**Прізвище:** Литвин

**Ім’я:** Андрій

**По-батькові:** Андрійович

**Група:** КН-406

**Варіант:** 9

**GitHub:**

**Кафедра:** САПР

**Дисципліна:** Теорія прийняття рішень

**Перевірив:** Кривий Р.З.

**ЗВІТ**

до лабораторної роботи №5

на тему: “Теорія ігор. Матричні ігри”

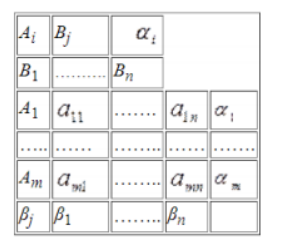
Мета:

Визначити основні поняття теорії ігор, властивості змішаних стратегій. Вивчити метод вирішення матричних ігор у змішаних стратегіях за допомогою введення до подвійних завдань лінійного програмування.

Теоретичні відомості:

Нехай у кожного з двох гравців *A* і *B* скінченне число можливих дій – чистих стратегій: гравець *A* володіє *m* чистими стратегіями *A1*, *A2*, …, *Am*, а гравець *B* – *n* чистими стратегіями *B1*, *B2*, …., *Bn*. Щоб гра була повністю визначена, необхідно вказати правило, яке кожній парі чистих стратегій (*Aі*; *Bj*) ставить у відповідність число *aij* – виграш гравця *A* за рахунок гравця *B* або програш гравця *B*. При *aij < 0* гравець *A* платить гравцю *B* суму *|аij|*. В грі, яка складається тільки з особистих ходів, вибір пари чистих стратегій (*Aі*; *Bj*) єдиним чином визначає її результат. Якщо ж в грі використовуються і випадкові ходи, то її результат обумовлюється середнім значенням виграшу (математичним сподіванням).

Якщо відомі значення *aij* виграшу для кожної пари (*Aі*; *Bj*) стратегій, то можна записати матрицю гри (платіжну матрицю):



Платіжна матриця – це табличний запис функції виграшу. Описані ігри називають матричними. Окрема партія в такій грі реалізується наступним чином. Гравець *A* вибирає один із рядків платіжної матриці (одну з своїх чистих стратегій). Елемент матриці, який стоїть на перетині вибраного рядка і стовпця, визначає виграш гравця *A* (програш гравця *B*).

Метою гравців є вибір найбільш вигідних стратегій, при яких гравець *A* вибирає максимальний виграш, а *B* – мінімальний програш. В теорії ігор виходять з припущення, що кожен гравець вважає свого супротивника розумним і намагається не дати йому досягти найкращого результату.

Індивідуальне завдання

У грі беруть участь два гравці: A і B. У розпорядженні кожного гравця є кінцеве безліч варіантів вибору - стратегій. Нехай - безліч стратегій гравця A, - безліч стратегій гравця B. З кожною парою стратегій пов'язаний платіж, який один з гравців виплачує іншому. Тобто, коли гравець А вибирає стратегію (свою i-ю стратегію), а гравець В - стратегію, то результатом такого вибору стає платіж. Оскільки стратегій кінцеве число, то платежі утворюють матрицю розмірності n x m, звану матрицею платежів (або матрицею гри). Рядки цієї матриці відповідають стратегіям гравця А, а стовпці - стратегіям гравця В.

**Порядок виконаних робіт:**

1. Вихідні дані беруть із варіантів індивідуальних завдань.
2. При вирішенні матричної гри потрібно вийти на наступні етапи:
   1. Знайти сідлову точку і перевірити, чи має гра вирішення в чистих стратегій.
   2. У випадку відсутності чистої стратегії, знайти рішення в оптимальних змішаних стратегіях
   3. Спростити платіжну матрицю (перевірити матрицю на домінуючі рядки і стовбці).
   4. Визначити оптимальні плани за допомогою одного з методів лінійного програмування.
   5. Знайдіть рішення гри.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **9.** | 3 | 6 | 11 | 7 | 4 |
| 8 | 10 | 4 | 9 | 11 |
| 7 | 4 | 8 | 6 | 10 |
| 5 | 12 | 10 | 7 | 11 |
| 8 | 6 | 11 | 5 | 3 |

Хід та результати розв’язку

1. Перевіряємо, чи платіжна матриця має сідлову точку. Якщо так, то виписуємо рішення гри у чистих стратегіях. Для зручності обрахунків напів ручний метод я проводив в Екселі, за допомогою засобів роботи з матрицями та функцією пошуку рішень – використовуючи симплекс метод.

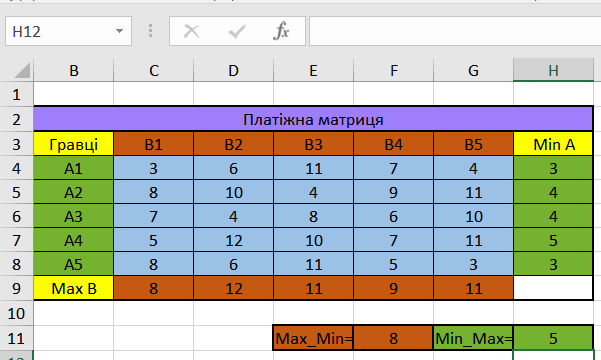


Рис.1

Знаходимо гарантований виграш, який визначається нижньою ціною гри *a = max(ai) = 5* та верхньою ціною гри *b = min(bj) = 8*. Що свідчить про відсутність сідлової точки, оскільки *a ≠ b*, тоді ціна гри знаходиться в межах *5 ≤ y ≤ 8*. Отже, знаходимо рішення гри у змішаних стратегіях.

1. Перевіряємо платіжну матрицю на домінуючі рядки та домінуючі стовпці. Іноді на підставі простого розгляду матриці гри можна сказати, що деякі чисті стратегії можуть увійти до оптимальної змішаної стратегії лише з нульовою ймовірністю.

Але в даному варіанті роботи не має домінуючих рядків чи стовпців. Тому гра залишається розміром 5 x 5.

1. Знаходимо рішення гри у змішаних стратегіях. Математичні моделі пари двоїстих завдань лінійного програмування можна записати так:

знайти мінімум функції *F(x)* при обмеженнях (для гравця №2):

знайти максимум функції *F(y)* при обмеженнях (для гравця №1):

За допомогою пошуку рішень середовища Exel розв’язуєм ці дві системи рівнянь (рис. 3-4):

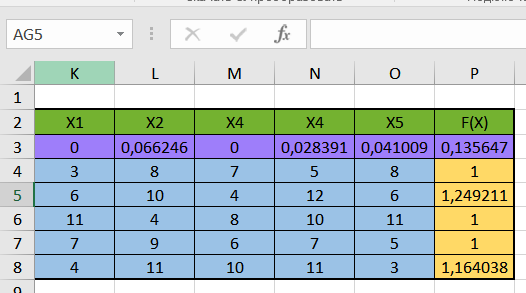


Рис.2

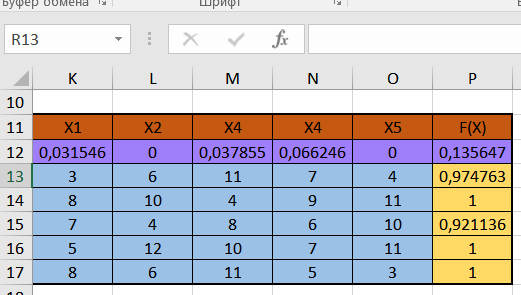


Рис.3

Знаходимо ціну гри та використовуємо формули змішаних стратегій, перевіряємо чи їх сума рівна 1. (рис. 8-9):

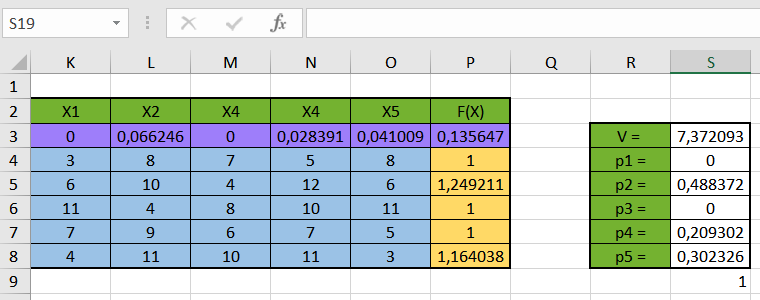


Рис.4

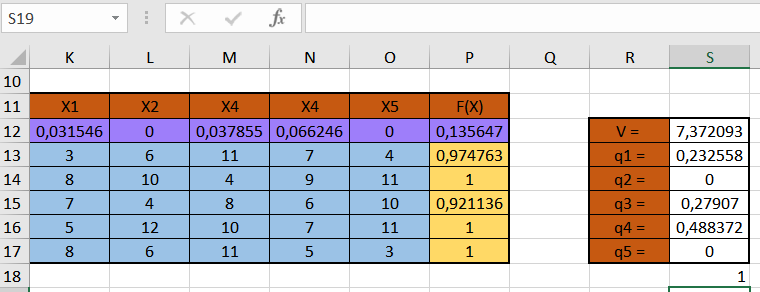


Рис.5

Отже, V = 7.37 ; P = (0; 0.49; 0; 0.21; 0.30); Q = (0.23; 0; 0.28; 0.49; 0)

Програма на python

*import numpy as np*

*from tabulate import tabulate*

*from pulp import LpVariable, LpProblem, LpMinimize, value, LpMaximize*

*def show\_table(data, head):*

*show = tabulate(data, headers=head, tablefmt='fancy\_grid', numalign='center')*

*print(show)*

*def simplex(matrix, x\_y):*

*variables = []*

*for i in range(0, len(matrix)):*

*variables.append(LpVariable(x\_y + str(i + 1), lowBound=0))*

*function = ''*

*if x\_y == "x":*

*print("Система рівнянь для стратегії А: ")*

*problem = LpProblem("Система рівнянь", LpMinimize)*

*for i in range(0, len(matrix)):*

*changed = sum(matrix[i] \* variables) >= 1*

*print(changed)*

*problem += changed*

*function = sum(variables)*

*print("Функція:")*

*print("F(%s) = %s --> MIN" %(x\_y, function))*

*else:*

*print("Система рівнянь для стратегії B: ")*

*problem = LpProblem("Система рівнянь", LpMaximize)*

*for i in range(0, len(matrix)):*

*changed = sum(matrix[i] \* variables) <= 1*

*print(changed)*

*problem += changed*

*function = sum(variables)*

*print("Функція:")*

*print("F(%s) = %s --> MAX" %(x\_y, function))*

*problem += function*

*problem.solve()*

*res = []*

*for variable in problem.variables():*

*res.append(variable.varValue)*

*F = value(problem.objective)*

*return res, F, variables*

*def Game\_price\_and\_probability(res\_x\_y):*

*V = 1 / sum(res\_x\_y)*

*probability = []*

*for element in res\_x\_y:*

*probability.append(element \* V)*

*return V, probability*

*matrix = np.loadtxt("lab 5.txt", dtype=int)*

*matrix\_list = matrix.tolist()*

*header = []*

*for i in range(0, len(matrix\_list)):*

*matrix\_list[i].insert(0, "A" + str(i+1))*

*header.append("B" + str(i+1))*

*header.insert(0, "Гравці")*

*print("Платіжна матриця:")*

*show\_table(matrix\_list, header)*

*transpose\_matrix = np.transpose(matrix)*

*res\_for\_x, Function\_x, head\_x = simplex(transpose\_matrix, "x")*

*head\_x.append("F(x)")*

*print("Розв'язки для гравця A:")*

*show\_table([res\_for\_x + [Function\_x]], head\_x)*

*res\_for\_y, Function\_y, head\_y = simplex(matrix, "y")*

*head\_y.append("F(y)")*

*print("Розв'язки для гравця B:")*

*show\_table([res\_for\_y + [Function\_y]], head\_y)*

*Price\_X, p\_x = Game\_price\_and\_probability(res\_for\_x)*

*head\_p = []*

*for i in range(0, len(p\_x)):*

*head\_p.append("p" + str(i+1))*

*head\_p.append("Ціна гри")*

*print("Ціна гри і ймовірності застосування стратегій для гравця А:")*

*show\_table([p\_x + [Price\_X]], head\_p)*

*Price\_Y, p\_y = Game\_price\_and\_probability(res\_for\_y)*

*head\_q = []*

*for i in range(0, len(p\_y)):*

*head\_q.append("q" + str(i+1))*

*head\_q.append("Ціна гри")*

*print("Ціна гри і ймовірності застосування стратегій для гравця B:")*

*show\_table([p\_y + [Price\_Y]], head\_q)*

Результати виконання програми

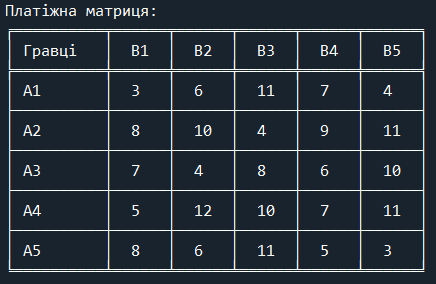


Рис.6 Матриця зчитується з файлу вже без домінуючих рядків чи стовпців, але в моєму варіанті таких не виявилося

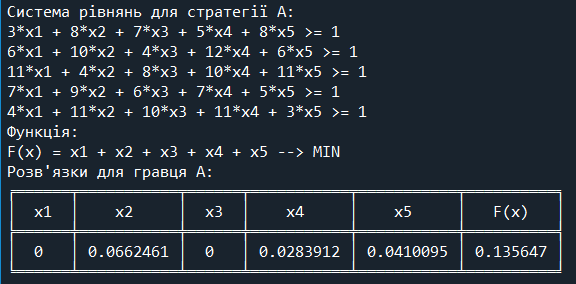


Рис.7 Розв’язки для гравця А

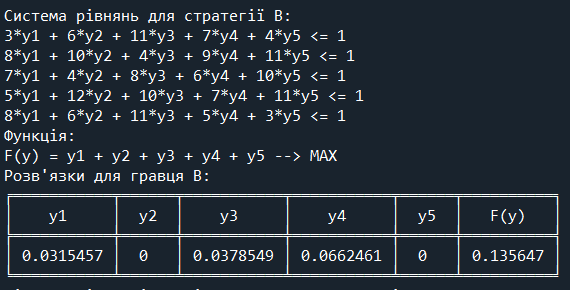


Рис.8 Розв’язки для гравця В

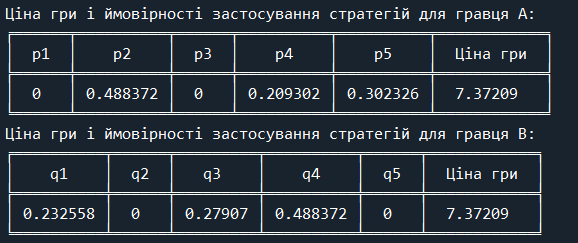


Рис.9 Ціни ігор і ймовірності застосування обох стратегій

Висновок

На даній лабораторній роботі я ознайомився з теорією матричних ігор, розв’язав одну із задану індивідуальним завданням в Excel – напів ручний метод, а також за допомогою мови програмування python – програмний метод. Був застосований симплекс-метод розв’язування для двох гравців і було знайдено змішані стратегії та ціну гри для кожного з учасників.