Prof. Alfio Ferrara

Lab of Reinforcement Learning

Final project proposals

Introduction

The final project consists in the implementation of a RL solution to a simple problem. Students have to provide access to a GitHub repository containing the code and reproducible results. Finally, the project will be discussed after a 10 minutes presentation in English with slides. The project discussion will be set by appointment, according to the following procedure:

1. Subscribe one of the official exam dates
2. Submit the link to your GitHub repository as soon as the project is ready
3. Contact Prof. Ferrara and set and appointment for the discussion
4. The appointment for the exam part on theory should be taken directly with Prof. Cesa-Bianchi

The project is an individual work.

Miscellaneous Hints

All the projects are presented in a simplified version, suitable for both a **Tabular** or **Approximate** solution. **State and Action spaces are presented as discrete spaces,** respectively. However, it is **possible and appreciated to try to work on a more complex setting**, exploring **approximate solutions** and **continuous action and state spaces** if needed.

There are **no constraints on the technology**. You can use the gymnasium environment, the relearn environment, or also your framework developed from scratch.

Focus of the project is to **evaluate the student understanding of the problems** and the exam will include **questions about theory** starting from the proposed project.

Generally speaking, the project proposals are just **ideas that can be modified or just taken as suggestions** for developing your own objectives.

Try to have fun!

Project proposal 1: Play RISK!

We are playing modified version of Risk! and it's our turn. The game works as follows:

- There is a map of territories connected by borders that can be modeled as an undirected graph
- We occupy one initial territory with N troops; all the other territory are occupied by E enemy troops distributed uniformely over the territories
- We can choose two actions: ATTACK or STAY
  - STAY: our turn ends and the enemy attacks one of our territories with fewer troops starting from one of his territories with more troops bordering ours
  - ATTACK: from one of your territories with more troops you attack one of the bordering enemy territories with fewer troops
- ATTACK procedure:
  - The attacker rolls three six-sided dice (3d6). The defender rolls three six-sided dice (3d6). The attacker's best die is compared to the defender's best, second best to second best, and so on. The highest die wins. In case of a tie, the defender wins. For each confrontation, the losing side loses one troop. The attacker never loose the last troop, which means that the last troop of the attacker, even if the attacker looses the corresponding dice confrontation, is not lost.
  - If the attacker wins, they occupy the enemy territory with all their troops but one that remains in the the starting territory.
- The game ends after 10 turns.
- Each time you occupy an enemy territory you gain K victory points. Each time you loose a troop, you loose 1 victory point.

Ideas for extension: Model a more realistic game of Risk! against a automatic agent opponent.

Goals

1. Explore the possibility to model this game as an MDP and discuss the possible options. You can also try to change the rules and modify the setting if needed. Present an implementation and discuss your design choices.
2. According to your model, choose a strategy to learn a policy that optimizes your final reward in terms of victory points.
3. Discuss the role of the parameters N, E, and K in your solution.

Project proposal 2: Manage a farm

You are the manager of a farm. You have an initial budget of 2000 €. Each year you have to take some decisions about how to invest you money, but you can do only one of the following things:

1. Buy one sheep: a sheep costs 1000 €
2. Growing wheat: when you choose this action, you spend 20 €

At the end of the year, you harvest the wheat and you sell your wool. Each sheep produces 1 wool unit that is sold for 10 €. Selling the harvested wheat instead gives you 50 €.

However, during the year, there is a probability α that your fields are devastated by a storm. In this case, your harvest will give you 0 €.

Moreover, if you have more than one sheep, there is a probability β that each pair of sheep generates a new sheep.

Your manager career ends if you run out of money or, in any case, after 30 years, when you will retire.

You want to have a long and prosperous career.

Ideas for extension: this can be adapted to any managerial task involving decisions about investments with different returns. Actions can be continuous in the quantity of the investment or modeled as a distribution of your budget.

Goal

1. Find a way to leave your heirs as much expected legacy as possible, which means that you want to learn the best investment strategy in order to maximize your total monetary reward at the time of your retire
2. Study how α and β influence the situation you have to deal with

Project proposal 3: Grid driving

We want to learn how to drive a car on a racetrack that contains at least one turn. The racetrack is modeled as a collection of discrete positions organized in a grid. There is a starting line that is also an ending line (an horizontal row of grid cells), in such a way that the racetrack is has a circular shape.

The car movement is determined by its velocity which is made of an horizontal and a vertical component. The maximum value of each component is 5 grid cells and the minimum value is 0. At the starting line the velocity is 0 for both the components.

The actions that the car can take are to change the two velocity components, each by +1, -1, or 0.

Each movement costs -1. The race ends when the finish line is reached.

However, if the car runs out of the racetrack, the race is not over, but the car must restart from the starting line with velocity 0.

(**optional**) Finally, on the track there are some randomly arranged obstacles that occupy some cells of the grid. Passing by these obstacles involves the risk of getting a flat tire and consequently reducing the speed to 0.

Ideas for extension: this can be adapted to any continuous self-driving problem or to a still discrete, but more complex setting. For this last option, you can get inspired by the Heat game rules.

Goal

1. Implement the grid driving environment by modeling states, actions and the transitions from one state to the other.
2. Implement an agent that learns by practice how to reach the finish line as fast as possible.

Project proposal 4: Healthcare

Here, you represent the government of a country and your responsibility is to take care of the healthcare policy, but you have a limited budget. In particular, every year you have three available actions:

- INVEST in healthcare
  - the *level of the healthcare* is incremented by 3 but the budget decreases by 2
- INVEST in education and prevention
  - the budged is reduced by 1, but the current *health risk index* is also reduced by 1 (however, it never becomes lower that 0)
- DO NOTHING and save your budget
  - the budget increases by 2 and the healthcare level does not change

After each year, you get the current budget as a reward. However, after each year the health risk level rises randomly by 1, 2, or 3 points. When this happens, if the health risk level becomes higher that the healthcare level, there is a risk of a pandemic occurring in the country. More specifically, given a the pair of health risk and healthcare levels, only if *health risk* $>$ *healthcare*, there is a non-zero probability that a pandemic occurs. In case of a pandemic, you are fired from your position and the simulation ends with a large final negative reward.

Try different initial levels for the budget, the healthcare level, and the health risk index.

Ideas for extension: this can be adapted to any political simulation environment, where political decisions have a non deterministic effect on a population or a system and where your reward is based on the consensus and the happiness of the population. You can get ispired by the Democracy4 videogame (or similar). Also electoral campaigns are a good starting point.

Goal

1. Explore how the optimal policy changes under different scenarios (in particular with respect to the probability of health risk increment and different discount factors)
2. Try to model a situation where the budget is not the priority. For example:
    1. The priority is to minimize the health risk with a limited budget
    2. When you are a politician, you never know when your job ends. Thus, we want to keep people safe now, not in a far future
    3. Every 5 years that are the elections. Your probability to stay in your place is proportional to the difference between health index and quality of the health care. Your priority is to keep your job of course...

Project proposal 5: AZUL

> Created by Elisabetta Rocchetti, Department of Computer Science, University of Milan

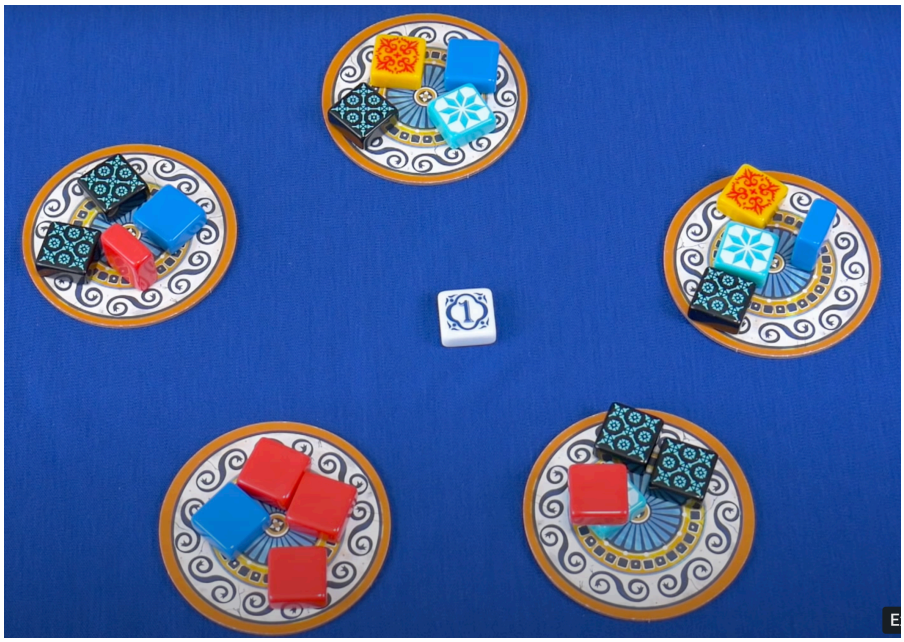
In this project, we invite you to dive into the colorful world of Azul, a popular board game that has players strategically drafting tiles to create beautiful mosaics. Unlike traditional board games that rely heavily on luck, Azul requires a blend of strategy, foresight, and adaptability, making it an excellent candidate for exploring reinforcement learning (RL) techniques.

Here is a Youtube video explaining the game: https://www.youtube.com/watch?v=y0sUnocTRrY

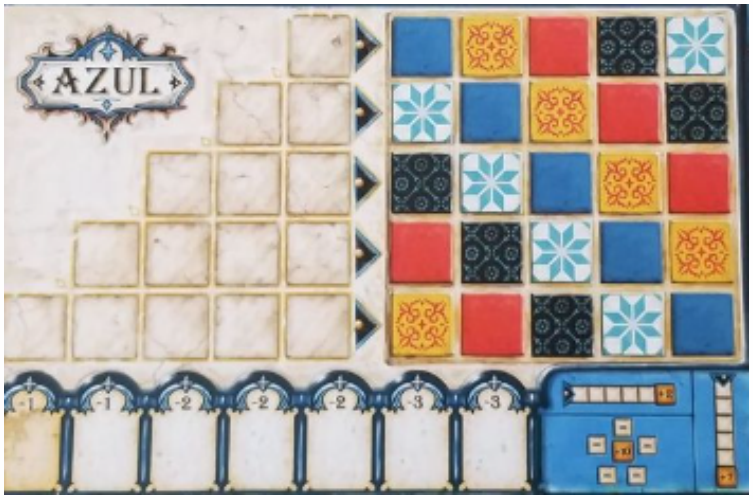
Azul Game Structure and Components

Azul's gameplay centers around drafting tiles and placing them to score points based on their arrangement on the player's board. Key components and structures include:

- **Tiles**: There are 100 tiles, with 20 tiles in each of the five colors (blue, yellow, red, black, and white).
- **Factory Displays**: For a two-player game, there are 5 factory displays, each initially filled with 4 randomly drawn tiles.



- **Player Boards**: Each board is divided into two sections:
  - **Pattern Lines**: To the left, with 5 horizontal rows increasing from 1 to 5 slots. Tiles here must be of the same color, and no two rows can share a color in their pattern lines.
  - **Wall**: On the right, a 5x5 grid associated with specific colors, arranged uniquely across rows. Completion of these rows and columns earns points.



Gameplay Overview

- **Drawing Tiles**: At the beginning of each round, each factory display is filled with 4 tiles (for the 2 player variant) drawn randomly from a bag containing all the tiles.
- **Drafting Tiles**: On their turn, a player chooses all tiles of the same color from a factory or the center of the table and places them in one of their pattern lines. Any excess tiles not fitting in the pattern line are placed in the floor line, incurring a penalty. If the player is the first choosing to pick up tiles from the center, then it also receives the “first player” tile and he or she must place it in the first slots of the floor line. This will give -1 as penalty, but the player will be the first to choose tiles the next round.
  - Players takes turns during the draft until no tile is available

- **Scoring and Tile Placement**: After all tiles have been drafted for the round, players then move tiles from completed pattern lines to the corresponding spaces on their walls. Points are scored immediately for each tile placed on the wall, based on the tile's position and adjacent tiles (see Reward System in Azul section for more details).
- **Round Ends**: At the end of each round, all tiles (apart from those remaining on the player boards) are put in the drawing bag again.
- **End Game Conditions**: The game ends after a round where at least one player has completed a horizontal line of tiles on their wall. Final scores include points for completed rows, columns, and sets of all one color on the wall.

Reward System in Azul

- **Placement Points**: Points are scored when tiles are moved from the pattern lines to the wall. Points are awarded based on the tile's position, adjacent tiles, and completing rows or columns.
  - The player earns 1 point for placing the tile itself.
  - If the tile is placed in a row with existing tiles, the player earns 1 additional point for each tile in that row (including the newly placed tile).
  - If the tile is placed in a column with existing tiles, the player earns 1 additional point for each tile in that column (including the newly placed tile).
  - If the tile completes a row or column, but there are no adjacent tiles in the row or column, only the point for the tile itself is awarded.
- **Bonus Points**: Additional points are granted for completing a row, column, or for collecting all tiles of the same color on the wall.
  - Completing a horizontal row awards a bonus (2 points).
  - Completing a vertical column awards a larger bonus (7 points).
  - Completing all tiles of the same color on the wall awards a specific bonus (10 points).
- **Penalties**: Players receive negative points for tiles that cannot be placed on the pattern lines or wall, influencing the strategic choice of tiles.
  - For any tiles that cannot be placed on the wall and are dropped into the floor line, a penalty is applied. This penalty increases with the number of tiles in the floor line. The floor line is structured as follows: there are 7 available slots, each of which gives penalty points depending on its position:
    - Slots 1 and 2 give -1 as penalty each
    - Slots 3,4 and 5 give -2 as penalty each
    - Slots 6 and 7 give -3 as penalty each
  - Example: if there are 4 tiles in the floor row, then the total penalty is: -1 -1 -2 -2 = -6

Modeling Azul as a Reinforcement Learning Problem

Your primary challenge is to frame Azul as a Markov Decision Process (MDP) suitable for RL exploration. This involves defining the game's states, actions, and rewards in the context of RL, with a particular focus on the unique aspects of Azul's gameplay, such as the player boards, the diverse types of tiles, and the drafting mechanics. Then, train a RL agent to learn Azul!

Simplified Azul Gameplay for RL

- For a two-player setup, the second player will be simulated by an agent performing random actions, providing a baseline challenge for the RL agent.
- Alternatively, the game can be reinvented as a single-player challenge, focusing on score maximization over a fixed number of rounds.
- The player boards could be simplified, removing the n-last pattern lines and the n-last rows and n-last columns from the wall.
  - Example