IPCA

ESCOLA
SUPERIOR
DE TECNOLOGIA

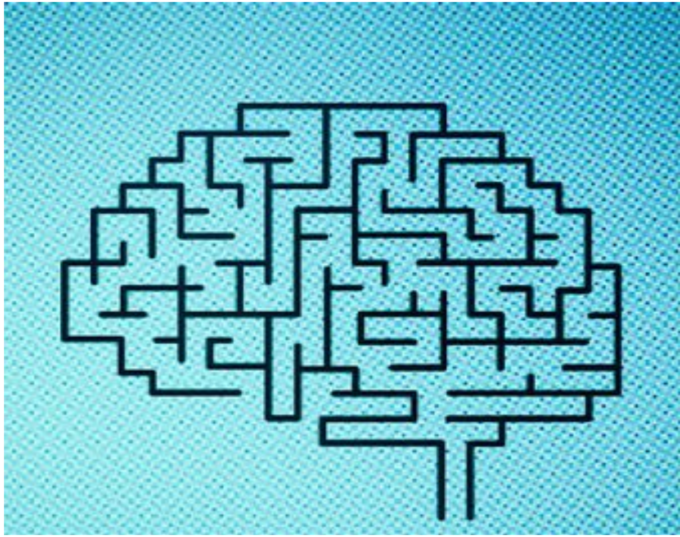
Project Guide 01

Artificial Intelligence (AI), 2025-26

Degree on Computer Systems Engineering

Introduction

- The goal of the project is to develop an intelligent agent, documenting the design and implementation steps
- Each work group needs to be made up of **5 students**
- The project must be available in a Github repository
    - The code and documentation must be merged into a Jupyter Notebook
    - The folder must contain all files required to run the solution
    - The folder must be shared with the lecturer
- A ZIP file of the repository must be submitted on Moodle before the deadline
- The presentation and defense of the project will take place on a date to be scheduled later.



Collaborative Platforms

The project should be hosted in a private **GitHub** repository

- Use the nomenclature "**IA25_P01_G##**", where G## corresponds to the group
- The lecturer must be added to the repository

The project must be submitted as a ZIP file labelled **IA_P01_G##.zip**, where G## corresponds to the group

The file must contain

- Jupyter notebook;
- readme.txt file with the link to the repository (Github);
- other files needed to run the notebook.

Theme — Class Timetable

CSP Problems can have an huge search space

A problem can have many of solutions of different categories:

- A **possible solution** is any solution, a complete assignment, whether or not it breaks any number of constraints.
- A **feasible solution** is a solution that does not break any (negative) hard constraints.
- An **optimal solution** is a solution with the highest score.
- The **best solution found** is the solution with the highest score found by an implementation in a given amount of time.

A problem can include different types of constraints

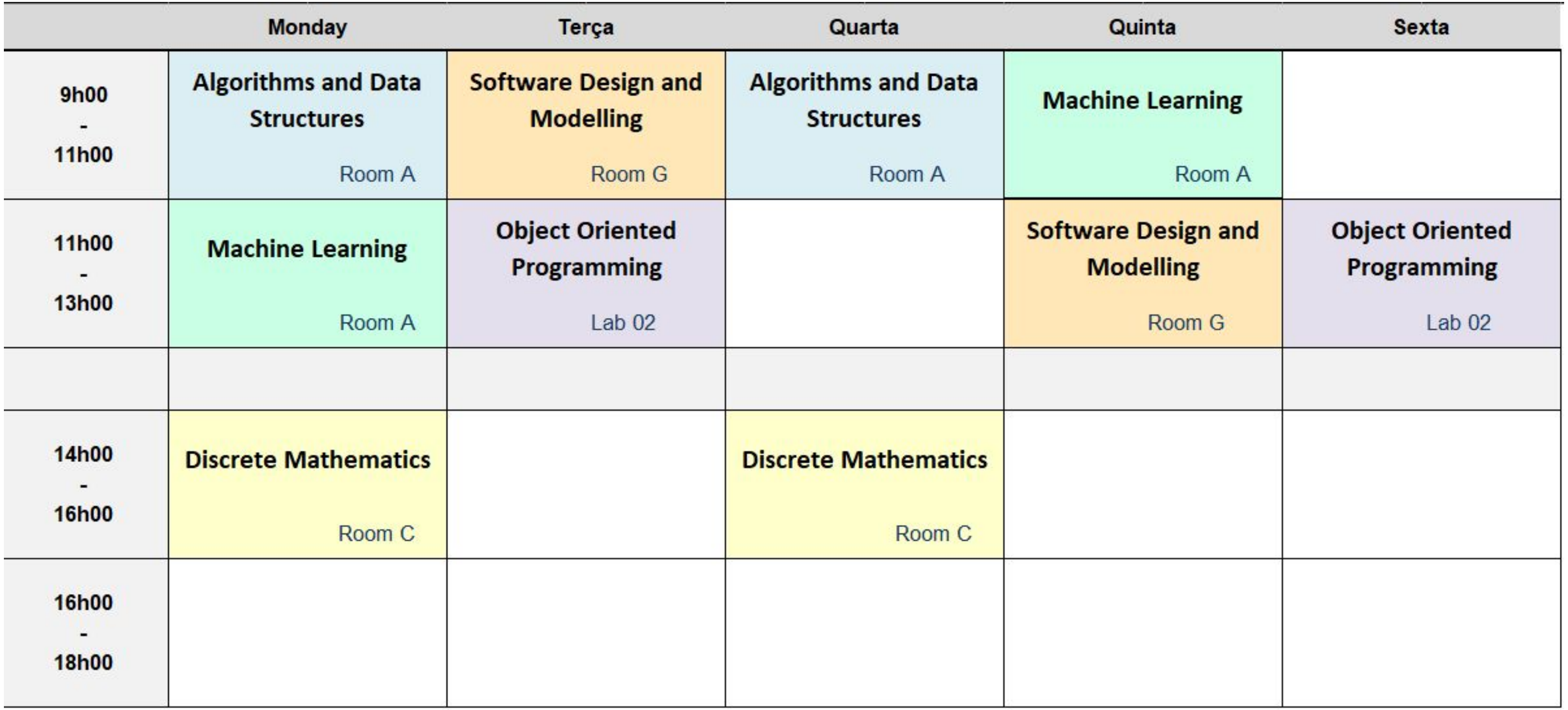
- **Hard constraints** — mandatory to get a valid solution.
- **Soft constraints** — preferred but not strictly required. so can be violated under certain circumstances.

Classes Timetable

- Each new academic year, the IPCA administrative team faces difficulties in creating class schedules.
- It is necessary to meet all restrictions related to teachers, courses, classrooms, etc.
- The project aims to provide a tool for generating class schedules for undergraduate courses at the higher education institution.

Source: https://meet.nyu.edu/academics/creating-your-ideal-course-schedule-as-an-nyu-student/

Timetable example



	Monday	Terça	Quarta	Quinta	Sexta
9h00 - 11h00	Algorithms and Data Structures Room A	Software Design and Modelling Room G	Algorithms and Data Structures Room A	Machine Learning Room A	
11h00 - 13h00	Machine Learning Room A	Object Oriented Programming Lab 02		Software Design and Modelling Room G	Object Oriented Programming Lab 02
14h00 - 16h00	Discrete Mathematics Room C		Discrete Mathematics Room C		
16h00 - 18h00					

Hard Constraints

- Classes last 2 hours.
- All classes have 10 lessons per week.
- Each course may have one or two lessons per week.
- A class cannot have more than 3 lessons per day.
- The timetable for a course may be subject to teacher availability.

source: https://nimbushosting.co.uk/blog/balancing-multiple-projects-essential-strategies-for-agencies

Soft Constraints

- The lessons of the same course must be in distinct days.
- Each class should have, if possible, only four days of lessons.
- Each day, lessons should be consecutive.

source: https://www.tech4law.co.za/business/practice-management-d58/be-flexible-man/

Challenges

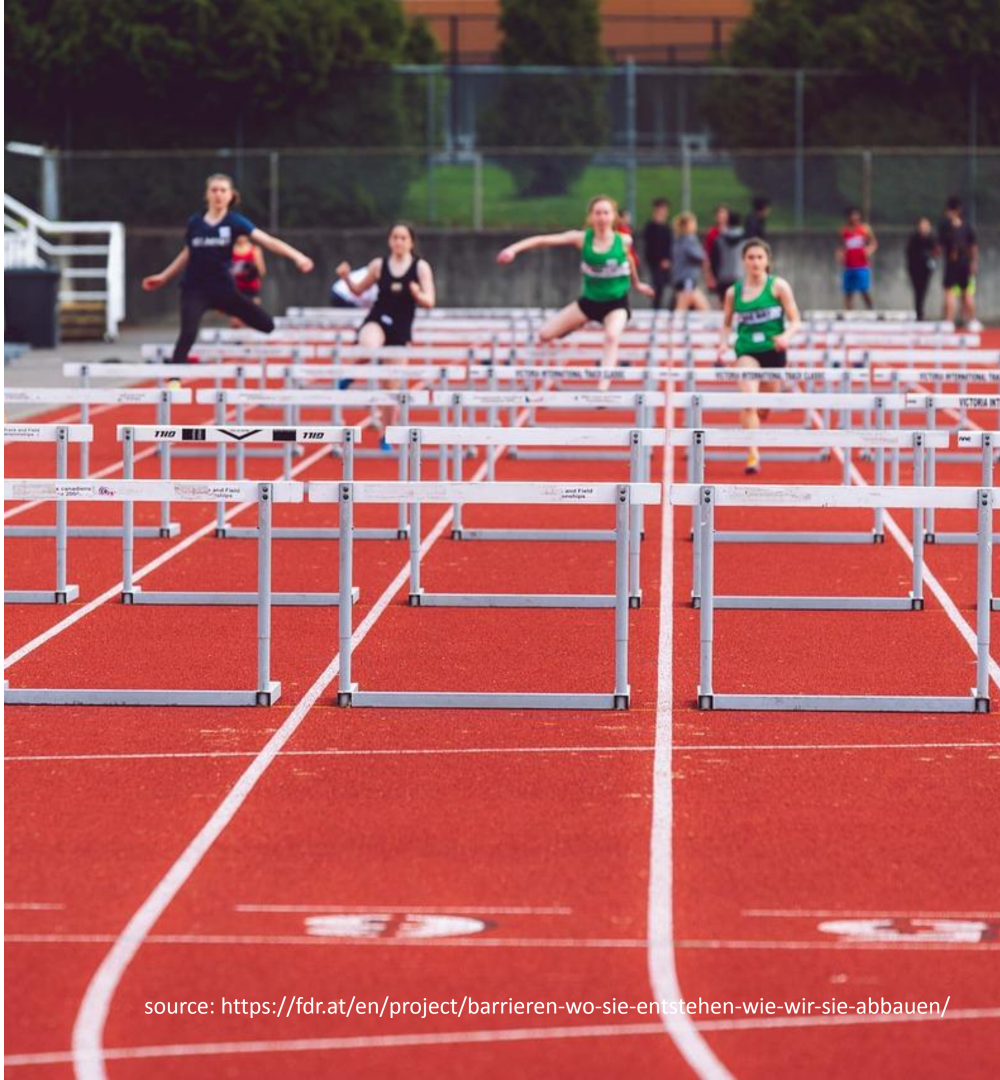
After implemented previous hard and soft constraints, you are invited to try to add some more.

Hard constraints

- When online classes are scheduled, limited to a maximum of three, they must be scheduled on the same day.
- Somes classes are required to be assigned to a specific classroom.

Soft constraints

- The number of classrooms used by each class should be minimised.



source: https://fdr.at/en/project/barrieren-wo-sie-entstehen-wie-wir-sie-abbauen/

Jupyter Notebook

Tools

- The project must be implemented in Python using a **Jupyter Notebook** and the **Constraint library**. Please check these links:
  - Constraint library site — https://pypi.org/project/python-constraint/
  - Constraint library documentation — http://labix.org/doc/constraint/

- Two datasets will be available in Moodle
  - a small, simpler dataset
  - a larger, more complex one

- The notebook must be well structured
  - Each step in a distinct block
  - Text blocks documenting the reasoning and decisions
  - Comments in the code blocks

Notebook structure

- Introduction
  - Establish here the context and the purpose of project
  - Identify the teammates: student name and number

- Agent design
  - Formulate the problem as a Constraint Satisfaction Problem (CSP)
  - Present the variables, their domain and the constraints definition
  - Highlight the heuristics applied to implement the soft constraints

Notebook structure (2)

- Agent running
  - Find how many solutions exist for the problem
  - Provide the best solution you can find
    - you can use an evaluation function to select the best solution from all available
  - Perform a critical analysis of the results
    - Provide execution times for the solvers used
    - identify some future improvements to the agent

- Conclusion
  - Draw up a conclusion about the outcomes, the development process and the tools used.
  - The structure of the notebook can be adapted according to each project characteristics.

Thank you!