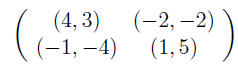
1. Теоретическая часть

Неантагонистические игры — это игры, в которых интересы сторон не обязательно противоположны. Такие игры описываются матрицами, например, такого вида:

В каждой клетке матрицы стоит пара чисел: выигрыши первого и второго игроков соответственно. Например, если первый игрок применит вторую стратегию, а второй применит первую стратегию, то первый игрок проиграет 1 единицу, а второй проиграет 4 единицы. Игры такого типа называют биматричными. Как видно, интересы игроков могут частично совпадать. В данной лабораторной работе рассматривается случай некооперативного поведения игроков: игроки полностью лишены возможности как-либо договариваться и согласовывать свои ходы. В таких случаях, как и в антагонистических играх, бывает уместно применять смешанные стратегии. Пусть дана матрица биматричной игры 2×2. Пусть каждый из игроков применяет смешанные стратегии: первый выбирает свою первую стратегию с вероятностью x, а вторую \_ с вероятностью 1 − x. Аналогично, второй игрок выбирает свои стратегии с вероятностями y и 1 − y. Тогда могут быть подсчитаны средние выигрыши обоих игроков: S1(x, y) для первого игрока и S2(x, y) для второго. При заданных вероятностях x и y выигрыш может быть изображен точкой (S1(x, y), S2(x, y)) в плоскости с координатами S1, S2. При изменении вероятностей выбора стратегий x и y эта точка также будет меняться. Для всех возможных наборов (x, y) точки (S1(x, y), S2(x, y))будут заполнять некоторую область в плоскости с координатами S1, S2.Таким образом, пара функций (S1(x, y) и S2(x, y)) задает отображение единичного квадрата [0, 1] × [0, 1] в плоскости x, y на некоторую область в плоскости S1, S2. Эта область и есть множество достижимых итогов.

1. Программный код

"""

Программа, иллюстрирующая множество всех возможных исходов в неантагонистической

биматричной игре для случая некооперативного поведения игроков

Автор: Афанасьев И.Е.

Дата написания: 20.09.2020

"""

# Матрица

#m = [[2, 1], [-1, -1], [-1, -1], [1, 2]]

m = [[4, 3], [-2, -2], [-1, -4], [1, 5]]

# Импортируем библиотеку для поиска экстремумов

from scipy.optimize import minimize

# формула выигрыша первого игрока

def S1(x):

return m[0][0]\*x[0] \* x[1] \

+ m[1][0]\*(1 - x[0]) \* x[1] \

+ m[2][0]\*x[0] \* (1 - x[1]) \

+ m[3][0]\*(1 - x[0]) \* (1 - x[1])

# формула выигрыша второго игрока

def S2(x):

return x[0] \* m[0][1]\*x[1] \

+ (1 - x[0]) \* m[1][1]\*x[1] \

+ x[0] \* m[2][1]\*(1 - x[1]) \

+ (1 - x[0]) \* m[3][1]\*(1 - x[1])

# Ищем экстремальную точку для первого игрока

res = minimize(S1, [1, 1])

print("Экстремум для первого игрока: ", res.x)

print("Значение выигрыша: ", S1(res.x))

# Ищем экстремальную точку для второго игрока

res = minimize(S2, [1, 1])

print("Экстремум для второго игрока: ", res.x)

print("Значение выигрыша: ", S2(res.x))

# Библиотека для матричных вычислений, нам она нужна для создания

# массива от 0 до 1

import numpy as np

xval = np.linspace(0, 1, 51)

yval = np.linspace(0, 1, 51)

# Библиотека для графиков

import matplotlib.pyplot as plt

x, y = np.meshgrid(xval, yval)

z1 = S1([x, y])

z2 = S2([x, y])

plt.scatter(z1, z2)

plt.show()

1. Результаты работы программы

Пусть:

m = [[1, 2], [-1, -1], [-1, -1], [2, 1]]

Результат:

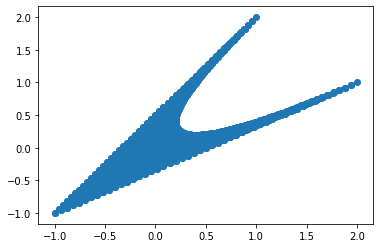
Экстремум для первого игрока: [0.6 0.6]

Значение выигрыша: 0.20000000000000007

Экстремум для второго игрока: [0.4 0.4]

Значение выигрыша: 0.2

Область достижимых исходов:



Пусть теперь:

m = [[4, 3], [-2, -2], [-1, -4], [1, 5]]

