Assignment 9

Subject

Topics of this session :

1. Game Theory.

This assignement is graded and must be submitted (individually) on Moodle before next week's class.

For each exercise, detail your reflexion steps :
- We are mostly interested in your actual thinking process.
- Even if you are unable to solve an exercise, write out what were you reflexion steps.
- For each attempted exercise, a written feedback will be provided (if time alllows it).

For coding :
- Noto (Online Jupyter NoteBook).
- Any other python coding environment you prefer using.

Exercise 1

Give an example of a 2x2 game with no dominating strategy for either player.

Exercise 2

Take the example where we make an agreement about how to split the money if we win at a game. Assuming my committment is true, and that if we collaborate, then our winning probability is 50%, then what is the optimal strategy for each player ? The utility for each player is :
- If we play and lose, I get 0, and you get some amount c (the joy of playing)
- If we play and win, and we agreed to split the money with a ratio R, I get R, and you get $-R + c$.
- If we don't play, we get 0.

To obtain the optimal solution, define the game as a 2-stage decision process.

Project – Part 6

This whole section must be done in groups of 2-3 people.

This week, we will model our problem into an MDP.

Questions ? victor.villin@unine.ch

1 Guided Project (Dungeon Gridworld)

Previously, we assumed a specific path for the *Weird Thing*. This time, we will consider it as an adversarial agent :

— At all times, the *Weird Thing* will move deterministically, by naively moving toward us (no shortest path algorithm used).
— It can move over holes, but not through walls.
— If it is adjacent to our agent, it will stand still.

Task :

1. Using the code in `dungeon_mdp_adversarial.ipynb`, change how the *Weird Thing* behaves so that it is adversarial.
2. Interpret how the agent's behaviour change, varying the **position** of the *Weird Thing* and the **rewards**.

2 Personal Project

Task :

1. Modify your problem's Markov Decision Process so that it has **exactly one** other agent.
2. Make this agent deterministic and adversarial (its goal are opposite to ours).
3. What if the goals of the other agent are not the opposite of our goals ?
4. Explore what happens for different combinations of agent behaviour, and different combinations of what the agents think about each other (in terms of rewards for example).
5. Document what you are doing precisely, and how you encode different preferences for each agent. Produce plots explaining how the behaviour changes.

   Note : If your problem does not have another agent, try adding one and make it behave deterministically in function of the main agent.