**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №5**

**з курсу**

**«**Системний аналіз та теорія прийняття рішень**»**

*студента 3 курсу*

*групи ПП-32*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Федосенка Станіслава Сергійовича

*Викладач:*

*Білий Р.О.*

**Київ – 2023**

**Тема:** Теорія ігор.

**Мета роботи:** Набути навички пошуку раціональних рішень в умовах конфліктів.

**Завдання**

**Варіант №4**

Завдання 1: розв’язання гри (в чистих стратегіях) з заданою матрицею платежів

1. Визначення за заданою матрицею платежів нижньої та верхньої ціни гри.

2. Встановити чи існує в грі рівновага в чистих стратегіях?

3. Розв’язання задачі програмно або з допомогою пакета MS Excel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | В1 | В2 | В3 | В4 |
| А1 | -1 | 9 | 6 | 8 |
| А2 | -2 | 10 | 4 | 6 |
| А3 | 5 | 3 | 0 | 7 |
| А4 | 7 | -2 | 8 | 4 |

*Завдання 2*: розв’язання гри

У нашому розпорядженні є три види озброєння: A1, A2, А3; у супротивника - три види літаків: B1, В2, В3. Наше завдання - вразити літак; задача противника- зберегти його неураженим. Літаки В1, В2 і В3 уражаються при використанні озброєння А1 відповідно з можливостями 0,9, 0,4 і 0,2; при використанні А2 - з вірогідністю 0,3, 0,6 і 0,8; при використанні А3 - з вірогідністю 0,5, 0,7 і 0,2.

**Хід роботи**

**Задача 1**

Розглянемо процес прийняття рішень обома сторонами, припускаючи, що обидва гравці діятимуть раціонально. Якщо гравець А не знає, як вчинить його противник, то, діючи найбільш доцільно і не бажаючи ризикувати, він вибере таку стратегію, яка гарантує йому найбільший з найменших виграшів за будь-якої стратегії противника. Тобто стратегія гравця A є максималізованою:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B1 | B2 | B3 | B4 | MIN(A) | MAXMIN(A) |
| A1 | -1 | 9 | 6 | 8 | -1 | 0 |
| A2 | -2 | 10 | 4 | 6 | -2 |
| A3 | 5 | 3 | 0 | 7 | 0 |
| A4 | 7 | -2 | 8 | 4 | -2 |

Зі всіх можливих стратегій доцільніше вибрати ту, що принесе максимальний можливий дохід (мінімальні можливі збитки, як у нашому випадку). У нашому випадку це стратегія 3. Отриману величину A3 – 0 називають **нижньою ціною гри**. Вона представляє собою максимальний гарантований виграш гравця A. Проведемо аналогічні розрахунки для гравця B.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | B1 | B2 | B3 | B4 |
|  | A1 | -1 | 9 | 6 | 8 |
|  | A2 | -2 | 10 | 4 | 6 |
|  | A3 | 5 | 3 | 0 | 7 |
|  | A4 | 7 | -2 | 8 | 4 |
|  | MAX(B) | 7 | 10 | 8 | 8 |
| MINMAX(B) | | 7 | | | |

Гравець В зацікавлений у тому, щоб виграш А був мінімальний. Він повинен переглянути кожну свою стратегію з точки зору максимального виграшу при цій стратегії та обрати ту, в якій цей виграш мінімальний. У нашому випадку це стратегія 1. Отриману величину B1 – 7 називають **верхньою ціною гри**. Вона представляє собою максимальний можливий програш гравця B (виграш гравця A).

В результаті ми дізналися, що нижня ціна гри – 0, верхня ціна гри – 7, а отже сідлової точки не існує, як і рівноваги в чистих стратегіях.

**Задача 2**

Спершу побудував матрицю ймовірностей

|  |  |  |  |
| --- | --- | --- | --- |
|  | B1 | B2 | B3 |
| A1 | 0,9 | 0,4 | 0,2 |
| A2 | 0,3 | 0,6 | 0,8 |
| A3 | 0,5 | 0,7 | 0,2 |

Далі для потрібно побудувати матрицю платежів. Розраховував значення за таким принципом: 1 – (шанс, що літак пролетів через засоби враження).

|  |  |  |  |
| --- | --- | --- | --- |
|  | B1 | B2 | B3 |
| A1 | 0,9 | 0,4 | 0,2 |
| A2 | 0,93 | 0,76 | 0,84 |
| A3 | 0,97 | 0,93 | 0,87 |

Можна зробити висновок, що найбільший шанс зберегти літак неураженим має третій літак, а найбільший шанс вразити літак має третій вид озброєння, адже в нього найменший шанс того, що літаки до нього долетять.

**Лістинг програми**

#%%  
import pandas as pd  
import numpy as np  
#%%  
matrix = pd.read\_excel('lab.xlsx', sheet\_name='задача 1',header=None, skiprows=2, usecols='D:G', nrows=4).to\_numpy()  
display(pd.DataFrame(matrix, columns=['B1','B2','B2','B2'], index=['A1','A2','A3','A4']))  
#%%  
minA = [min(row) for row in matrix]  
maxB = [max(row[i] for row in matrix) for i in range(len(matrix[0]))]  
display(pd.DataFrame(minA, columns=['minA'], index=['A1','A2','A3','A4']))  
display(pd.DataFrame(maxB, columns=['maxB'], index=['B1','B2','B3','B4']))  
#%%  
min\_max\_B = min(maxB)  
max\_min\_A = max(minA)  
print('Нижня ціна гри: A', np.argmax(minA), '-' , max\_min\_A)  
print('Верхня ціна гри: B', np.argmin(maxB), '-' , min\_max\_B)  
  
if(min\_max\_B == max\_min\_A):  
 print('Сідлова точка існує')  
else:  
 print('Сідлової точки не існує')

#%%  
import pandas as pd  
#%%  
matrix = pd.read\_excel('lab.xlsx', sheet\_name='задача 2',header=None, skiprows=4, usecols='E:G', nrows=3).to\_numpy()  
print('Матриця ймовірностей')  
display(pd.DataFrame(matrix, columns=['B1','B2','B2'], index=['A1','A2','A3']))  
#%%  
payments\_matrix = matrix  
# count first col  
var1 = matrix[0][0]  
var2 = matrix[1][0]  
var3 = matrix[2][0]  
payments\_matrix[1][0] = 1 - ((1 - var1)\*(1 - var2))  
payments\_matrix[2][0] = 1 - ((1 - var1)\*(1 - var2)\*(1 - var3))  
  
# count second col  
var1 = matrix[0][1]  
var2 = matrix[1][1]  
var3 = matrix[2][1]  
payments\_matrix[1][1] = 1 - ((1 - var1)\*(1 - var2))  
payments\_matrix[2][1] = 1 - ((1 - var1)\*(1 - var2)\*(1 - var3))  
  
# count third col  
var1 = matrix[0][2]  
var2 = matrix[1][2]  
var3 = matrix[2][2]  
payments\_matrix[1][2] = 1 - ((1 - var1)\*(1 - var2))  
payments\_matrix[2][2] = 1 - ((1 - var1)\*(1 - var2)\*(1 - var3))  
#%%  
print('Матриця платежів')  
display(pd.DataFrame(payments\_matrix, columns=['B1','B2','B2'], index=['A1','A2','A3']))

**Висновок**

В ході виконання даної лабораторної роботи, я набув навички пошуку раціональних рішень в умовах конфліктів, навчився знаходити нижню та верхню ціну гри.