1. Levels and goals

II. From Probability to Navigation

Giacomo Bergami

Newcaslte University

Objectives

- Understanding how probabilities can be used to model different game properties:
  1. Player's skills, **PvP**: Simplified Tennis (PONG)
  2. Player's chances at advancing the game: Snakes and Ladders.
- Using stochastic models for establishing the interconnection between probabilities and time:
  1. Which is the probability of ending/winning the game in n turns?
  2. Which is the average number of turns required for ending the game?
- Understanding how changes in the model and/or its probability distribution alters the previous questions, and their effect on gameplay.

Game Modelling Goals

Pacing & Statistic Analysis

Pacing is a fundamental analysis in Game Balancing:

- The aim of the game is to put obstacles and power-ups with trade-offs.
- Obstacles should never be insurmountable or unfair.

Statistical Analysis is often used to analyse games for balancing:

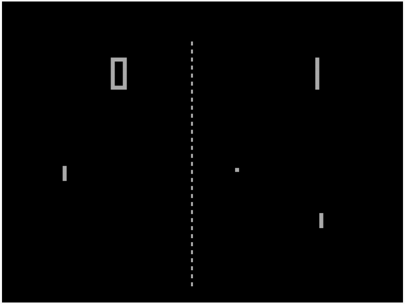
- We can identify faults (e.g., unbalanced areas) and make corrections
- The usual approach is to collect *logs* from real plays, and then infer statistics via *post-mortem analysis*.
- Still, is it possible to analyse the game play before sending the product to testers and/or sending it to production?

Chances vs. Skills

- Chance allows a weaker player to beat a stronger one
- The outcome of the game should be mainly influenced by the skills
- Still, the skills of a player are hard to assess correctly.
- As a first approximation, we can model the player's skills as the chances of winning:
  - In a simplistic game, a player could only perform two actions: one, advancing the game towards the next turn, and the other making the player fail thus letting the opponent win.

Player versus Player

Simplified Tennis (PONG) (1/3)

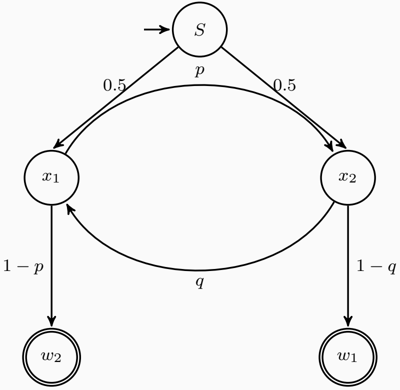


- One single graph represents one single (tennis) play (i.e., *round*).
- One of the two players, the *server*, starts the game (2 possible initial states). This player is randomly picked (S).
- We have 2 states identifying one of the two players actively playing (x_1 and x_2), and 2 states (w_1 and w_2) declaring the winning for the given tennis game (e.g., *round*).
- The transition probabilities (p, q) represent the player's expertise in performing the correct action

Simplified Tennis (PONG) (2/3)

- The game is *memoryless*: the ability of responding to a service is completely unrelated to the previous move.
- We can represent such a game via a probabilistic process, namely a Discrete Time Markov Chain (DTMC), represented as a transition matrix T.
- Such matrix can be graphically represented as a weighted directed graph.

Simplified Tennis (PONG) (3/3)



$$T = \begin{array}{c} \\ S \\ x_1 \\ x_2 \\ w_1 \\ w_2 \end{array} \begin{array}{c} \begin{array}{ccccc} S & x_1 & x_2 & w_1 & w_2 \end{array} \\ \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & p & 0 & 1-p \\ 0 & q & 0 & 1-q & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

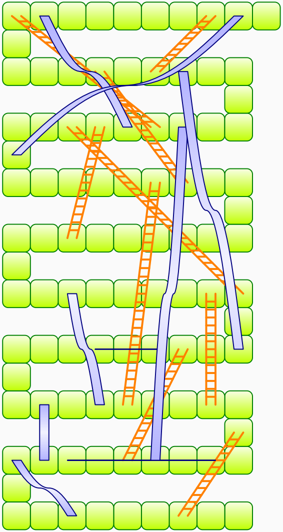
- **State Space**: Determining the server (S), Player1's turn (x_1), Player2's turn (x_2), Player 1 wins (w_1), Player 2 wins (w_2).
- **Initial State**: S.
- **Actions & Transitions**:
  1. $P1$ serves, $S \xrightarrow{0.5} x_1$
  2. $P2$ serves, $S \xrightarrow{0.5} x_2$
  3. $P1$ responds, $x_1 \xrightarrow{p} x_2$
  4. $P2$ responds, $x_2 \xrightarrow{q} x_1$
  5. $P1$ does not catch, $x_1 \xrightarrow{1-p} w_2$
  6. $P2$ does not catch, $x_2 \xrightarrow{1-q} w_1$
- **Goal Test**: Either w_1 or w_2 are reached.

Player versus Environment

Player versus Environment (PvE)

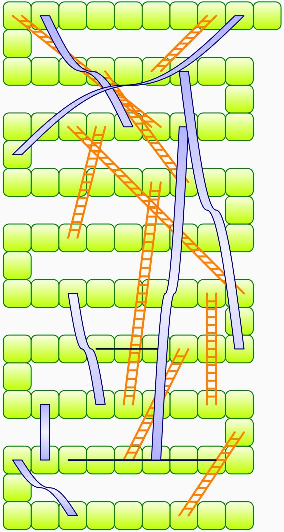
- They are also referred as *One-person games*, or *Games against nature*.
- Before tackling the problem of an interaction with another agent, we should first ask ourselves whether we are able to model agent that is capable of taking meaningful decisions:
  - No decision is with no effect,
  - but, one among them, will lead to the optimal solution.
- This person may take the wrong decision, but there does not exist a conscious opponent.
- One-person games are greatly investigated in AI (robotics) and in Planning Theory.

Snakes and Ladders (1/3)



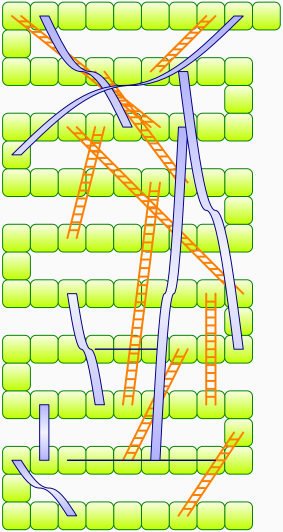
- The game contains 100 cells:
- The player starts the game from the first cell.
- **Actions**:
  1. At each turn, the player rolls a die:
  2. if they step into the lower end of a ladder, the player progresses towards its upper end;
  3. if they step into the upper end of a snake, the player backtracks towards its lower end;
  4. otherwise, the player stays in the reached state.
- The game ends when the last cell is reached.

Snakes and Ladders (2/3)



- The game is *memoryless*: at a given point in the game, the player's progression from the current square is independent of how they arrived at that square.
- Each edge weight is now independent from each player's skills.

Snakes and Ladders (3/3)



Without Snakes and Ladders:

- Each non-rigged die generates numbers in $I = \{ 1, 2, 3, 4, 5, 6 \}$ with probability $\mathbb{P}(I = i) = {}^1\!/_6$ for each $i \in I$.
- Given a initial cell i, I can only move from i to one of the following states: $\{ i+1, \ldots, i+6 \} \setminus \{ n \in \mathbb{N} \mid n > 100 \}$.

With Snakes and Ladders:

- arriving to a starting point of a snake/ladder from cell i has 0 probability,
- while the probability of reaching the end of the snake/ladder from cell i is increased by ${}^1\!/_6$.

Measuring easiness through probability and time

Average Hitting Time

The *average hitting time* determines the mean number of turns x_i required to reach one of the goals for the first time from state i. We can prove that this reduces to solve the following *linear (equation) system* for all the states i:

$$\begin{cases} x_i = 0 & i \text{ goal} \\ -x_i + \sum_{j \neq i} T_{ij} x_j = -1 & \text{otherwise} \end{cases}$$

This reduces to solve a linear system $A\vec{x} = \vec{b}$, where:

$$\vec{b}_i = \begin{cases} 0 & i \text{ goal} \\ -1 & \text{otherwise} \end{cases} \qquad A_{ij} = \begin{cases} 1 & i = j, i \text{ goal} \\ -1 & i = j, i \text{ not goal} \\ 0 & i \neq j, i \text{ goal} \\ T_{ij} & \text{otherwise} \end{cases}$$

Average Hitting Time: Snakes and Ladders (1/2)

Given that our problem has only one initial state, cell$=1$, we are interested in x_1.

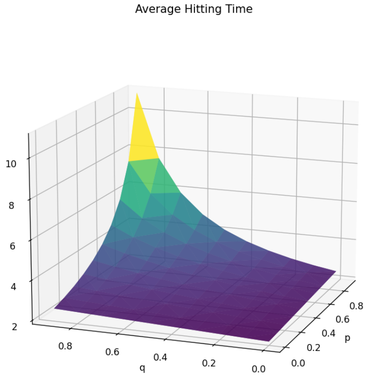
The following result confirm the intuitive results in length of game-play from performing *random walks* lover the stochastic process:

- Without Snakes and Ladders: ~ 29 turns.
- With only Ladders: ~ 18 turns.
- With only Snakes: ~ 46 turns.
- With both Snakes and Ladders: ~ 25 turns.

Average Hitting Time: PONG

By solving the system, we obtain the following *Average Hitting Time*:

$$x_s = \frac{4 + p + q - 2pq}{2 - 2pq}$$



We now exploit *average hitting time* for determining how log one single round of the PONG game will take dependently from the success probabilities p and q for both players.

- If only one player is strong, it will take as little as ~ 2 turns after tossing the coin.
- The more experienced the two players are, the longer it will take!

Probability of winning in n turns

Initial Probability Distribution μ:

- Given that in our game we have only one initial state, S, we will have an **initial probability distribution** μ as an empty vector except for $\mu_S = 1$.

Accepting States/Goal Tests:

- Given that the two states declaring the winner, w_1 and w_2, have no outgoing edges and given that all the intermediate states can reach such a state, we want to assess the probability to reach such states.

Reachability Probability at step n:

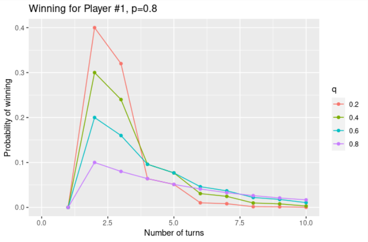
- μT returns a vector $\mu^{(1)}$, where $\mu_i^{(1)}$ indicates the probability of reaching a state i in one step.
- The probability of Player1 winning in 1 turn is $\mu_{w_1}^{(1)} = 0$.
- The probability of reaching any game configuration in two steps is $\mu^{(2)} = \mu^{(1)} T = \mu T T = \mu T^2$:
- so, the probability of Player1 winning in n turns is given by $[\mu T^n]_{w_1}$

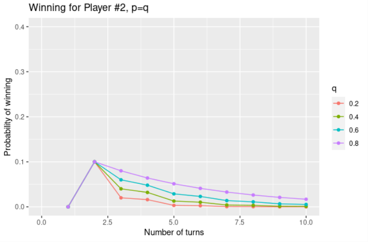
Probability of winning in n turns: PONG (1/2)

In the previous slide we discussed how long should it take to end the game when p and q vary:

- This gives no information of the chances of winning of one player in a given amount of turns.
- To simplify the analysis, we assume that Player1 is an expert, $p = 0.8$, while we test varying opponent (Player2) skills.
- Furthermore, we want to analyse the probability of Player1 and Player2 of winning after n turns in the game:
- This can be assessed by assessing the Reachability Probability at step n.

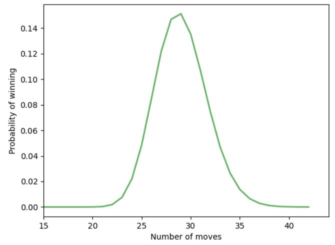
Probability of winning in n turns: PONG (2/2)



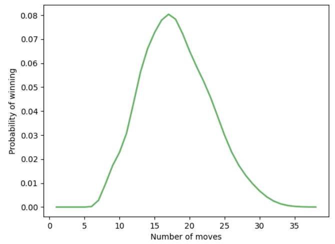


- We need to first toss the coin before assessing a winning probability.
- The two players have higher chances of winning at the first stages of the game.
- When both opponents have the same skills ($q = 0.8$), they both have the same distribution of chances of winning.
- The lower the skills for Player2, the higher are the changes of winning for Player1.

Snakes and Ladders: Probability of winning in n turns (1/2)



Without Snakes and Ladders.

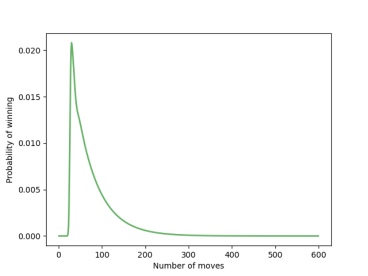


With Ladders.

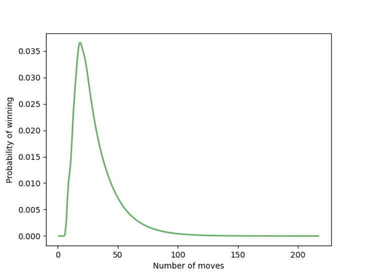
- We need at least 17 turns to win the game.
- It is more likely to win the game with probability 0.15 after 29 turns.

- The game fastens up, we now need at least 6 turns to win the game.

Snakes and Ladders: Probability of winning in n turns (2/2)



With Snakes.



With Snakes and Ladders.

- The game slows down, as we might get stuck in a loop.
- Furthermore, the overall chances of winning decreases.
- Adding back the ladders, the chances of winning increase as well as the length of loops decrease.

Wrap Up: Snakes and Ladders

This simple game allows us to determine how power-ups (*ladders*) and penalties (*snakes*) might affect the duration and the probability of reaching the end of the game in a given amount of steps:

- By only adding ladders, the time for the game-play is considerably reduced:
  1. We generate fewer and shorter paths, but there might be multiple possible way to get paths of the same length:
  2. The probability of reaching the final state in a given amount of steps is reduced.
- By only adding snakes, the time of the game-play is considerably increased:
  1. Even if it is extremely improbable, players might get stuck in loops, thus increasing the length of the possible paths;
  2. Therefore, the probability of reaching the end of the game in a given amount of steps if considerably decreased.
- Intuitively, adding both snakes and ladder will provide a trade-off situation between possible length of game-play and probability of reaching the end of the game.

Lessons Learned

Creating a game: Balancing Process

While assessing the game balance, we should proceed as follows:

1. First, we should create a balanced game:
   - *E.g., we should generate a board with neither snakes nor ladders.*
2. Next, we might consider the average running time (*average hitting time*) and the probability of winning the game in a given number of steps.
   - *Collect such values as reasonable outcomes for the game.*
3. Introduce minimal changes, so to always determine how single changes affect the overall system.
4. Compare the values obtained with the previous configuration, which can be thought as reasonable if returns similar or better values than the previous configuration.
5. If the game configuration is reasonable, go back to the 3rd item and continue to refine the game.

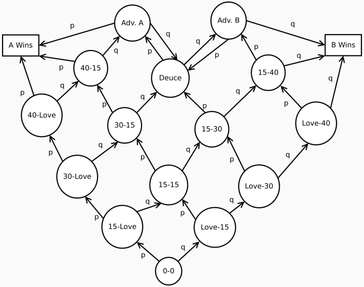
More complex games

Monopoly

The proposed methodology can be also applied to board games such as Monopoly:

- The real Monopoly is not a stochastic model:
  - if a pawn is in jail, the person in control might choose whether to stay in jail if doubles are not thrown.
  - if a pawn has been in jail two turns, it must be moved out of jail.
- This is because the current models are *memoryless*. There are two possible directions to be taken:
  - Force the process to be memoryless, and change the jail rule.
  - Simulate the notion of memory by exploding the number of possible states.
- Still, simplification of the model will give you a first intuition of how the real game might actually work.

Semi-Realistic Tennis



- Each PONG process could be one single nested process within the diagram on the left.
- Please observe: from theory, nested process are equivalent to un-nested ones, so:
  1. Replace each circle state c from the left with a whole PONG game instance, where
  2. the ingoing edges to w_1 (and w_2) state have now the same target ot $c \xrightarrow{p} c'$ (and $x \xrightarrow{q} c''$), and
  3. all the ingoing edges to a circle are redirected to each node S.
  4. Remove the unconnected w_1 and w_2.
- **Exercise**: compare the results of PONG with the ones with Semi-Realistic Tennis.

Conclusions

This section introduced the key concepts to understand the relevance of determining the *average hitting times* and the *probability of terminating a game in a given amount of steps*. We also showed how changes in probability affect the overall game duration.