**Министерство образования Российской Федерации**

# МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет: Информатика и системы управления Кафедра: Информационная безопасность (ИУ8)

# ТЕОРИЯ ИГР И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

**Лабораторная работа №3 на тему:**

«Неантагонистические игры. Критерий выбора оптимальных стратегий в бескоалиционных играх нескольких игроков»

Вариант 3

|  |  |
| --- | --- |
| Преподаватель: | Коннова Н.С. |
| Студент: | Валиев В.С. |
| Группа: | ИУ8-104 |

Москва, 2025

# Цель работы

Изучить критерий выбора стратегий в неантагонистической бескоалиционной игре двух игроков на основе равновесия Нэша и оптимальности по Парето. Проверить данные критерии на примере игр

«Семейный спор», «Перекресток», «Дилемма заключенного». Исследовать свойства оптимальных решений неантагонистических бескоалиционных игр на примере биматричных (2 × 2) игр.

# Задание

1. Сгенерировать случайную биматричную игру (10 × 10). Найти ситуации, равновесные по Нэшу и оптимальные по Парето, а также пересечение множеств этих ситуаций. Выполнить проверку реализованных алгоритмов на примере игр «Семейный спор», «Перекресток», «Дилемма заключенного».
2. Для заданной биматричной игры

Γ(𝐴, 𝐵) = ((3, 1) (5, 1)),

(9, 6) (2, 3)

пользуясь теоремой о свойствах оптимальных решений, найти ситуации, равновесные по Нэшу, для искомой игры и для ее смешанного расширения.

# Ход работы

## Задание 1

Биматрица случайной игры (10 × 10) приведена далее на рисунке 1: первое значение в ячейке соответствуют стратегиям первого игрока, второе значение – второго игрока.



Рисунок 1 – Биматрица случайной игры с выделенными оптимальными и устойчивыми стратегиями.

Так, устойчива по Нэшу ситуация: (31, 72). Оптимальны по Парето: (99, -70), (91, 24), (41, 98), (98, -50), (89, 93). Одновременно устойчивая по Нэшу и оптимальная по Парето: (84, 94).

Рассмотрим некоторые частные случаи игр и проведем проверку реализованных алгоритмов на них.

## Игра «Семейный спор»

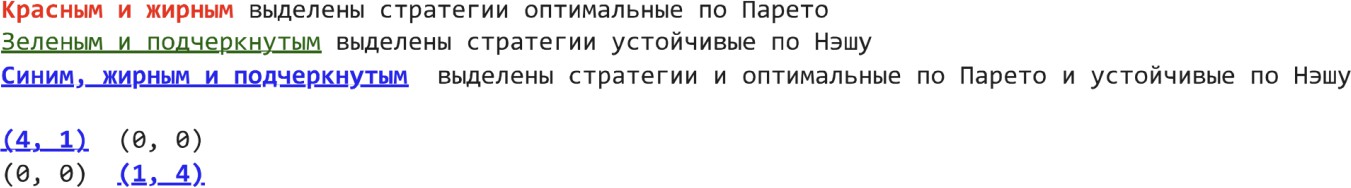


Рисунок 2 – Биматрица игры «Семейный спор» с выделенными оптимальными и устойчивыми стратегиями.

## Игра «Перекресток»

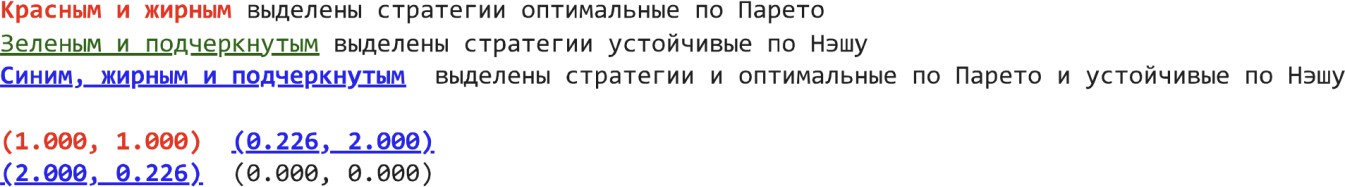


Рисунок 3 – Биматрица игры «Перекресток» с выделенными оптимальными и устойчивыми стратегиями.

## Игра «Дилемма заключенного»

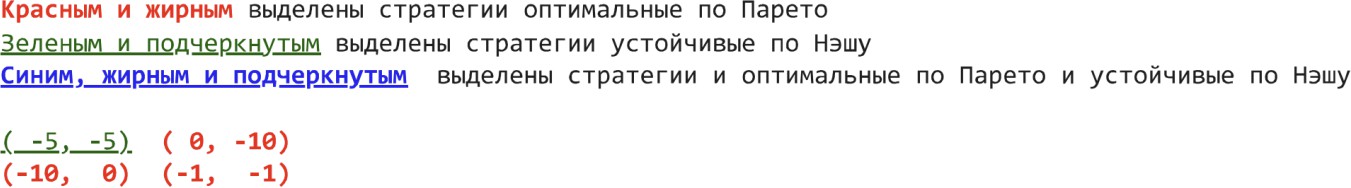


Рисунок 4 – Биматрица игры «Дилемма заключенного» с выделенными оптимальными и устойчивыми стратегиями.

## Задание 2

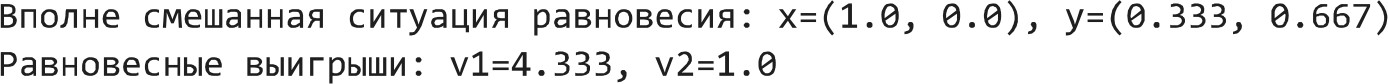
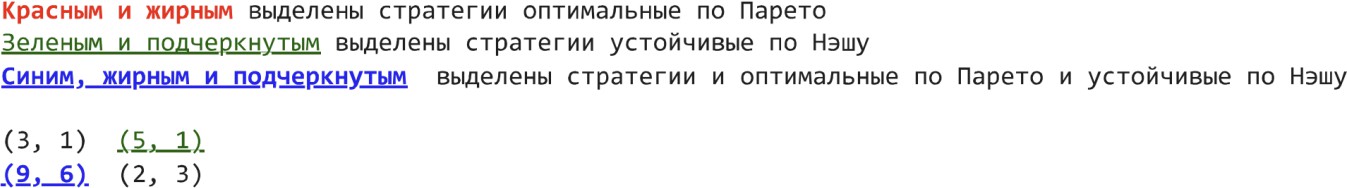


Рисунок 2 – Матрица игры для задания 2.

# Выводы

В ходе лабораторной работы были изучены критерии выбора стратегий в неантагонистической некооперативной игре двух участников: равновесие Нэша и оптимальность по Парето. Проведено исследование свойств оптимальных решений и проверена корректность работы созданной программы на примере биматричных игр размерности (2×2) и (10×10).

Равновесие Нэша демонстрирует устойчивость стратегий, но может не обеспечивать коллективно выгодного результата.

Парето-оптимальные решения могут увеличить выигрыши обоих игроков, но могут быть неустойчивыми с точки зрения индивидуальной рациональности.

В случае отсутствия равновесия в чистых стратегиях смешанное расширение гарантирует существование решения

import random import numpy as np

random.seed(14)



def generate\_bimatrix(GAME\_SIZE, MAX\_VAL, MIN\_VAL): rows, cols= GAME\_SIZE

matrix=[]

for\_ in range(rows): row= []

for\_ in range(cols):

playerl = random.randint(MIN\_VAL, MAX\_VAL) player2 = random.randint(MIN\_VAL, MAX\_VAL) row.append((playerl, player2))

matrix.append(row) return matrix

def print\_matrix(matrix, pareto\_indices=None, nash\_indices=None):

# ANSI escape-KOAW RESET= "\033[0m" BOLD= "\033[1m" UNDERLINE= "\033[4m" RED= "\033[91m" GREEN= "\033[32m" BLUE= "\033[34m"

if pareto\_indices or nash\_indices:

print(f"{RED}{BOLD}KpaCHblM " lK"PHblM{RESET} BblAe/leHbl CTpaTern" onT"Ma/lbHble no napeTo") print( f"{GREEN}{UNDERLINE}3e11eHblM " nOAYepKHyTblM{RESET} BblAe/leHbl CTpaTern" YCTOi<Y-Bble no H3wy")

print(f"{BLUE}{BOLD}{UNDERLINE}C"H"M, lK"PHblM " nOAYepKHYTblM{RESET} BblAe/leHbl CTpaTern" " onT"Ma/lbHble no napeTO " ycrni<Y-Bble no H3WY print()

def get\_style(i, j):

is\_pareto = (i, j) in (pareto\_indices or[]) is\_nash = (i, j) in (nash\_indices or[])

if is\_pareto and is nash:

return f"{BLUE}{BOLD}{UNDERLINE}", RESET

if is\_pareto:

return f"{RED}{BOLD}", RESET

if is\_nash:

return f"{GREEN}{UNDERLINE}", RESET

return "", RESET

if not matrix: return

transposed= list(zip(\*matrix))

# <l>yHKLj"R A/lR pac«eTa MaKrnMa/lbHOi< A/l"Hbl def get\_max\_lengths(pair\_index):

return [

max(

len(f"{pair[pair\_index]:.3f}"

if isinstance(pair[pair\_index], float) else str(pair[pair\_index]))

for pair in col for col in transposed

max\_pl get\_max\_lengths(0) max\_p2 get\_max\_lengths(l)

for i, row in enumerate(matrix): line = []

for j, (a, b) in enumerate(row): style\_start, style\_end = get\_style(i, j)

a\_str b\_str

f"{a:.3f}" if isinstance(a, float) else str(a) f"{b:.3f}" if isinstance(b, float) else str(b)

cell = f"{style\_start}({a\_str: >{max\_pl[j]}}, {b\_str: >{max\_p2[j]}}){style\_end}" line.append(cell)

print(" ".join(line))

def find\_pareto\_optimal(matrix): pareto\_optimal = []

rows= len(matrix) if rows== 0:

return pareto\_optimal

cols= len(matrix[0]) for i in range(rows):

for j in range(cols):

current\_a, current\_b matrix[i][j] dominated= False

# nposepAeM see BO3MOlKHble KOM6-Hal1"" for x in range(rows):

for yin range(cols):

other\_a, other\_b = matrix[x][y]

# Ecn" Ha•AeHa CTpaTer"A, KOTopaA AOM"H"pyeT TeKYWYffi

if (other\_a >= current\_a and other\_b >= current\_b) and\ (other\_a > current\_a or other\_b > current\_b):

dominated= True break

if dominated: break

if not dominated: pareto\_optimal.append((i, j))

return pareto\_optimal

def find\_nash\_equilibria(matrix): nash\_equilibria = []

if not matrix or not matrix[0]: return nash\_equilibria

rows len(matrix) cols len(matrix[0])

for i in range(rows): for j in range(cols):

# nony4aeM Bbl"rpblW" "rpOKOB pl\_payoff, p2\_payoff = matrix[i][j]

# nposepAeM ycnos"e AnA nepsoro "rpoKa (MaKc"MYM s cTon6t1e) is\_pl\_max = all(matrix[x][j][0] <= pl\_payoff for x in range(rows))

# nposepAeM ycnos"e AnA BTOporo "rpoKa (MaKC"MYM B CTPOKe) is\_p2\_max = all(matrix[i][y][l] <= p2\_payoff for yin range(cols))

# Ecn" o6a ycnos"A s nonHeH if is\_pl\_max and is\_p2\_max:

nash\_equilibria.append((i, j)) return nash\_equilibria

v **l.-1cxoAHas:t 61-1Marp1-14a**

game\_matrix = generate\_bimatrix(GAME\_SIZE=(10, 10), MAX\_VAL=100, MIN\_VAL=-100) print\_matrix(game\_matrix)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -73, 57) | ( 79, 93) ( 66, 34) | (-37, -31) | ( 88, -35) | (-26, 87) | (-82, 68) | 15, -23) | ( 19, | 75) | ( 1, 0) |
| 99, -70) | (-33, -43) (-20, -9) | (-34, -8) | ( 61, 61) | ( 32, -62) | (-59, 41) | 70, 69) | (-30, | -58) | (-98, 66) |
| -83, -69) | ( 52, -14) ( -93, -79) | (-30, -48) | ( -3, 3) | ( 49, 12) | ( 55, -76) | 64, 75) | (-72, | 48) | ( 50, 63) |
| 62, -7) | (-54, -76) ( 91, 24) | ( 31, 72) | ( -51, -31) | ( 15, 59) | (-45, 24) | (-28, 28) | (-35, | -74) | (-69, -78) |
| -29, -29) | (-70, -93) (-59, 89) | ( 5, -71) | ( 75, 34) | ( 50, -77) | ( 8, 22) | ( 75, -57) | ( 36, | -2) | ( 16, -21) |
| 24, 17) | ( 11, 6) ( 52, -76) | (-35, 21) | ( 0, -40) | ( 15, 57) | ( 25, -83) | ( 45, -64) | ( 25, | 80) | (-32, 46) |
| 12, -96) | ( 80, 3) ( 8, -95) | (-12, 81) | ( 45, -6) | ( 20, -33) | ( -9, -23) | ( 35, -52) | (-90, | 60) | ( 43, 20) |
| -35, -76) | ( -1, -75) ( -92, -51) | (-21, -94) | (-100, 50) | ( 29, -28) | ( 33, -9) | (-36, -71) | (-92, | -63) | ( 98, -50) |
| (-100, -42) | (-36, 28) (-33, -54) | (-37, -47) | ( 54, 74) | (-16, -26) | ( 84, 94) | ( -6, -92) | ( 43, | 52) | ( 89, 93) |
| ( -60, 30) | ( 65, -50) ( -7, 25) | ( -2, -38) | ( -1, 39) | ( 41, 98) | (-20, 24) | ( 56, -92) | (-37, | 64) | ( 9, 20) |

pareto = find\_pareto\_optimal(game\_matrix) nash = find\_nash\_equilibria(game\_matrix)

print\_matrix(game\_matrix, pareto, nash)

**KpaCHblM 11 lKl1pHblM** BblAe/leHbl CTpaTern11 onT11MaJ1bHble no napeTO

**3e/leHblM *vi* nOAYepKHYTblM BblAeJleHbl CTpaTeno1 ycTO Y"1Bble no H3WY**

**(11Hl1M, lKl1pHblM 11 nOA4epKHYTblM** BblAe/leHbl CTpaTern11 **11 onTli1MaJlbHble** no napeTo 11 YCTOa411Bble no H3wy

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ( -73, 57) | ( 79, 93) | ( 66, 34) | (-37, -31) | 88, -35) | (-26, 87) | (-82, 68) | ( 15, -23) | ( 19, 75) | ( 1, 0) |
| ( **99, -70)** | (-33, -43) | ( -20, -9) | (-34, -8) | 61, 61) | ( 32, -62) | (-59, 41) | ( 70, 69) | (-30, -58) | (-98, 66) |
| ( -83, -69) | ( 52, -14) | ( -93, -79) | (-30, -48) | -3, 3) | ( 49, 12) | ( 55, -76) | ( 64, 75) | (-72, 48) | ( 50, 63) |
| ( 62, -7) | (-54, -76) | ( **91, 24)** | ( 31, 72) | -51, -31) | ( 15, 59) | (-45, 24) | (-28, 28) | (-35, -74) | (-69, -78) |
| ( -29, -29) | (-70, -93) | (-59, 89) | ( 5, -71) | 75, 34) | ( 50, -77) | ( **8,** 22) | ( 75, -57) | ( 36, -2) | ( 16, -21) |
| ( 24, 17) | ( 11, 6) | ( 52, -76) | (-35, 21) | 0, -40) | ( 15, 57) | ( 25, -83) | ( 45, -64) | ( 25, 80) | (-32, 46) |
| ( 12, -96) | ( 80, 3) | ( 8, -95) | (-12, 81) | 45, -6) | ( 20, -33) | ( -9, -23) | ( 35, -52) | (-90, 60) | ( 43, 20) |
| ( -35, -76) | ( -1, -75) | ( -92, -51) | (-21, -94) | (-100, 50) | ( 29, -28) | ( 33, -9) | (-36, -71) | (-92, -63) | ( **98, -50)** |
| (-100, -42) | (-36, 28) | (-33, -54) | (-37, -47) | ( 54, 74) | (-16, -26) | ( **84, 94)** | ( -6, -92) | ( 43, 52) | ( **89, 93)** |
| ( -60, 30) | ( 65, -50) | ( -7, 25) | ( -2, -38) | ( -1, 39) | ( **41, 98)** | (-20, 24) | ( 56, -92) | (-37, 64) | ( 9, 20) |

### v CeMeV!HblVI cnop

battle\_of\_sexes\_matrix [(4, 1), (0, 0)],

[(0, 0), (1, 4)],

pareto = find\_pareto\_optimal(battle\_of\_sexes\_matrix) nash = find\_nash\_equilibria(battle\_of\_sexes\_matrix)

print\_matrix(battle\_of\_sexes\_matrix, pareto, nash)

**:B, KpaCHblM 11 lK11pHblM** BblAe/leHbl CTpaTern11 onT11MaJ1bHble no napeTO 3eneHblM **11** nOA4epKHYTblM BblAe/leHbl CTpaTern11 ycT0 411Bble no H3WY

**CMHMM, lKMpHblM M nOA4epKHYTblM** BblAe/leHbl CTpaTern11 **11** onnMa/lbHble no napeTO **11** ycTOa411Bble no H3wy

**a.....ll** (0, 0)

(0, 0) **.(1, g\_)\_**

### v nepeKpecrnK

EPSILON= random.random() crossroad\_matrix = [

[(1.000, 1.000), (1 - EPSILON, 2.000)],

[(2.000, 1 - EPSILON), (0.000, 0.000)],

pareto = find\_pareto\_optimal(crossroad\_matrix) nash = find\_nash\_equilibria(crossroad\_matrix)

print\_matrix(crossroad\_matrix, pareto, nash)

**:B, KpaCHblM 11 lKMpHblM** BblAe/leHbl CTpaTern11 onT11MaJ1bHble no napeTO 3eneHblM **11** nOA4epKHYTblM BblAe/leHbl CTpaTer1111 ycT0 411Bble no H3WY

CMHMM, lK11pHb1M M nOA4epKHYTblM BblAe/leHbl CTpaTern11 11 onT11MaJ1bHble no napeTO 11 ycTOa411Bble no H3WY

(1.000, 1.000)

(2.000, 0.226)

(0.226, 2.000)

(0.000, 0.000)

### v ,l],i-ineMMa 3aKnl04eHHoro

prisoners\_dilemma\_matrix =

[(-5, -5), (0, -10)],

[(-10, 0), (-1, -1)],

pareto = find\_pareto\_optimal(prisoners\_dilemma\_matrix) nash = find\_nash\_equilibria(prisoners\_dilemma\_matrix)

print\_matrix(prisoners\_dilemma\_matrix, pareto, nash)

**KpaCHblM 11 lKMpHblM** BblAe/leHbl CTpaTern11 onT11MaJ1bHble no napeTO 3eneHblM **11** nOA4epKHYTblM BblAeJ1eHbl CTpaTerMH ycTOa411Bble no H3wy

**(MHMM, lKMpHblM M nOA4epKHYTblM** BblAe/leHbl CTpaTern11 11 onT11MaJ1bHble no napeTO 11 ycTOa411Bble no H3WY

( -5, -5)

**(-10, 0)**

( **0, -10)**

**(-1, -1)**



ROUND= 3

def get\_mbs(bimatrix):

player\_A, player\_B = bimatrix[:, :, 0], bimatrix[:, 1]

if not np.linalg.det(player\_A):

raise Exception('Matrix A is singular') if not np.linalg.det(player\_B):

raise Exception('Matrix Bis singular')

u = np.array([l] \* bimatrix.shape[0])

vl = round(float(l / (u.dot(np.linalg.inv(player\_A)).dot(u))), ROUND) v2 = round(float(l / (u.dot(np.linalg.inv(player\_B)).dot(u))), ROUND) x = v2 \* u.dot(np.linalg.inv(player\_B))

y = vl \* np.linalg.inv(player\_A).dot(u)

return vl, v2, tuple(round(float(v), ROUND) for v in x), tuple(round(float(v), ROUND) for v in y)

matrix=

[(3, 1), (5, l)l,

[(9, 6), (2, 3)],

pareto = find\_pareto\_optimal(matrix) nash = find\_nash\_equilibria(matrix)

print\_matrix(matrix, pareto, nash)

**KpaCHblM 11 lKl1pHblM** BblAe/leHbl CTpaTern11 onT11MaJ16Hble no napeTO 3eneHblM **11** nOAYepKHYTblM BblAeJ1eHbl CTpaTer1111 ycToi,y11Bble no H3wy

**(11HMM, lKMpHblM M nOAYepKHYTblM** BblAe/leHbl CTpaTer1111 11 onT11MaJ1bHble no napeTO 11 ycToi,Yl1Bble no H3WY

(3, 1)

*!2.,\_§.)\_*

li.,\_ll

(2, 3)

matrix= np.array(matrix)

vl, v2, x, y = get\_mbs(matrix)

print(f'Bno11He CMewaHHaR c11Tyau11R paBHOBeC11R: X={x}, y={y}') print(f'PaBHOBeCHble Bbl11Cpb1Wl1: vl={vl}, v2={v2}')

**T** Bno11He CMewaHHaR c11Tyau11R paBHOBeC11R: x=(l.0, 0.0), y=(0.333, 0.667) PaBHOBeCHble Bbl11Cpb1Wl1: v1=4.333, v2=1.0