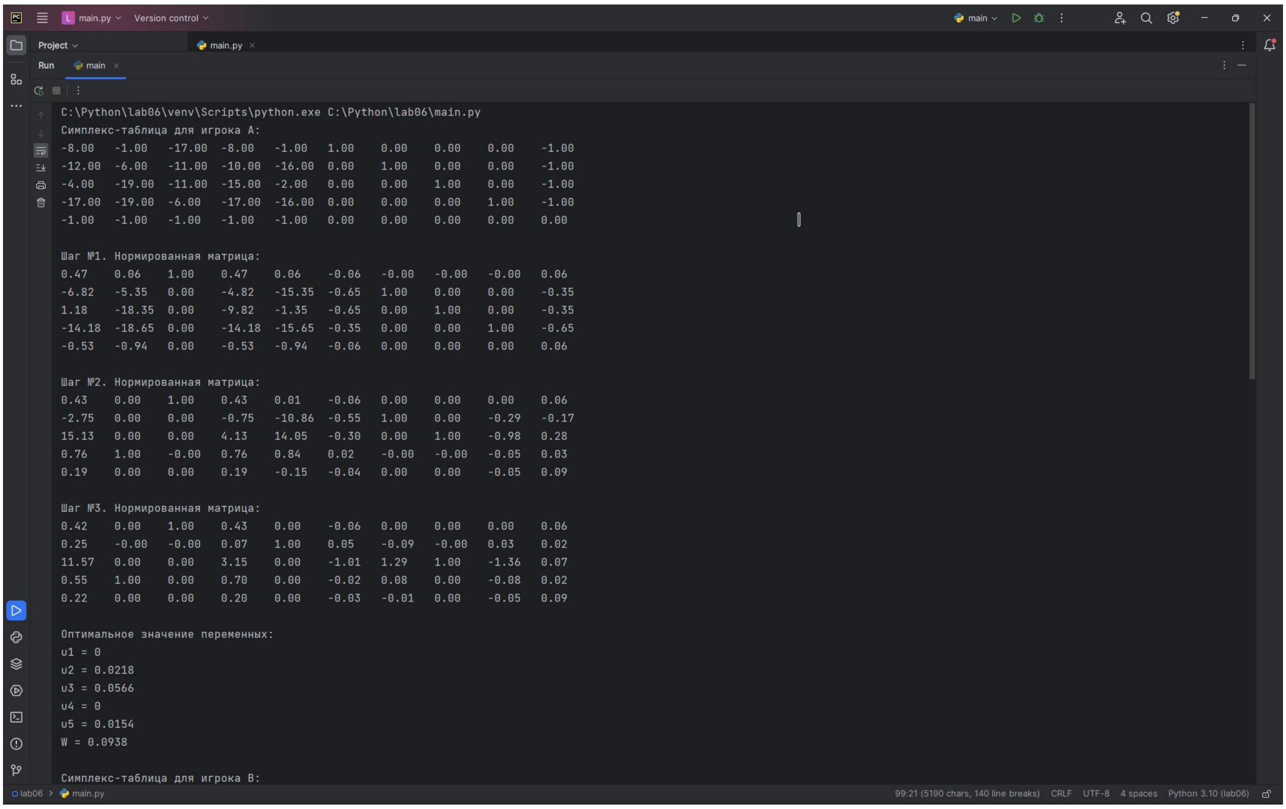
```
import numpy as np


def dummy_variable(self, params, function):
    A_extended = np.hstack((self, np.eye(self.shape[0])))
    c_extended = np.concatenate((function, np.zeros(self.shape[0])))
    A_extended = np.vstack((A_extended, c_extended))
    b_extended = np.append(params, 0)
    A_extended = np.column_stack((A_extended, b_extended))
    return A_extended


def norm_c(matrix):
    max_element_row_index = np.argmax(matrix[-1, :-1])
    max_element_column = np.max(matrix[:, max_element_row_index])
    max_element_column_index = np.argmax(matrix[:, max_element_row_index])

    matrix[max_element_column_index, :] /= max_element_column
    for i in range(matrix.shape[0]):
        if i != max_element_column_index:
            ratio = matrix[i, max_element_row_index] /
matrix[max_element_column_index, max_element_row_index]
            matrix[i, :] -= ratio * matrix[max_element_column_index, :]

    return matrix


def norm_b(matrix):
    max_element_column_index = np.argmin(matrix[:-1, -1])
    max_element_row_index = np.argmin(matrix[max_element_column_index, :-1])
    max_element_row = matrix[max_element_column_index,
max_element_row_index]

    matrix[max_element_column_index, :] /= max_element_row
    for i in range(matrix.shape[0]):
        if i != max_element_column_index:
            ratio = matrix[i, max_element_row_index] /
matrix[max_element_column_index, max_element_row_index]
            matrix[i, :] -= ratio * matrix[max_element_column_index, :]

    return matrix


def simplex_max(matrix):
```

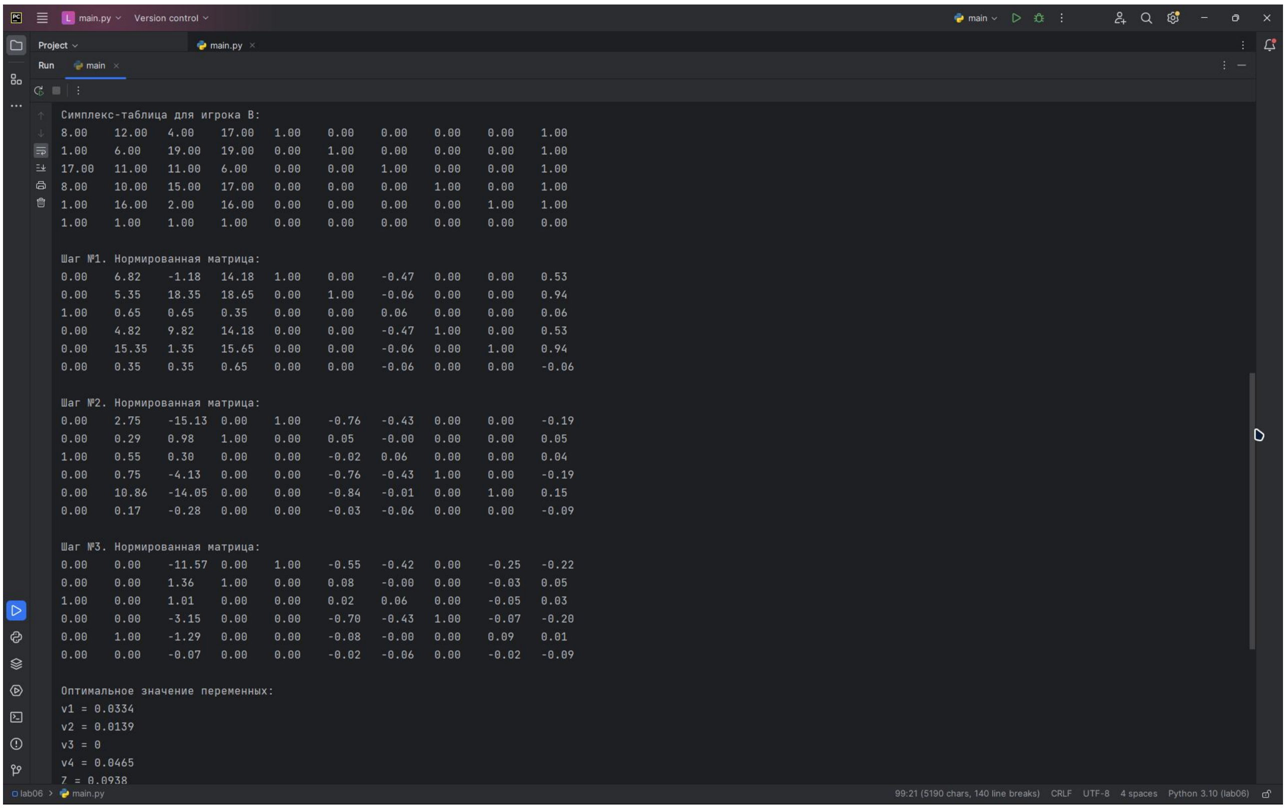
```
    i = 1
    while np.any(matrix[-1, :-1] > 0):
        norm_c(matrix)

        print(f"\nШаг №{i}. Нормированная матрица:")
        print_matrix(matrix)
        i += 1

    basic_variables = []
    for col in range(matrix.shape[1] - 1):
        if abs(np.all(matrix[:, col] == 0)) or abs(np.count_nonzero(matrix[:, col])) != 1:
            basic_variables.append(0)
        else:
            row_index = np.argmax(np.abs(matrix[:, col]))
            basic_variables.append(matrix[row_index, -1])

    optimal_value = matrix[-1, -1]
    return basic_variables, optimal_value


def simplex_min(matrix):
    i = 1
    while np.any(matrix[:-1, -1] < 0):
        norm_b(matrix)

        print(f"\nШаг №{i}. Нормированная матрица:")
        print_matrix(matrix)
        i += 1

    basic_variables = []
    for col in range(matrix.shape[1] - 1):
        if abs(np.all(matrix[:, col] == 0)) or abs(np.count_nonzero(matrix[:, col])) != 1:
            basic_variables.append(0)
        else:
            row_index = np.argmax(np.abs(matrix[:, col]))
            basic_variables.append(matrix[row_index, -1])

    optimal_value = matrix[-1, -1]
    return basic_variables, optimal_value


# def print_matrix(matrix):
#     for row in matrix:
#         rounded_row = [f"{val:.4f}" for val in row]
#         print(" || ".join(rounded_row))

def print_matrix(matrix):
```

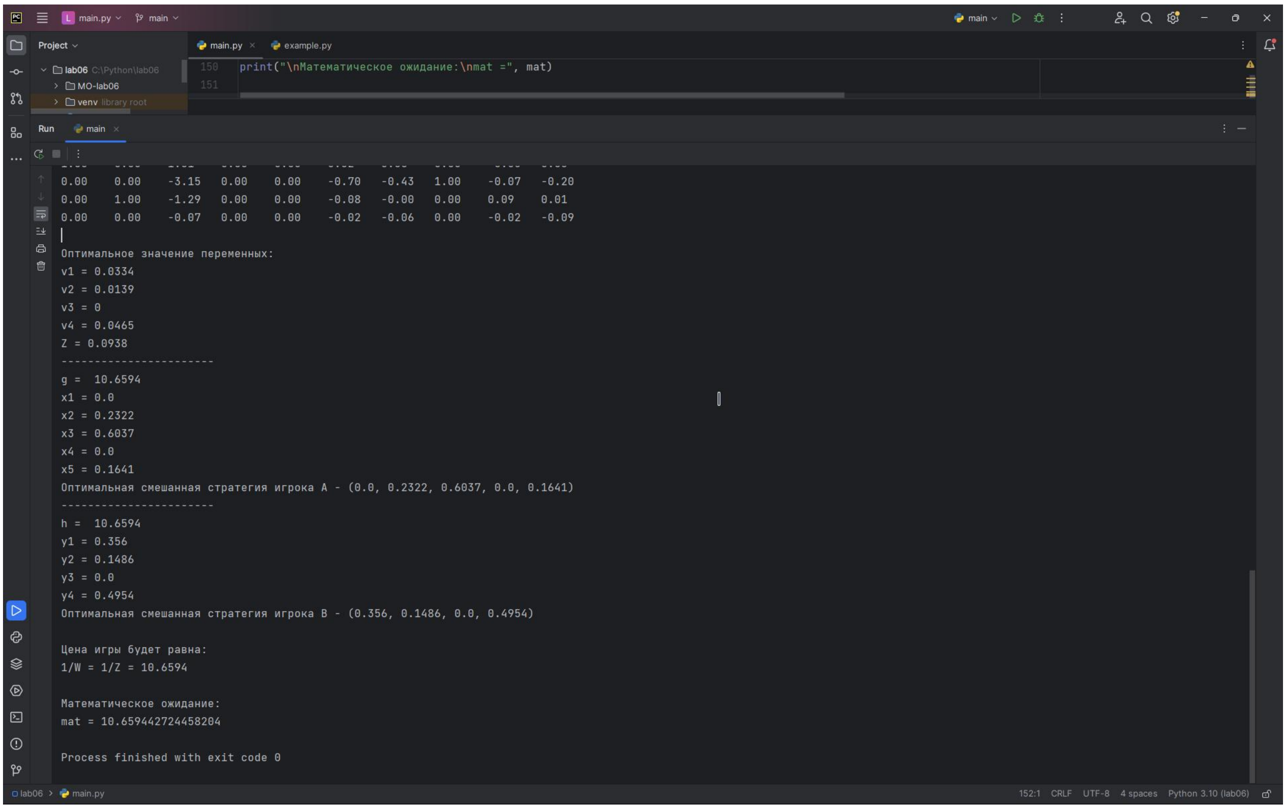
```
    for row in matrix:
        for value in row:
            print(f"{value:.2f}\t", end="")
        print()


A = np.array([
    [8, 12, 4, 17],
    [1, 6, 19, 19],
    [17, 11, 11, 6],
    [8, 10, 15, 17],
    [1, 16, 2, 16]
])

c = np.array([1, 1, 1, 1])
b = np.array([1, 1, 1, 1, 1])

print("Симплекс-таблица для игрока A:")
print_matrix(dummy_variable(-1 * np.transpose(A), -1 * c, -1 * b))

result_variables_A, result_value_A = simplex_min(dummy_variable(-1 *
np.transpose(A), -1 * c, -1 * b))

print("\nОптимальное значение переменных:")
print(f"u1 = {round(result_variables_A[0], 4)} \nu2 = {round(result_variables_A[1],
4)} \nu3 = {round(result_variables_A[2], 4)} \nu4 = {round(result_variables_A[3],
4)} \nu5 = {round(result_variables_A[4], 4)}")

print("W =", round(result_value_A, 4))

print("\nСимплекс-таблица для игрока B:")
print_matrix(dummy_variable(A, b, c))

result_variables_B, result_value_B = simplex_max(dummy_variable(A, b, c))

print("\nОптимальное значение переменных:")
print(f"v1 = {round(result_variables_B[0], 4)} \nv2 = {round(result_variables_B[1],
4)} \nv3 = {round(result_variables_B[2], 4)} \nv4 = {round(result_variables_B[3],
4)}")

print("Z =", round(-result_value_B, 4))


g = 1 / result_value_A
print("----------------------\ng = ", round(g, 4))
print(f"x1 = {round(g * result_variables_A[0], 4)} \nx2 = {round(g *
result_variables_A[1], 4)} \nx3 = {round(g * result_variables_A[2], 4)} \nx4 =
```

```
{round(g * result_variables_A[3], 4)} \nx5 = {round(g * result_variables_A[4], 4)}")
print(f"Оптимальная смешанная стратегия игрока A - ({round(g * result_variables_A[0], 4)}, {round(g * result_variables_A[1], 4)}, {round(g * result_variables_A[2], 4)}, {round(g * result_variables_A[3], 4)}, {round(g * result_variables_A[4], 4)})")

h = -1 / result_value_B
print("-----------------------\nh = ", round(h, 4))
print(f"y1 = {round(h * result_variables_B[0], 4)} \ny2 = {round(h * result_variables_B[1], 4)} \ny3 = {round(h * result_variables_B[2], 4)} \ny4 = {round(h * result_variables_B[3], 4)}")
print(f"Оптимальная смешанная стратегия игрока B - ({round(h * result_variables_B[0], 4)}, {round(h * result_variables_B[1], 4)}, {round(h * result_variables_B[2], 4)}, {round(h * result_variables_B[3], 4)})")

print("\nЦена игры будет равна:\n1/W = 1/Z =", round(g, 4))

mat = 0
for i in range(len(A)):
    for j in range(len(A[0])):
        mat += A[i][j] * g * result_variables_A[i] * h * result_variables_B[j]
print("\nМатематическое ожидание:\nmat =", mat)
```

Приложение Б




main.py
Version control
Project
main.py
Run
main
Симплекс-таблица для игрока B:
8.00 12.00 4.00 17.00 1.00 0.00 0.00 0.00 0.00 1.00
1.00 6.00 19.00 19.00 0.00 1.00 0.00 0.00 0.00 1.00
17.00 11.00 11.00 6.00 0.00 0.00 1.00 0.00 0.00 1.00
8.00 10.00 15.00 17.00 0.00 0.00 0.00 1.00 0.00 1.00
1.00 16.00 2.00 16.00 0.00 0.00 0.00 0.00 1.00 1.00
1.00 1.00 1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
Шаг №1. Нормированная матрица:
0.00 6.82 -1.18 14.18 1.00 0.00 -0.47 0.00 0.00 0.53
0.00 5.35 18.35 18.65 0.00 1.00 -0.06 0.00 0.00 0.94
1.00 0.65 0.65 0.35 0.00 0.00 0.06 0.00 0.00 0.06
0.00 4.82 9.82 14.18 0.00 0.00 -0.47 1.00 0.00 0.53
0.00 15.35 1.35 15.65 0.00 0.00 -0.06 0.00 1.00 0.94
0.00 0.35 0.35 0.65 0.00 0.00 -0.06 0.00 0.00 -0.06
Шаг №2. Нормированная матрица:
0.00 2.75 -15.13 0.00 1.00 -0.76 -0.43 0.00 0.00 -0.19
0.00 0.29 0.98 1.00 0.00 0.05 -0.00 0.00 0.00 0.05
1.00 0.55 0.30 0.00 0.00 -0.02 0.06 0.00 0.00 0.04
0.00 0.75 -4.13 0.00 0.00 -0.76 -0.43 1.00 0.00 -0.19
0.00 10.86 -14.05 0.00 0.00 -0.84 -0.01 0.00 1.00 0.15
0.00 0.17 -0.28 0.00 0.00 -0.03 -0.06 0.00 0.00 -0.09
Шаг №3. Нормированная матрица:
0.00 0.00 -11.57 0.00 1.00 -0.55 -0.42 0.00 -0.25 -0.22
0.00 0.00 1.36 1.00 0.00 0.08 -0.00 0.00 -0.03 0.05
1.00 0.00 1.01 0.00 0.00 0.02 0.06 0.00 -0.05 0.03
0.00 0.00 -3.15 0.00 0.00 -0.70 -0.43 1.00 -0.07 -0.20
0.00 1.00 -1.29 0.00 0.00 -0.08 -0.00 0.00 0.09 0.01
0.00 0.00 -0.07 0.00 0.00 -0.02 -0.06 0.00 -0.02 -0.09
Оптимальное значение переменных:
v1 = 0.0334
v2 = 0.0139
v3 = 0
v4 = 0.0465
Z = 0.0938
lab06 > main.py
99:21 (5190 chars, 140 line breaks) CRLF UTF-8 4 spaces Python 3.10 (lab06)


main.py
main
Project
main.py
example.py
lab06 C:\Python\lab06
MO-lab06
venv library root
print("\nМатематическое ожидание:\nmat =", mat)
Run
main
0.00 0.00 -3.15 0.00 0.00 -0.70 -0.43 1.00 -0.07 -0.20
0.00 1.00 -1.29 0.00 0.00 -0.08 -0.00 0.00 0.09 0.01
0.00 0.00 -0.07 0.00 0.00 -0.02 -0.06 0.00 -0.02 -0.09
Оптимальное значение переменных:
v1 = 0.0334
v2 = 0.0139
v3 = 0
v4 = 0.0465
Z = 0.0938
----
g = 10.6594
x1 = 0.0
x2 = 0.2322
x3 = 0.6037
x4 = 0.0
x5 = 0.1641
Оптимальная смешанная стратегия игрока A - (0.0, 0.2322, 0.6037, 0.0, 0.1641)
----
h = 10.6594
y1 = 0.356
y2 = 0.1486
y3 = 0.0
y4 = 0.4954
Оптимальная смешанная стратегия игрока B - (0.356, 0.1486, 0.0, 0.4954)
Цена игры будет равна:
1/W = 1/Z = 10.6594
Математическое ожидание:
mat = 10.659442724458204
Process finished with exit code 0
lab06 > main.py
152:1 CRLF UTF-8 4 spaces Python 3.10 (lab06)