Міністерство освіти і науки України
Національний авіаційний університет
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютеризованих систем управління

Лабораторна робота № 7
з дисципліни «Системи підтримки прийняття рішень»
на тему «Теорія ігор. Рішення матричних ігор у змішаних стратегіях»
Варіант № 3

Виконав:
студент ФККПІ
групи СП-425
Клокун В. Д.
Перевірила:
Яковенко Л. В.

Київ 2020

1. Мета роботи

Ознайомитись з теорією ігор і змішаними стратегіями.

2. Хід роботи

ЗАДАЧА Вирішіть графічно наступні ігри, в яких платежі виплачують гравцеві A.

Табл. 1: Задані ігри двох осіб з нульовою сумою

(а)

	B_1	B_2	B_3
A_1	1	-3	7
A_2	2	4	6

(б)

	B_1	B_2
A_1	5	8
A_2	6	5
A_3	5	7

РОЗВ'ЯЗАННЯ Графічний метод полягає у побудові графіків виплат для гравця, який нас цікавить, і пошуку точок перетину. На основі точок перетину робиться висновок про розв'язок гри.

Розроблюємо програму, яка будуватиме графіки виплат для гравця A для представлених ігор. Вона складатиметься з модуля, який розв'язуватиме задачу (лістинг А.1). Запускаємо програму і спостерігаємо результат (рис. 1).

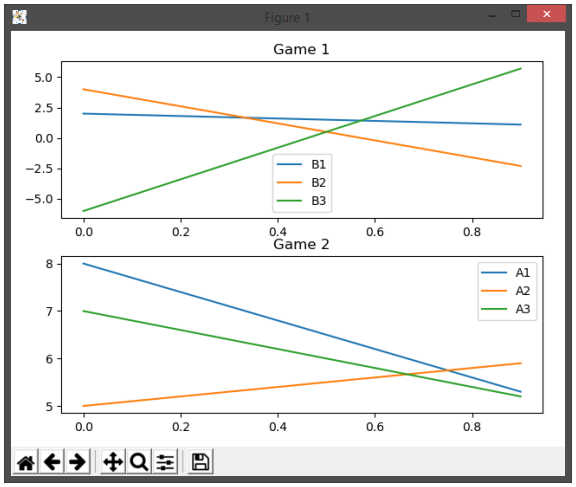

Рис. 1: Результат роботи розробленої програми

Видно, що розроблена реалізація розв'язала поставлену задачу і визначила точки перетину виплат при різних стратегіях.

3. Висновок

Виконуючи дану лабораторну роботу, ми ознайомились з теорією ігор і змішаними стратегіями.

А. Лістинг коду програмної реалізації

Лістинг А.1: Файл `main.py`

```
import matplotlib.pyplot as plt
import numpy as np


GAME_1 = [
    [  1, -3,  7],
    [  2,  4, -6],
]


GAME_2 = [
    [  5,  8],
    [  6,  5],
    [  5,  7],
]

GAMES = [GAME_1, GAME_2]


def transpose(matrix):
    return list(map(list, zip(*matrix)))


def build_coefficients_from_game(game):
    coeffs = []

    game_transposed = transpose(game)

    for row in game_transposed:
        new_row = [row[0] - row[1], row[1]]
        coeffs.append(list(reversed(new_row)))

    return coeffs


def calc_fn_by_coefficients(coeffictients, x):
```

```
    """Calculates the value of a polynomial function f(x) by its
     ↪ coefficients.
    """
    res = 0
    for idx, c in enumerate(coeffictients):
        res += (c * x**idx)

    return res


def main():
    x = np.arange(0, 1, 0.1)
    fig, axs = plt.subplots(2, 1)
    ax1, ax2 = axs

    coeffs = build_coefficients_from_game(GAME_1)
    print(coeffs)
    ax1.set_title("Game 1")
    for idx, c in enumerate(coeffs, start=1):
        ax1.plot(
            x,
            [calc_fn_by_coefficients(c, i) for i in x],
            label="B{}".format(idx)
        )

    plt.tight_layout()

    coeffs = build_coefficients_from_game(transpose(GAME_2))
    print(coeffs)
    ax2.set_title("Game 2")
    for idx, c in enumerate(coeffs, start=1):
        ax2.plot(
            x,
            [calc_fn_by_coefficients(c, i) for i in x],
            label="A{}".format(idx)
        )

    ax1.legend()
    ax2.legend()
    plt.show()


if __name__ == '__main__':
    main()
```