Michał Wilkosz

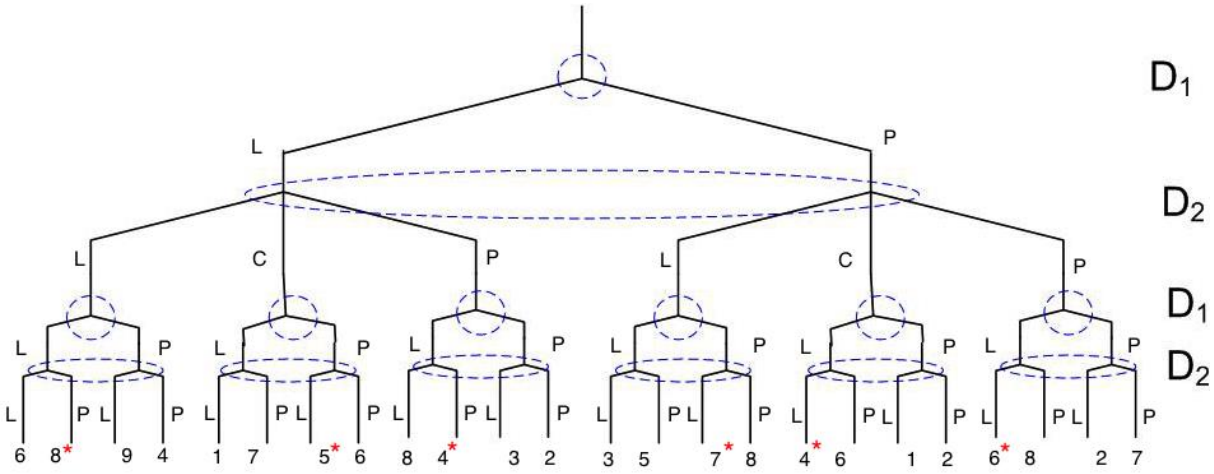
Decision Trees – Laboratory 07/04/20

A. Paper part

1. Introduction

The tree shown below is the fifth example from the shared PDF file. In this case, of zero sum game D1 maximizes and D2 minimizes. Set of numbers representing profits of each players was generated randomly.

2. Process of problem solving



D2 / D1	L	P
L	6	8*
P	9	4

D2 / D1	L	P
L	1	7
P	5*	6

D2 / D1	L	P
L	8	4*
P	3	2

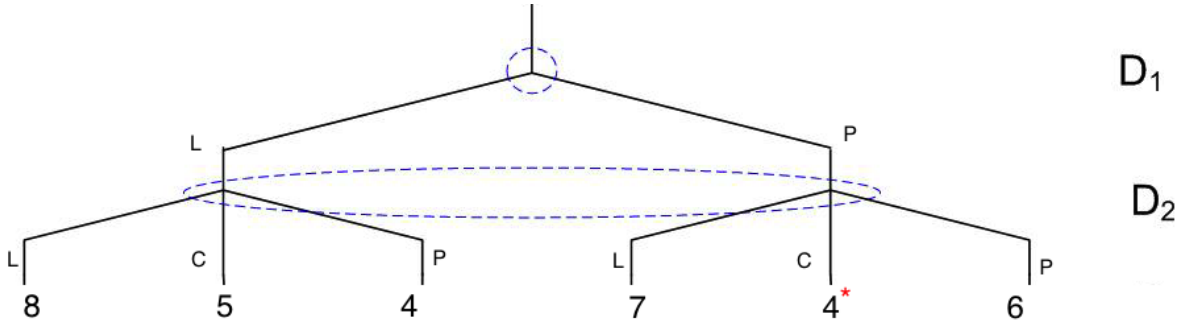
D2 / D1	L	P
L	3	5
P	7*	8

D2 / D1	L	P
L	4*	6
P	1	2

D2 / D1	L	P
L	6*	8
P	2	7

$$d_{21o}\begin{cases} P\ if\ \ u_{11}=L, & u_{12}=C \\ P\ if\ \ u_{11}=P, & u_{12}=L \\ L\ otherwise \end{cases} \qquad d_{22o}\begin{cases} P\ if\ u_{11}=L, u_{12}=L, u_{21}=P \\ P \qquad\quad if\ u_{11}=L, u_{12}=P \\ L\ otherwise \end{cases}$$

3. Finally obtained tree



D2 / D1	L	C	P
L	8	5	4
P	7	4*	6

Saddle point of considered decision tree : 4

Conclusions:

- Zero sum game represents the way in which each player tries to play safely.
- Due to the order of players moves, opponents have only partial ability to achieve the planned solution.
- Every strategy increasing one player's profit reduces the payout of the remaining player.

B. Code part

The solution of code part of laboratory was developed in Python language with external Numpy library (for calculations and matrices). Therefore, the exe file size exceeds the allowable size for PZE. Below is the link from which you can download the .exe file and .py file of source code. Exe file needs to be unarchived to work. Also all source code with an example of the effect of work is included below.

Link:
https://drive.google.com/file/d/1MNPr68KsCfp0Z5T6HnSdzfw8xWAI5j4c/view?usp=sharing

Source code:

```
import numpy as np

class ZeroSumGames():

    def __init__(self):
        self.final_matrix=None
        self.matrix=None
        self.n=0
        self.saddle_points_list=[]

    def saddle_point(self,matrix):
        #Finds saddle point of given matrix.
        D1=matrix.sum(axis=1).argmax()
        #D1 maximizes. Returns row coordinate of the maxmin element.
        D2=matrix.sum(axis=0).argmin()
        #D2 minimizes. Returns column coordinate of the minmax element.
        matrix_saddle_point=matrix[D1,D2]
        return matrix_saddle_point

    def create_arrays(self,n):
        #Creates n-
random matrix of size 2x2 and calculate their saddle points.
        #The result of this function is a list of saddle points.
        for i in range(n):
            A=np.random.randint(100,size=(2,2))
            print(i+1,"randomly generated matrix of size 2x2 is {}".format(A))
            self.saddle_points_list.append(self.saddle_point(A))
        print("Saddle points of generated matrices are {}".format(self.saddle_
points_list))
        return self.saddle_points_list

    def result(self):
        self.create_arrays(6)
        #Creates 6 random matrices according to the last(5th) tree structure.
        self.final_matrix=np.empty((2,3))
        #Creates final matrix of dimension obtained for given tree.
        with np.nditer(self.final_matrix, op_flags=["readwrite"]) as it:
            pos=0
            for x in it:
                x[...]=self.saddle_points_list[pos]
                pos+=1
        #Insert saddle points into the final matrix.
        print("\nThe final matrix formed from the saddle points of each matrix
 {}".format(self.final_matrix))
        print("\nThe result of considered decision tree is {}".format(self.sad
dle_point(self.final_matrix)))

if __name__ == "__main__":
```

```
    decision_tree=ZeroSumGames()
    decision_tree.result()
    input("Press Enter to continue...")

    pass

#SOLUTION OF RANDOMLY GENERATED EXAMPLE#

# 1 randomly generated matrix of size 2x2 is [[69  8]
#                                             [89 38]]
# 2 randomly generated matrix of size 2x2 is [[18 97]
#                                             [98 21]]
# 3 randomly generated matrix of size 2x2 is [[16 94]
#                                             [67 33]]
# 4 randomly generated matrix of size 2x2 is [[20 23]
#                                             [91 25]]
# 5 randomly generated matrix of size 2x2 is [[34 48]
#                                             [30 38]]
# 6 randomly generated matrix of size 2x2 is [[45 65]
#                                             [85 99]]
# Saddle points of generated matrices are [38, 98, 16, 25, 34, 85]

# The final matrix formed from the saddle points of each matrix [[38. 98. 16.]
#                                                                [25. 34. 85.]
]

# The result of considered decision tree is 38.0
```