Lab session 5: Multi-agent interaction

Pierre-Alexandre Murena

May 3, 2024

1 Instructions

- This practical session is evaluated, and the grade will count as 30% of the final grade for the course.
- Working in small groups (2-3 max) is possible, but it is expected that each student submits their answers (no group submission accepted).
- Submission must be done in the form of a zip file, containing the code files and a report (PDF). The name of the zip file must be of the form `homework_NAME.zip`, where `NAME` is your family name.
- The report must be named `report_NAME.pdf` and must contain all the answers to the questions. Do not forget to write your name in the report.
- Ideally, reports are written in LaTeX, but other formats are accepted, as long as they are clean and easily readable: in particular, you can scan your hand-written solutions.
- In case you have figures in the report, feel free to use photos of hand-drawn figures, again as long as they are clean and easily readable.
- Add comments to your code that explain its behaviour in detail.
- The code will be evaluated using some tests that are not given to you.
- Your solutions must be submitted by May 24, 23:59. All homeworks submitted after this deadline will be ignored and considered as not submitted.
- The zip files must be submitted on the dedicated directory on StudIP.
- You have 3 weeks in total. However, the questions are not simple, so I strongly invite you to work on it as early as possible.

2 Nash equilibria (not graded)

Question 1. Find the pure Nash equilibria (Nash equilibria in pure strategies) of the following bi-matrix games:

$$G_1 = \begin{pmatrix} 1,3 & -1,5 & 4,2 \\ 0,2 & 7,3 & 0,1 \\ -2,1 & 1,0 & 5,1 \end{pmatrix} \qquad G_2 = \begin{pmatrix} 5,0 & 2,1 & 0,5 \\ 4,4 & 3,0 & 0,3 \\ 3,2 & 4,1 & 2,4 \end{pmatrix}$$

Question 2. Find the Nash equilibria (in pure and mixed strategies) of the following game:

$$\begin{pmatrix} 5,2 & 1,1 \\ 3,1 & 2,3 \end{pmatrix}$$

3 Application questions (5 points)

Question 3 (1 point)**.** Find the pure Nash equilibria of the following bi-matrix game:

$$G = \begin{pmatrix} 2,4 & 4,1 & -1,2 & 7,3 \\ 1,1 & 4,4 & 5,1 & 2,5 \\ -1,1 & 3,2 & 3,2 & 4,4 \end{pmatrix}$$

Question 4 (1 point)**.** Find the Nash equilibria (in pure and mixed strategies) of the following game:

$$\begin{pmatrix} 4,3 & 3,-1 \\ 3,0 & 5,2 \end{pmatrix}$$

Question 5 (3 points)**.** Consider a grid world of 2 by 2 cells (2 rows, 2 columns). The agent is located on one of the four cells and aims to collect a coin that can be located to any of the four cells. The agent does not know the position of the cell and cannot observe its own position. However, it can see the content of its cell (in particular it sees if the cell contains the coin). The agent can perform four actions: Left, Right, Up, Down, which makes it move to the corresponding neighborhood cell (or stay in its cell, in case it is at the border).

Write the full transition model for the corresponding belief state search problem, in the form of a graph.

4 Non-zero-sum games in extensive form (14 points)

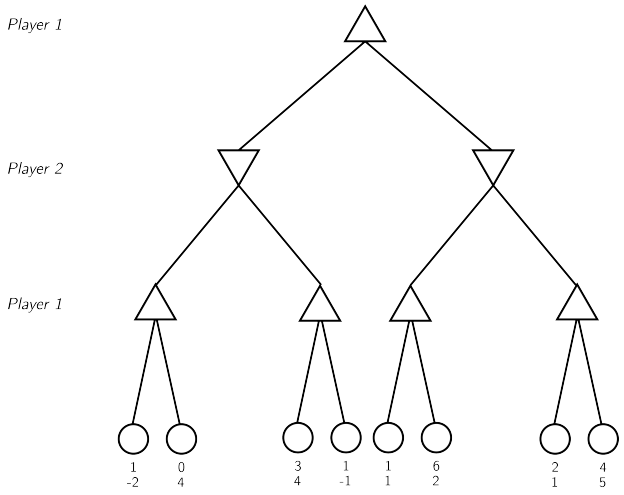
In zero-sum games, Player 2 loses what Player 1 wins. For this reason, the minimax algorithm assumes that Player 1 *maximizes* the payoff and that Player 2 *minimizes* the payoff. The goal of this problem is to implement a search in the general case.

In the general case, terminal nodes of the graph contain the utility for all players. The goal for each player is to maximize their own utility.

4.1 Examples (4 points)

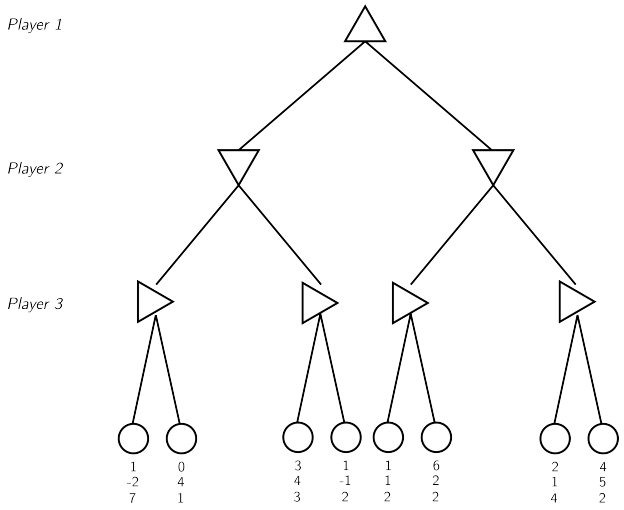
In order to understand how the algorithm would work in the general case, we propose to analyze two simple examples first

We first consider the following sequential 2-player game.



Question 6 (3 points)**.** Using backward induction, what is the optimal strategy for each player? Write the game in a normal form game and show that the strategy obtained with backward induction is indeed a Nash equilibrium.

We now consider here the following sequential 3-player game.



Question 7 (1 point)**.** Using backward induction, what is the optimal strategy for each player?

4.2 Algorithm (6 points)

In the file `ones_game.py`, you will find the implementation of a game in extensive form, with total information (deterministic, no simultaneous action). Unlike for the adversarial setting (zero-sum game), the players may receive different absolute utilities in the final state.

In this game, players are asked to choose 0 or 1, the goal being to play as many 1's as possible. The game is organized in rounds, in which Player 1 plays first, and Player 2 plays next. If the two players play a 1 in a same round, their utility decreases by 1. They also receive a bonus of 1 every time they play a 0 after a 1 or a 1 after a 0.

Since it is not a zero-sum game, this game cannot be solved by minimax.

Question 8 (5 points)**.** In the function `solve` of file `game.py`, adapt the code of the minimax algorithm to solve games in extensive form in this more general case. Test it on the proposed game.

This method performs some depth-first search in the game tree and requires to explore all terminal nodes. For the minimax algorithm, we had introduced alpha-beta pruning as a method to prune the game tree and to explore relevant branches only.

Question 9 (1 point)**.** Can alpha-beta pruning be applied directly? Justify your answer.

4.3 Application: The two monkeys and the banana (4 points)

We consider a variant of the monkey and banana problem, with two monkeys but no box. The game works as follows:

- Monkeys act sequentially, one after another (Monkey 1, Monkey 2, Monkey 1 etc).
- Each monkey is at a given position (x_1 and x_2) in the grid.
- The available actions are:
  - Move left: moves to the left position. If the other monkey is there already, then the agent climbs on the other monkey. If the agent is on top of the other monkey, it goes down. If the other monkey is on top of the agent, the action is not available.
  - Move right: moves to the right position. If the other monkey is there already, then the agent climbs on the other monkey. If the agent is on top of the other monkey, it goes down. If the other monkey is on top of the agent, the action is not available.
  - Grab: tries to grab the banana. Leads to a terminal state if the agent is on top of the other monkey at the same position as the banana.
  - Nothing: the agent does nothing.
- The goal is to grab the banana in less than N steps.
  - If either of the two monkeys grabs the banana before step N, both monkeys receive a payoff of 1.
  - If none of the monkeys manages to grab the banana before step N, both monkeys receive a payoff of 0.

This game is implemented in file `monkeys.py`.

Question 10 (2 points)**.** For a room of length $L = 4$, with banana located at $l = 2$ and monkeys initially located in $x_1 = 1$ and $x_2 = 4$, and having horizon $N = 7$, what is the optimal strategy for the two agents? Use the code of Question 8.

In this first version, the monkeys collaborate because they both benefit from a victory.

Question 11 (1 point)**.** Change the utility of the monkeys as follows:

- If a monkey grabs the banana, it gets a reward of 1 and the other monkey gets a reward of 0
- If none of the monkeys manages to grab the banana before step N, both monkeys receive a payoff of 0.

What is the new strategy?

Question 12 (1 point)**.** Change the utility of the monkeys as follows:

- If a monkey grabs the banana, it gets a reward of 2 and the other monkey gets a reward of 1
- If none of the monkeys manages to grab the banana before step N, both monkeys receive a payoff of 0.

What is the new strategy?

5 A simple card game (11 points)

In this section, we study a simple 2-player card game, the goal of which is to show the card with the highest values. The rules of the games are as follows:

- We have N cards, with values from 1 to N.
- We first shuffle the cards.
- Player 1 first picks k cards
- Player 2 then picks k cards.
- Player 1 chooses whether to play:
  - If P1 refuses to play, P1 receives a payoff of -1.
  - If P1 chooses to play, P2 chooses whether to play:
    - If P2 refuses to play, P1 receives a payoff of 1.
    - If P2 accepts to play:
      - P2 picks a new card from the deck (if any)
      - P1 and P2 show their highest card
      - The player with the highest card receives a payoff of $\alpha \geq 1$.

The game is a **zero-sum game**. The parameters of this game are: N, the total number of cards, k, the number of cards initially picked by each player, and α, the payoff in case of victory.

5.1 Full observability (2 points)

Question 13 (2 points)**.** What is a necessary and sufficient condition on N, k and α for the game to be fully observable by both agents? In that case, formalize the game as an adversarial search problem and give the minimax strategy[1].

5.2 Solution in case of partial observability (9 points)

We now consider the case of partial observability: in this case, none of the player knows which cards are possessed by the other player, nor which cards are in the deck.

A solution to solve this problem is to execute a minimax search for each possible shuffling of the cards. For each shuffling, assuming that the shuffling is known *to both players*, we can compute the value of the game. For each initial action a of Player 1, the expected payoff is then:

$$\mathbb{E}[V(a)] = \sum_{\sigma} P(\sigma) V_{\sigma}(a)$$

where σ is a shuffling and $V_{\sigma}(a)$ is the payoff of the game for shuffling σ if Player 1 plays a.

Question 14 (3 points)**.** If no other information is available, what is $P(\sigma)$? Implement a method computing the value $V_{\sigma}(a)$ and the expected value $\mathbb{E}[V(a)]$.

Question 15 (1 point)**.** What is the number of possible shuffles of N cards? Comment on the potential impact on the computation of $\mathbb{E}[V(a)]$.

In practice, we prefer sampling a subset of permutations randomly:

$$\mathbb{E}[V(a)] \simeq \frac{1}{p} \sum_{i=1}^{p} V_{\sigma_p}(a)$$

This is called a **Monte-Carlo approximation**.

Question 16 (3 points)**.** Implement the method `solve_MC` which returns the approximation of the expected value for all a. Note that the method takes as input the variable p, which is the number of shuffles to sample. For $N = 5$ and $k = 1$, show and discuss the influence of p on the quality of the approximation and on the final decision.

Question 17 (2 points)**.** Discuss *empirically* the impact of the victory payoff α. You can show in particular how it impacts the decision to take risks for Player 1.

5.3 Bluff (6 bonus points)

Imagine now that Player 1 is a cheater and could observe one card of Player 2 without being seen.

Question 18 (1 point)**.** How would you change the method to incroporate this piece of information? [2] Would it be the same if Player 2 had selected the card to show?

[1] The question is not to implement it, but to find the solution theoretically.

[2] We don't ask you to code this modification, but if you want to do it, you will be rewarded with 1 bonus point.

Question 19 (1 point)**.** In case Player 2 has the highest card, is it in their interest to let Player 1 know about it?

We now consider a variant of the game where Player 2 first shows one of their cards.

Question 20 (Bonus, 4 points)**.** Implement a solution for Player 2.