МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3
по дисциплине «Теория принятия решений»
Тема: Принятие решений в матричных играх

Студентка гр. 1384	____________________	Усачева Д.В.
Преподаватель	____________________	Попова Е.В.

Санкт-Петербург
2025

Цель работы.

Найти решение задач в матричных играх.

Задание.

Графически и аналитически решить матричную игру 2 × N для матрицы C2.

Графически и аналитически решить матричную игру M × 2 для матрицы C3.

С помощью инструментальных средств решить матричную игру M × N для матрицы C4.

Подсчитать относительную погрешность результатов.

Биматричная игра задана матрицами A и B. Найти решение игры графическим методом. Подсчитать выигрыши.

Решить эту игру как матричную антагонистическую с нулевой суммой: решить графически и аналитически игру 2×2 для матрицы A, а затем для матрицы B. Сравнить результаты с биматричными.

Вариант 17.

$$C2 = \begin{pmatrix} 1.7 & -1 & 6.4 & 3.7 \\ 4.8 & 5 & 2 & 6.1 \end{pmatrix}; C3 = \begin{pmatrix} 1.7 & 5.3 \\ 3.4 & 7.9 \\ 8.6 & 3.1 \\ 2.8 & 7.5 \\ 9.3 & 4.5 \end{pmatrix}; C4 = \begin{pmatrix} -4 & 4 & -11 \\ 2 & 2 & -5 \\ 1 & -6 & 9 \end{pmatrix}$$

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix}; B = \begin{pmatrix} 2 & 0 \\ 2 & 6 \end{pmatrix}$$

Основные теоретические положения.

Теория игр – это математическая теория конфликтных ситуаций.

Игра – модель конфликтной ситуации.

Градация игр по неопределенности исходов:

- комбинаторные;
- азартные;
- стратегические.

Игра с нулевой суммой, если выигрыш одного игрока равен проигрышу другого.

Матричная игра – это игра с нулевой суммой, в которой задается выигрыш игрока 1 в виде матрицы.

Игра называется конечной, если у каждого игрока существует конечное число возможных стратегий.

Решить задачу по теории игр – это найти оптимальные стратегии игроков (чистые или смешанные) и выигрыш (результат игры).

Антагонистические игры

Определение: Пара $(x^0, y^0) \in X \times Y$ называется седловой точкой функции F, если

$$\forall x \in X, y \in Y \quad F(x, y^0) \leq F(x^0, y^0) \leq F(x^0, y) \quad (1)$$

или эквивалентно

$$\max_{x \in X} F(x, y^0) = F(x^0, y^0) = \min_{y \in Y} F(x^0, y)$$

Антагонистическая игра задается набором $\Gamma = \langle X, Y, F(x, y) \rangle$

Определение: Антагонистическая игра имеет решение, если $F(x, y)$ имеет седловую точку.

Если (x^0, y^0) – седловая точка,

то $\left(x^0, y^0, v = F(x^0, y^0)\right)$- решение игры, v- значение игры.

Лемма: Если $(x^0, y^0) \cdot (x^*, y^*)$ - 2 седловые точки, то

$$F(x^0, y^0) = F(x^*, y^*)$$

Доказательство: Если $F(x^*, y^*)$ - седловая точка, то

$$F(x, y^*) \le F(x^*, y^*) \le F(x^*, y)$$

$$F(x^*, y^*) \le F(x^*, y^0) \le F(x^0, y^0) \le F(x^0, y^*) \le F(x^*, y^*)$$

Определение: Антагонистическая игра называется матричной, если множества стратегий игроков конечны

$$X = \{1, \ldots, m\} \quad Y = \{1, \ldots, n\}$$

i – стратегия первого игрока, j – стратегия второго игрока.

$$A = \left(a_{ij}\right)_{m \times n}$$

Определение: Наилучший гарантированный результат для 1 игрока

$$\mathrm{V} = \sup_{x \in X} \inf_{y \in Y} F(x, y)$$

Стратегия x^0 называется максиминной, если

$$V = \inf_{y \in Y} F(x^0, y)$$

Наилучший гарантированный результат для 2 игрока

$$\mathrm{V} = \inf_{y \in Y} \sup_{x \in X} F(x, y)$$

Стратегия y^0 называется минимаксной, если

$$\mathrm{V} = \sup_{x \in X} F(x, y^0)$$

Лемма: В любой антагонистической игре Γ справедливо $\underline{v} \le \overline{v}$

Определение: Если в строке матрицы А элементы строки A_i не меньше соответствующих элементов строки A_k, а по крайней мере один строго больше, то строка A_i называется доминирующей, а строка A_k доминируемой.

При решении игры можно уменьшить размеры матрицы путем удаления из неё доминирующих столбцов и доминируемых строк.

Т1: в матричных играх существует равновесие по Нэшу в классе смешанных стратегий.

Т2: для того, чтобы ситуация $(\mathrm{x}^*, \mathrm{y}^*)$ была равновесием по Нэшу, необходимо и достаточно, чтобы для любых чистых стратегий $\mathrm{i} \in \mathrm{M}$ и $\mathrm{j} \in \mathrm{N}$, выполнялись условия:

$$\mathrm{H}_1(\mathrm{x}_\mathrm{i}, \mathrm{y}^*) \le \mathrm{H}_1(\mathrm{x}^*, \mathrm{y}^*)$$

$$\mathrm{H}_2(\mathrm{x}^*, \mathrm{y}_\mathrm{j}) \le \mathrm{H}_2(\mathrm{x}^*, \mathrm{y}^*)$$

Теорема фон Неймана: всякая матричная игра имеет решение в смешанных стратегиях.

Выполнение работы.

<u>Игра 2xN</u>

$$C2 = \begin{pmatrix} 1.7 & -1 & 6.4 & 3.7 \\ 4.8 & 5 & 2 & 6.1 \end{pmatrix} maxmin = 2$$

$$minmax = 4.8$$

$$minmax \neq maxmin$$

$$\text{нет седловой точки}$$

$$2 \leq V \leq 4.8$$

Можно уменьшить размерность матрицы путём удаления доминирующих столбцов. Здесь столбец $\begin{pmatrix} 3.7 \\ 6.1 \end{pmatrix}$ доминирует над $\begin{pmatrix} -1 \\ 5 \end{pmatrix}$, поэтому его можно убрать.

$$C2 = \begin{pmatrix} 1.7 & -1 & 6.4 \\ 4.8 & 5 & 2 \end{pmatrix}$$

Теперь можем провести графическое решение:

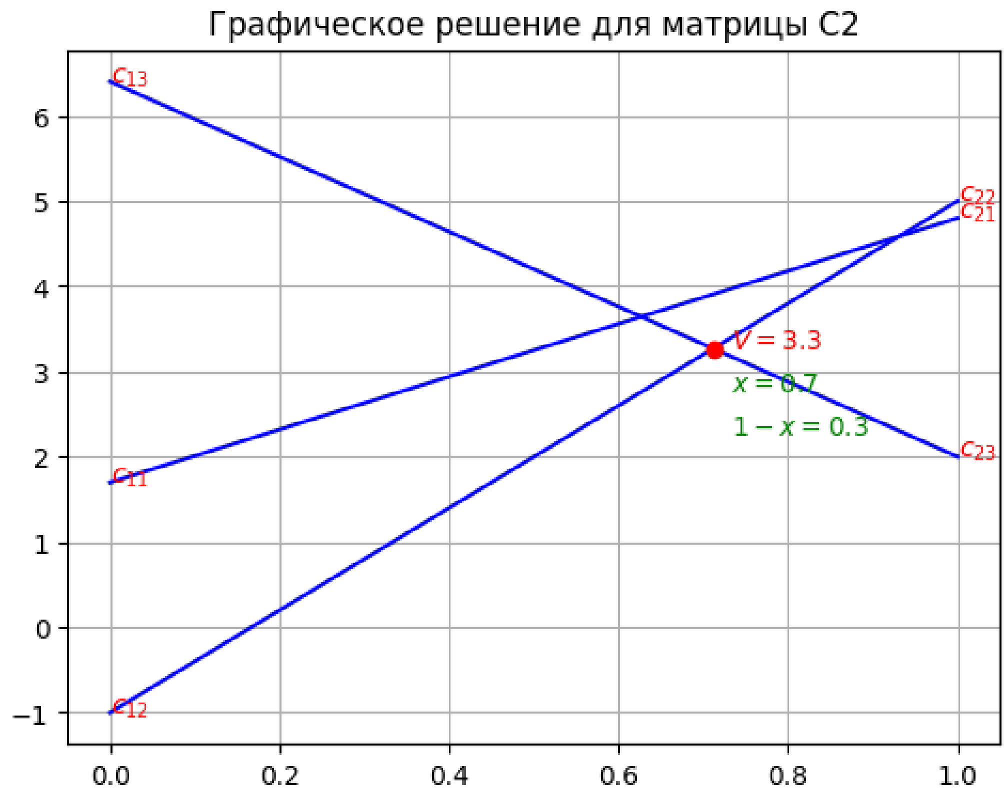

Рисунок 1 – Графическое решение матричной игры

По формулам вычислим следующие значения:

$$p_1 = \frac{c_{2j} - c_{2k}}{c_{1k} + c_{2j} - (c_{1j} + c_{2k})} = 0.2885$$

$$p_2 = \frac{c_{1k} - c_{1j}}{c_{1k} + c_{2j} - (c_{1j} + c_{2k})} = 0.7115$$

$$V = \frac{c_{1k}c_{2j} - c_{1j}c_{2k}}{c_{1k} + c_{2j} - (c_{1j} + c_{2k})} = 3.2692$$

Игра Mx2

$$C3 = \begin{pmatrix} 1.7 & 5.3 \\ 3.4 & 7.9 \\ 8.6 & 3.1 \\ 2.8 & 7.5 \\ 9.3 & 4.5 \end{pmatrix} maxmin = 4.5$$

$$minmax = 7.9$$

$$minmax \neq maxmin$$

нет седловой точки

$$4.5 \leq V \leq 7.9$$

Можно уменьшить размерность матрицы путём удаления доминируемых строк. $(1.7 \quad 5.3)$ и $(2.8 \quad 7.5)$ доминируемы строкой $(3.4 \quad 7.9)$, а строка $(8.6 \quad 3.1)$ доминируема строкой $(9.3 \quad 4.5)$.

$$C3 = \begin{pmatrix} 3.4 & 7.9 \\ 9.3 & 4.5 \end{pmatrix}$$

Теперь можем провести графическое решение:

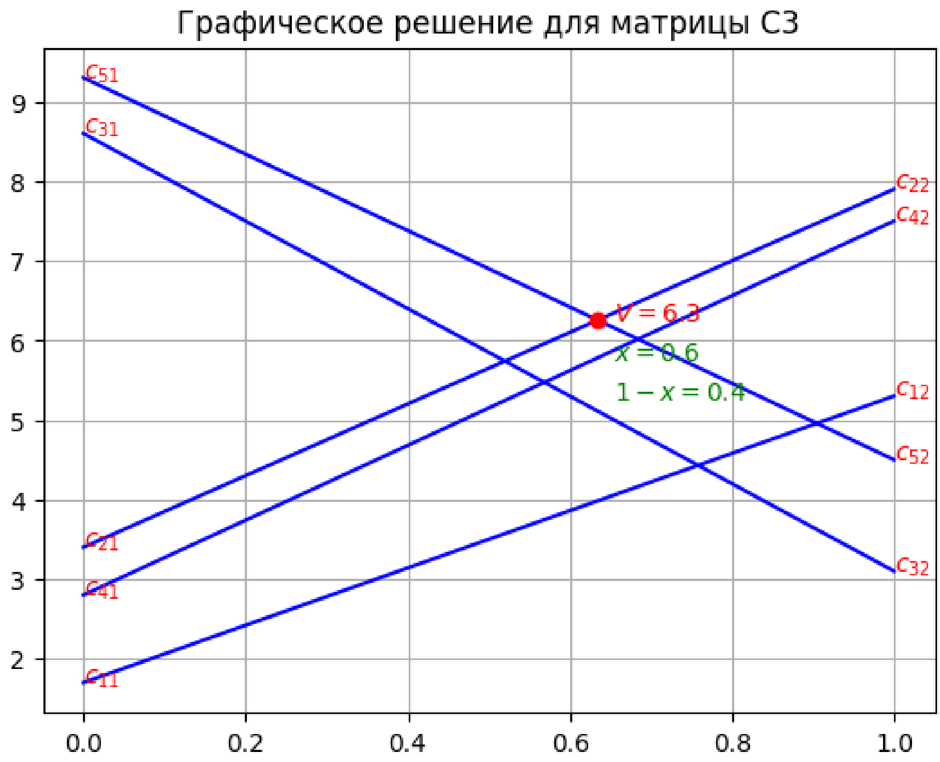

Рисунок 2 – Графическое решение матричной игры M x 2

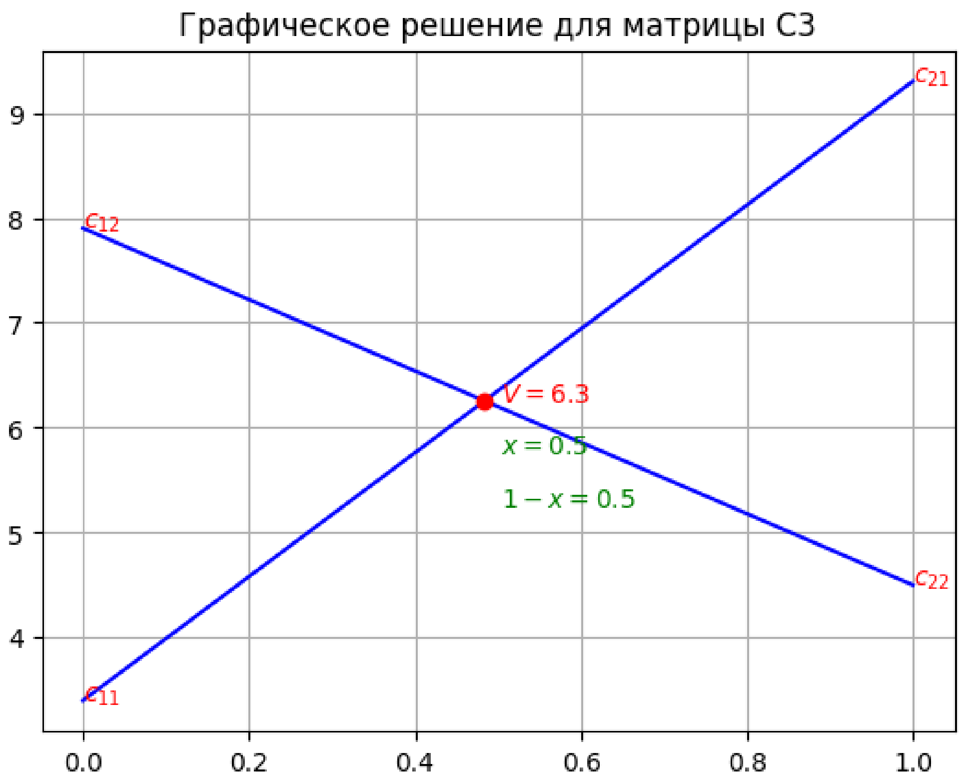

Рисунок 3 – Графическое решение матричной игры 2 x 2

Решение для второго игрока:

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

$$q_1 = \frac{c_{22} - c_{12}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{4.5 - 7.9}{3.4 + 4.5 - (7.9 + 9.3)} = 0.3655$$

$$q_2 = 0.6345$$

$$V = \frac{c_{22}c_{11} - c_{12}c_{21}}{c_{11} + c_{22} - (c_{12} + c_{21})} = \frac{4.5 \cdot 3.4 - 7.9 \cdot 9.3}{3.4 + 4.5 - (7.9 + 9.3)} = 6.2559$$

<u>Игра MxN</u>

$$C4 = \begin{pmatrix} -4 & 4 & -11 \\ 2 & 2 & -5 \\ 1 & -6 & 9 \end{pmatrix} maxmin = -5$$

$$minmax = 2$$

$$minmax \neq maxmin$$

$$\text{нет седловой точки}$$

$$-5 \leq V \leq 2$$

Размерность матрицы уменьшить не получится. Решим задачу симплекс методом.

Для первого игрока:

$$p_1 = 0, p_2 = \frac{15}{22}, p_3 = \frac{7}{22}$$

$$V = -\frac{6}{11}$$

Для второго игрока:

$$q_1 = 0, q_2 = \frac{7}{11}, q_3 = \frac{4}{11}$$

<u>Биматричная игра</u>

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix}; B = \begin{pmatrix} 2 & 0 \\ 2 & 6 \end{pmatrix}$$

Составим функции МО выигрышей:

$$H_1(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}\, x_i y_j$$

$$H_1(x, y) = (3 - 2 - 1 + 4)\text{xy} + (2 - 4)\text{x} + (1 - 4)\text{y} + 4 = 4\text{xy} - 2\text{x} - 3\text{y} + 4$$

$$H_2(x,y) = \sum\nolimits_{i=1}^{m} \sum_{j=1}^{n} b_{ij}\, x_i y_j$$

$$H_2(x,y) = (2-0-2+6)xy + (0-6)x + (2-6)y + 6 = 6xy - 6x - 4y + 6$$

Рассмотрим условие равновесия:

$$H_1(0,y) \le H_1(x,y)$$

$$4 - 3y \le 4\mathrm{xy} - 2\mathrm{x} - 3\mathrm{y} + 4 \Rightarrow\ 2\mathrm{x} \le 4\mathrm{xy}$$

$$H_1(1,y) \le H_1(x,y)$$

$$\mathrm{y} + 2 \le 4\mathrm{xy} - 2\mathrm{x} - 3\mathrm{y} + 4 \Rightarrow\ 2(2\mathrm{y} - 1) \le 2\mathrm{x}(2\mathrm{y} - 1)$$

$$H_2(x,0) \le H_2(x,y)$$

$$4\mathrm{y} \le 6\mathrm{xy}$$

$$H_2(x,1) \le H_2(x,y)$$

$$2 \le 6xy - 6x - 4y + 6 \Rightarrow\ 4(\mathrm{y} - 1) \le 6\mathrm{x}(\mathrm{y} - 1)$$

Следовательно:

Пусть $x = 0$: $y \le \frac{1}{2}$

Пусть $x = 1$: $\frac{1}{2} \le \mathrm{y}$

При $0 < x < 1$: $y = \frac{1}{2}$

Пусть $y = 0$: $\mathrm{x} \le \frac{2}{3}$

Пусть $y = 1$: $\frac{2}{3} \le \mathrm{x}$

При $0 < y < 1$: $x = \frac{2}{3}$

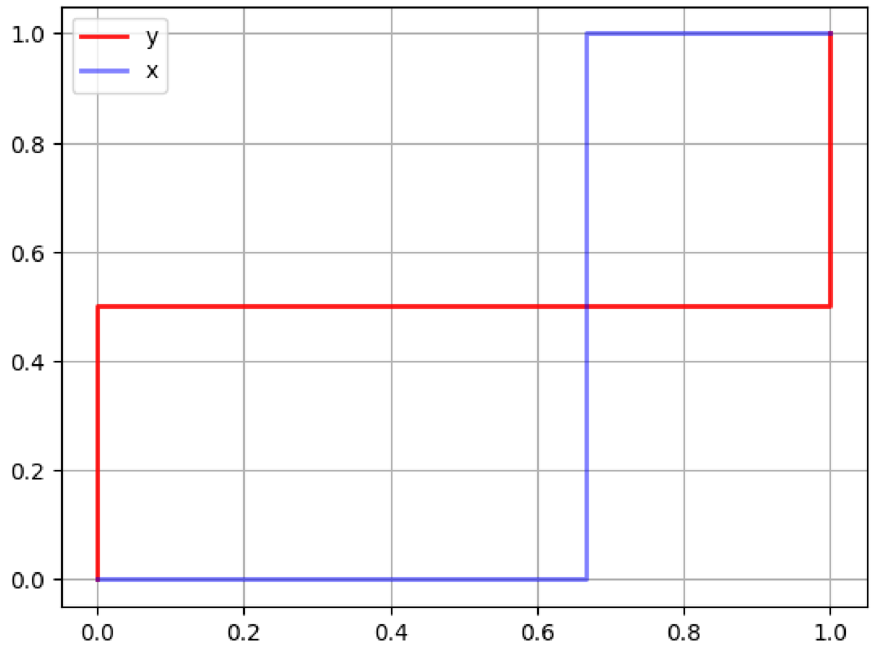

Рисунок 4 – Графическое решение биматричной игры

Области пересекаются в точке:

(x, y) = (0, 0) ⇒ $H_1(0,0) = 4$, $H_2(0,0) = 6$

(x, y) = (2/3, 1/2) ⇒ $H_1\left(\frac{2}{3},\frac{1}{2}\right) = \frac{5}{2}$, $H_2\left(\frac{2}{3},\frac{1}{2}\right) = 2$

(x, y) = (1, 1) ⇒ $H_1(1,1) = 3$, $H_2(1,1) = 2$

<u>Решим, как матричную антагонистическую с нулевой суммой</u>

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix} maxmin = 2$$

$$minmax = 3$$

$$minmax \neq maxmin$$

нет седловой точки

$$2 \leq V \leq 3$$

$$p_1 = \frac{3}{4}\ p_2 = \frac{1}{4}$$

$$q_1 = \frac{1}{2} q_2 = \frac{1}{2}$$

$$V = 2.5$$

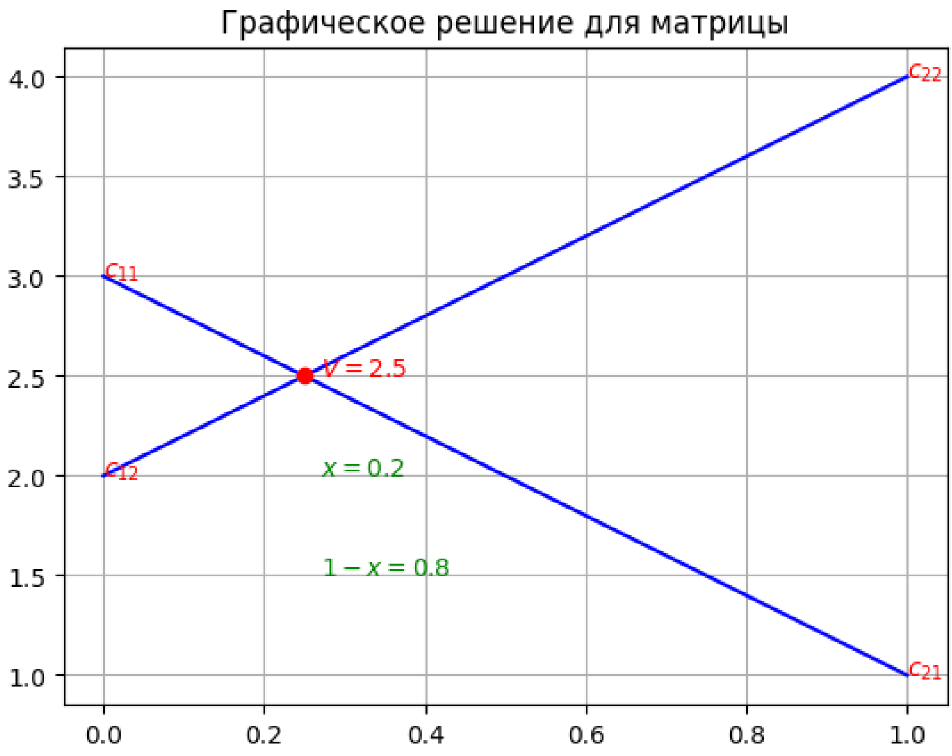

Рисунок 5 – Графическое решение матричной игры

$$B = \begin{pmatrix} 2 & 0 \\ 2 & 6 \end{pmatrix} maxmin = 2$$

$$minmax = 2$$

$$minmax = maxmin$$

есть седловая точка

$$V = 2$$

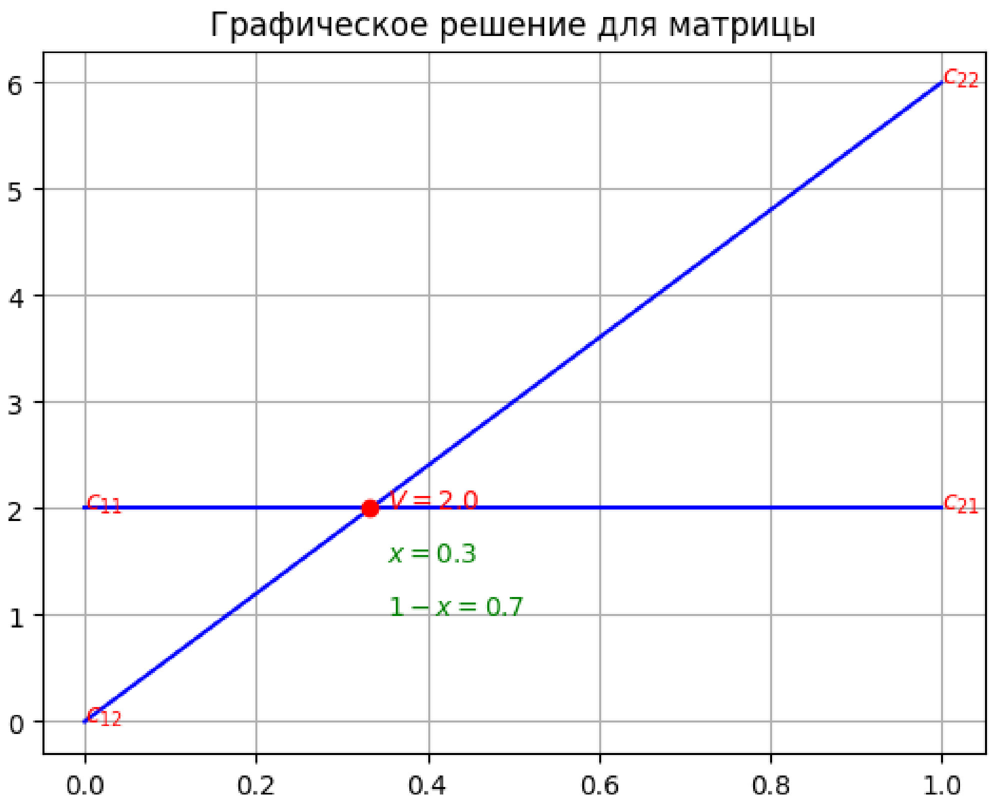

Рисунок 6 – Графическое решение матричной игры

Заметим, что результаты совпали с результатами биматричной игры при (x, y) = (2/3, 1/2).

Вывод.

В ходе выполнения данной работы были рассмотрены различные типы матричных игр, включая игры с нулевой суммой и биматричные игры, с целью нахождения оптимальных стратегий для игроков и значений игры. Для матриц C2 и C3 было установлено отсутствие седловых точек, и значения игры были определены в заданных пределах, с последующим уменьшением размерности матриц для упрощения задачи и нахождения оптимальных стратегий. Игра MxN для матрицы C4 была решена симплекс-методом, что позволило определить оптимальные стратегии и значение игры. В биматричной игре (матрицы A и B) найдено равновесие Нэша в смешанных стратегиях с выигрышами H1 = 2.5 и H2 = 2, что совпало с результатами решения игры как антагонистической с нулевой суммой для матриц A и B. Таким образом, работа подтвердила эффективность графических и аналитических методов для решения задач теории игр.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import linprog

def plot_solution(matrix, title):
    lines = []
    for i in range(matrix.shape[1]):
        c1i = matrix[0][i]
        c2i = matrix[1][i]
        plt.plot([0, 1], [c1i, c2i], color='blue')
        plt.text(1, c2i, f'$c_{{2{i+1}}}$', color='red')
        plt.text(0, c1i, f'$c_{{1{i+1}}}$', color='red')
        lines.append({'a': c2i - c1i, 'b': c1i})

    low_intersection = []
    for i in range(len(lines) - 1):
        for j in range(i + 1, len(lines)):
            a1, b1 = lines[i].values()
            a2, b2 = lines[j].values()
            if a1 == a2:
                continue
            x = (b2 - b1) / (a1 - a2)
            y = a1 * x + b1
            if 0 < x < 1 and (not low_intersection or y <
low_intersection[1]):
                low_intersection = (x, y)

    if low_intersection:
        x_inters, y_inters = low_intersection
        plt.scatter(x_inters, y_inters, color='red', zorder=5)
        plt.text(x_inters + 0.02, y_inters, f'$V={round(y_inters, 4)}$',
color='red')

    plt.title(title)
    plt.xlabel("Стратегии игрока")
    plt.ylabel("Выигрыш")
    plt.grid()
    plt.show()

def solve_task_1():
    print("=== Решение задачи 1: Матричная игра 2 × N ===")
    C2 = np.array([[1.7, -1, 6.4], [4.8, 5, 2]])
    print("Матрица C2:")
    print(C2)
    plot_solution(C2, "Графическое решение для матрицы C2")

    j, k = 1, 2
    c1, c2 = C2[0], C2[1]
    p1 = (c2[j] - c2[k]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    p2 = (c1[k] - c1[j]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    V = (c1[k] * c2[j] - c1[j] * c2[k]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    print(f"Вероятности для первого игрока: p1 = {p1:.4f}, p2 = {p2:.4f}")
    print(f"Значение игры: V = {V:.8f}")
    q1 = (c2[k] - c1[k]) / (c1[j] + c2[k] - (c1[k] + c2[j]))
```

```
    print(f"Вероятность для второго игрока: q1 = {q1:.4f}\n")

def solve_task_2():
    print("=== Решение задачи 2: Матричная игра M × 2 ===")
    C3 = np.array([[3.4, 7.9], [9.3, 4.5]])
    print("Матрица C3:")
    print(C3)
    plot_solution(C3, "Графическое решение для матрицы C3")

    j, k = 0, 1
    q1 = (C3[j][1] - C3[k][1]) / (C3[k][0] + C3[j][1] - (C3[k][1] +
C3[j][0]))
    print(f"Вероятность для второго игрока: q1 = {q1:.4f}\n")

def solve_task_3():
    print("=== Решение задачи 3: Матричная игра M × N ===")
    C4 = np.array([[2, 2, -5], [1, -6, 9], [-4, 4, -11]]) + 11
    print("Матрица C4:")
    print(C4)

    def solve_lp(c, A_ub, b_ub):
        res = linprog(c, A_ub=A_ub, b_ub=b_ub, method='highs')
        if res.success:
            Z = res.fun
            V = 1 / Z
            p = res.x * V
            return V, p
        return None, None

    V1, p = solve_lp([1, 1, 1], -C4.T, [-1, -1, -1])
    if V1 is not None:
        print(f"Решение для первого игрока: V = {V1-11:.4f}, p = {p}")

    V2, q = solve_lp([-1, -1, -1], C4, [1, 1, 1])
    if V2 is not None:
        print(f"Решение для второго игрока: V = {V2-11:.4f}, q = {q}")

def solve_task_4():
    print("=== Решение задачи 4: Биматричная игра ===")
    A = np.array([[3, 2], [1, 4]])
    B = np.array([[2, 0], [2, 6]])
    print("Матрица A:")
    print(A)
    print("Матрица B:")
    print(B)

    _x, _y = 2/3, 1/2
    plt.grid()
    plt.plot([0, 0, 1, 1], [0, _y, _y, 1], c='red', alpha=0.9, label='y',
linewidth=2)
    plt.plot([0, _x, _x, 1], [0, 0, 1, 1], c='blue', alpha=0.5, label='x',
linewidth=2)
    plt.legend()
    plt.show()

def solve_task_5(matrix):
    print("=== Решение задачи 5: Матричная игра 2 × 2 ===")
    plot_solution(matrix, "Графическое решение для матрицы")
```

```
    j, k = 0, 1
    c1, c2 = matrix[0], matrix[1]
    p1 = (c2[j] - c2[k]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    p2 = (c1[k] - c1[j]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    V = (c1[k] * c2[j] - c1[j] * c2[k]) / (c1[k] + c2[j] - (c1[j] + c2[k]))
    q1 = (matrix[j][1] - matrix[k][1]) / (matrix[k][0] + matrix[j][1] -
(matrix[k][1] + matrix[j][0]))
    print(f"Вероятности для первого игрока: p1 = {p1:.4f}, p2 = {p2:.4f}")
    print(f"Вероятность для второго игрока: q1 = {q1:.4f}")
    print(f"Значение игры: V = {V:.8f}\n")

A = np.array([[3, 2], [1, 4]])
B = np.array([[2, 0], [2, 6]])

solve_task_1()
solve_task_2()
solve_task_3()
solve_task_4()
solve_task_5(A)
solve_task_5(B)
```