AMERICAN UNIVERSITY OF BEIRUT

MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

Department of Industrial Engineering & Management

INDE 303 - Operations Research II - Stochastic Models & Applications
Final Project - Optimal Strategy for the PIG Game

By:

Ralph Mouawad

Dana Dahan

Lea Bou Sleiman

Hussein Daou

May 09th, 2023

To:

Dr. Maher Nouiehed

Contents

1 Introduction

The game known as PIG involves multiple players taking turns rolling a die, with the objective of being the first to reach a predetermined score (such as 100). During each turn, the player adds the value of the die to their score. However, if the player rolls a '1', they lose all the points they accumulated during that particular round. This creates a strategic decision for the players, as they must choose between rolling the die multiple times in an attempt to increase their score, or stopping after a successful roll to avoid losing everything.

2 Optimal Score for each turn

We want to determine the optimal point at which one should stop rolling the die. We must find the maximum value of 'x' for which rolling the die again would result in a loss of points. In other words, we need to find the point at which it is better to hold our score rather than risk losing points by rolling again.

To do this, we will try to find the expected number of points, one throw after scoring 'x', that will result in a score less than or equal to x.

$$E(\text{points after 1 turn of obtaining } x) \leq x$$

$$\tfrac{1}{6}(x+1) + \tfrac{1}{6}(x+2) + \tfrac{1}{6}(x+3) + \tfrac{1}{6}(x+4) + \tfrac{1}{6}(x+5) + \tfrac{1}{6}(0) \leq x$$

$$\tfrac{1}{6}(5x+15) \leq x$$

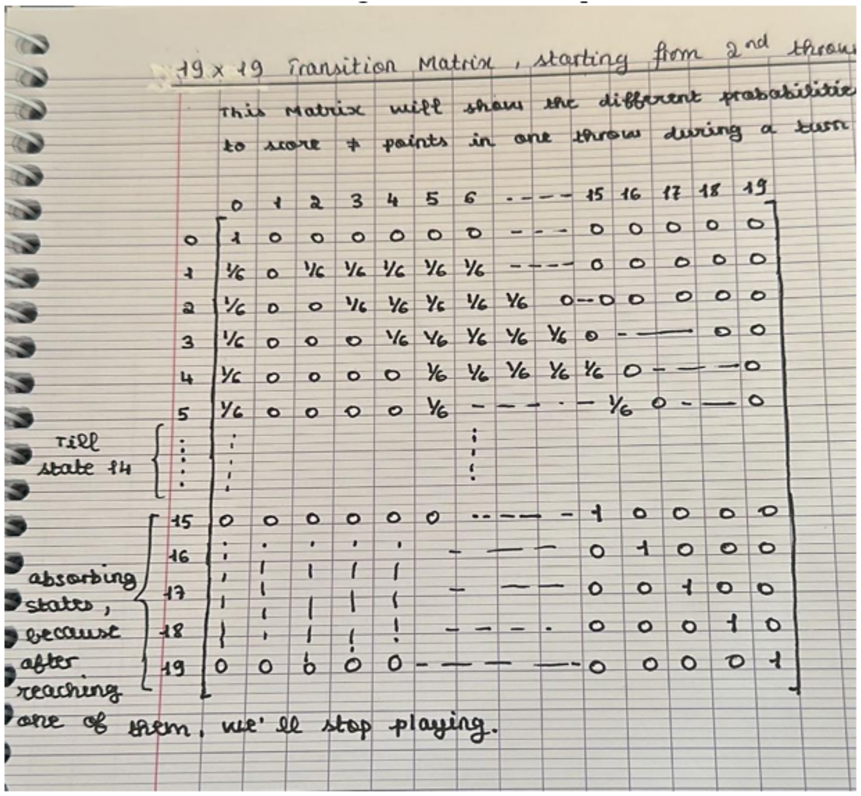
$$-\tfrac{1}{6}x \leq -\tfrac{15}{6}$$

$$x \geq 15.$$

Therefore, the maximum value of 'x' for which rolling the die again would result in a loss of points is 15. If our score is less than 15, it is better to roll the die again to try and improve our score. However, if our score is 15 or greater, it is better to hold our score to avoid the risk of losing points.

3 Markov Chain Transition Matrix

In this part, we are interested in modeling our game using a transition matrix. We will begin by computing a matrix that shows the probabilities of scoring different numbers on our second roll. We will then move on to the transitional matrix that displays the probabilities of reaching any score starting at a current score. This will be done using the holding method when we score 15 points or more during the turn.



19 x 19 Transition Matrix, starting from 2nd throw

This Matrix will show the different probabilities to score ≠ points in one throw during a turn

$$\begin{array}{c|cccccccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & \cdots & 15 & 16 & 17 & 18 & 19 \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
1 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & \cdots & 0 & 0 & 0 & 0 & 0 \\
2 & 1/6 & 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \;\; 0 \cdots 0 & 0 & & 0 & 0 & 0 \\
3 & 1/6 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 1/6 \;\; 1/6 \;\; 0 & \cdots & \cdots & \cdots & 0 & 0 \\
4 & 1/6 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 \;\; 1/6 \;\; 1/6 & 0 & \cdots & \cdots & \cdots & 0 \\
5 & 1/6 & 0 & 0 & 0 & 0 & 1/6 & \cdots & \cdots & 1/6 & 0 & \cdots & \cdots & 0 \\
\vdots & \vdots & & & & & & \vdots & & & & & & \\
15 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 1 & 0 & 0 & 0 & 0 \\
16 & \vdots & \vdots & \vdots & \vdots & \vdots & & \cdots & \cdots & 0 & 1 & 0 & 0 & 0 \\
17 & \vdots & \vdots & \vdots & \vdots & \vdots & & \cdots & \cdots & 0 & 0 & 1 & 0 & 0 \\
18 & \vdots & \vdots & \vdots & \vdots & \vdots & & \cdots & \cdots & 0 & 0 & 0 & 1 & 0 \\
19 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 & 1
\end{array}$$

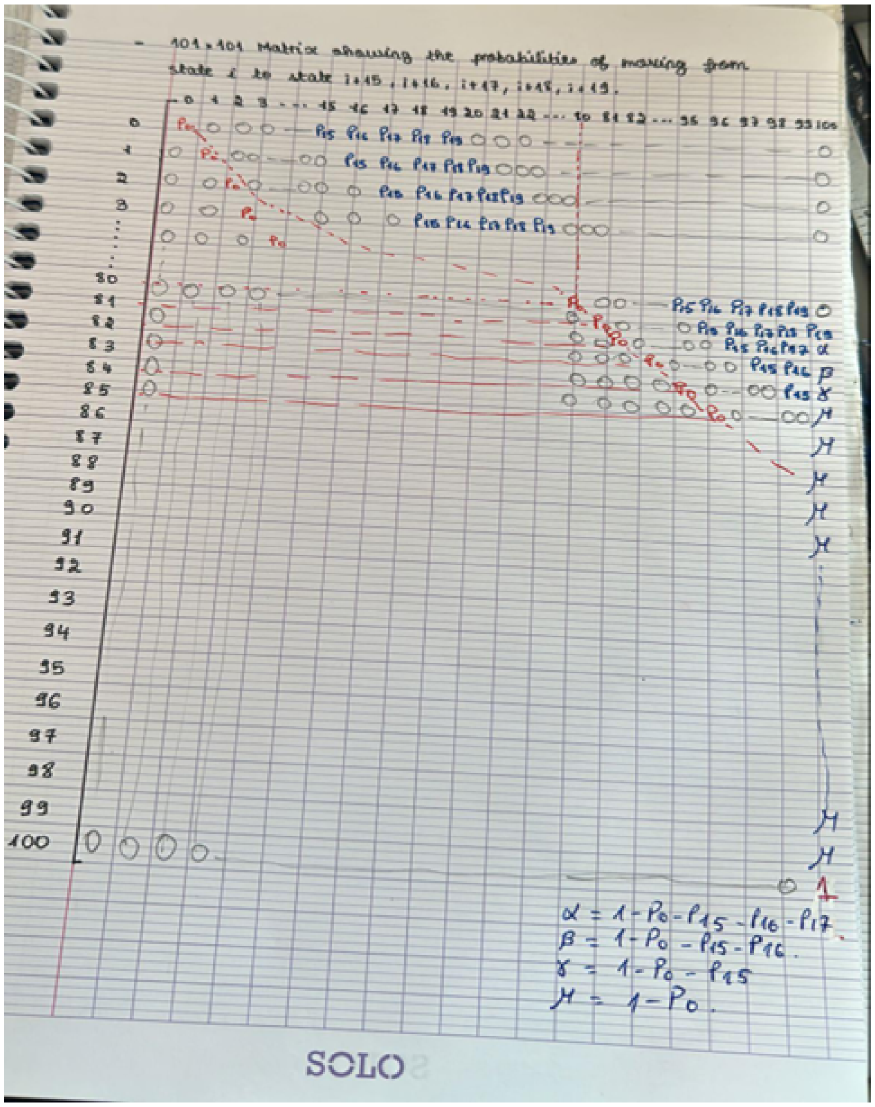
Till state 14

absorbing states, because after reaching one of them, we'll stop playing.

Figure 1: 19x19 transition matrix showing different states of points for the second throw

The chance of losing all our accumulated points on a single throw is 1/6, which occurs when we roll a 6. Likewise, there is a 1/6 probability of obtaining each of the scores 1, 2, 3, 4, or 5 in a single throw. Using the transitional matrix, we can determine how many points we can expect to gain during a turn. As previously established, we continue rolling the die until we have accumulated at least 15 points. Depending on the number of points we have before a roll, we may end up with 16, 17, 18, or 19 points. For instance, if we have 14 points, rolling a 5 on the next throw would give us

a total of 19 points for the turn. The states 15,16,17,18, and 19 are absorbing states because once we reach one of them, we'll stop playing because as we said before, it is better to stop playing once we have 15+ points. Let's move now to the 2nd transition matrix:
At each state 'j', we'll compute the probability of staying at state 'j' (score 0 point on this turn), and the probabilities of moving from state 'j' to: j+15, j+16, j+17, j+18, j+19.



- 101 x 101 Matrix showing the probabilities of moving from state i to state i+15, i+16, i+17, i+18, i+19.
$\alpha = 1 - P_0 - P_{15} - P_{16} - P_{17}$
$\beta = 1 - P_0 - P_{15} - P_{16}$
$\gamma = 1 - P_0 - P_{15}$
$\mu = 1 - P_0$

Figure 2: 101x101 transition matrix showing different states of points for each turn

We denote by:

P_0: probability of getting 0 points in that turn (remain in initial state).

P_{15}: probability of getting 15 points in that turn (from j to j+15).

P_{16}: probability of getting 16 points in that turn (j to j+16).

P_{17}: probability of getting 17 points in that turn (j to j+17).

P_{18}: probability of getting 18 points in that turn (j to j+18).

P_{19}: probability of getting 19 points in that turn (j to j+19).

Special cases: states 82+:

When we have 82 points, we can remain at state 82 with probability P_0 (score 0 point on that turn), or score 97 with probability P_{15}, 98 with probability P_{16}, 99 with probability P_{17}, or 100 with probability $1 - P_0 - P_{15} - P_{16} - P_{17}$, because getting 100 or 101 points won't matter here.

For 83 points: we score 100 points with probability $1 - P_0 - P_{15} - P_{16}$

For 84 points: we score 100 points with probability $1 - P_0 - P_{15}$

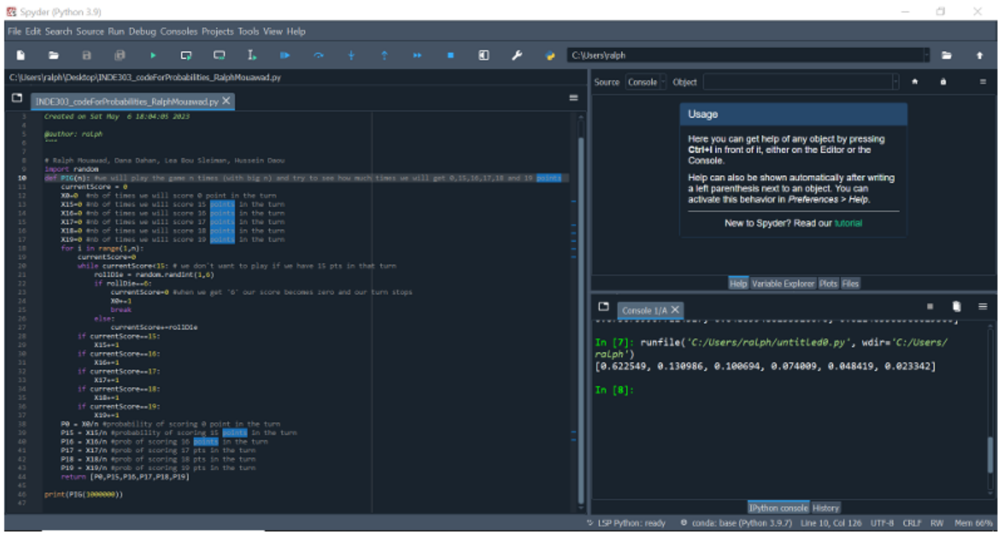
From 85 till 99 points: we score 100 points with probability $1 - P_0$.

State 100 is an absorbing state, because the game will stop. $P_{100,100} = 1$.

(The transitional matrix is presented in the excel sheet).

In order to compute P_0, P_{15}, P_{16}, P_{17}, P_{18}, and P_{19}, we simulated the game in Python.

We defined a function PIG(n), where n represents the number of turns we're playing the game. X_i for i={0,15,16,17,18,19} represents the number of times we score 'i' points in a turn. Each time, we set the current score to zero (which denotes the fact that we're starting a new turn) and generate a random integer between 1 and 6. If the number 6 is obtained, the current score will be set to zero, and $X_0 = X_0 + 1$. Otherwise, we keep rolling the die (here generating random numbers) and add the points to our current score until it is greater than or equal than 15 (our max number of points we want to attain during each turn). At the end, we check what score 'i' we got, and add it to the number of times we got X_i. After repeating this iteration n times, we computed the probabilities P_i = X_i/n {i = 0,15,16,17,18,19}. We played the game a large number of times to get more accurate results.



```python
Created on Sat May  6 18:04:05 2023

@author: ralph
"""

# Ralph Mouawad, Dana Dahan, Lea Bou Sleiman, Hussein Daou
import random
def PIG(n): #we will play the game n times (with big n) and try to see how much times we will get 0,15,16,17,18 and 19 points
    currentScore = 0
    X0=0  #nb of times we will score 0 point in the turn
    X15=0 #nb of times we will score 15 points in the turn
    X16=0 #nb of times we will score 16 points in the turn
    X17=0 #nb of times we will score 17 points in the turn
    X18=0 #nb of times we will score 18 points in the turn
    X19=0 #nb of times we will score 19 points in the turn
    for i in range(1,n):
        currentScore=0
        while currentScore<15: # we don't want to play if we have 15 pts in that turn
            rollDie = random.randint(1,6)
            if rollDie==6:
                currentScore=0 #when we get '6' our score becomes zero and our turn stops
                X0+=1
                break
            else:
                currentScore+=rollDie
        if currentScore==15:
            X15+=1
        if currentScore==16:
            X16+=1
        if currentScore==17:
            X17+=1
        if currentScore==18:
            X18+=1
        if currentScore==19:
            X19+=1
    P0 = X0/n #probability of scoring 0 point in the turn
    P15 = X15/n #probability of scoring 15 points in the turn
    P16 = X16/n #prob of scoring 16 points in the turn
    P17 = X17/n #prob of scoring 17 pts in the turn
    P18 = X18/n #prob of scoring 18 pts in the turn
    P19 = X19/n #prob of scoring 19 pts in the turn
    return [P0,P15,P16,P17,P18,P19]

print(PIG(1000000))
```

```
In [7]: runfile('C:/Users/ralph/untitled0.py', wdir='C:/Users/ralph')
[0.622549, 0.130986, 0.100694, 0.074009, 0.048419, 0.023342]

In [8]:
```

We obtained the following probabilities:

$P_0 = 0.622549$

$P_{15} = 0.130986$

$P_{16} = 0.100694$

$P_{17} = 0.074009$

$P_{18} = 0.048419$

$P_{19} = 0.023342$

The python code will be submitted with the report.

4 Expected number of turns per player

We want now to compute the expected number of turns per player until we reach a score of 100.

We define:

$E(N_i)$: Expected number of turns to reach 100 points starting from i. We know that in general:

$$E(N_i) = 1 + E(N_i)P_0 + E(N_{i+15})P_{15} + E(N_{i+16})P_{16} + E(N_{i+17})P_{17} + E(N_{i+18})P_{18} + E(N_{i+19})P_{19}$$

In other words, when we play one turn, we'll either stay in $E(N_i)$ with probability P_0 because we got a '6' somewhere, or move to $E(N_{i+15,16,17,18,19})$ multiplied by $P_{15,16,17,18,19}$. We added +1 because we played one turn to arrive at one of these states.

For Example:

$$E(N_0) = 1 + E(N_0)P_0 + E(N_{15})P_{15} + E(N_{16})P_{16} + E(N_{17})P_{17} + E(N_{18})P_{18} + E(N_{19})P_{19}$$

$$E(N_{15}) = 1 + E(N_{15})P_0 + E(N_{30})P_{15} + E(N_{31})P_{16} + E(N_{32})P_{17} + E(N_{33})P_{18} + E(N_{34})P_{19}$$

Special Cases:

At 81:

$$E(N_{81}) = 1 + E(N_{81})P_0 + E(N_{96})P_{15} + E(N_{97})P_{16} + E(N_{98})P_{17} + E(N_{99})P_{18} + 1 \cdot P_{19}$$

At 82:

$$E(N_{82}) = 1 + E(N_{82})P_0 + E(N_{97})P_{15} + E(N_{98})P_{16} + E(N_{99})P_{17} + 1 \cdot (1 - P_0 - P_{15} - P_{16} - P_{17})$$

At 83:

$$E(N_{83}) = 1 + E(N_{83})P_0 + E(N_{98})P_{15} + E(N_{99})P_{16} + 1 \cdot (1 - P_0 - P_{15} - P_{16})$$

At 84:

$$E(N_{84}) = 1 + E(N_{84})P_0 + E(N_{99})P_{15} + 1 \cdot (1 - P_0 - P_{15})$$

At 85:

$$E(N_{85}) = 1 + E(N_{85})P_0 + 1 \cdot (1 - P_0)$$

At 86 and beyond:

$$\ldots$$

At 99:

$$E(N_{99}) = 1 + E(N_{99})P_0 + 1 \cdot (1 - P_0)$$

At 100:

$$E(N_{100}) = 0$$

At 81 points, if we score 19 points, we arrive at 100, so it costs us 1 turn.
At 83 points, if we score more than 16 points, we arrive at 100 and it costs us 1 turn, with probability $1 - P_0 - P_{15} - P_{16}$.
At 85 points, we can either remain at 85 or go directly to 100 points with probability $1 - P_0$, and it would cost us only 1 turn. It will be the same case up to 99 points.
Based on our general formula:

$$E(N_i) = 1 + E(N_i)P_0 + E(N_{i+15})P_{15} + E(N_{i+16})P_{16} + E(N_{i+17})P_{17} + E(N_{i+18})P_{18} + E(N_{i+19})P_{19},$$

as well as our special cases, we'll get a system of too many recursive equations, and to solve it, we will use code in Python.
We tried to compute the expected number at i and recursively call our function on $i + 15, i + 16, \ldots$. Unfortunately, our code didn't run, so we couldn't compute the final result, but it should be between 16 and 18.

5 Conclusion

In conclusion, we have determined an optimal strategy for playing the PIG game. It is best to aim for a score of 15 points per turn and stop there to avoid losing all accumulated points. We have computed various expected values, such as the number of turns required for each player to reach a score of 100 points.