L01: Introduction to Planning Algorithms

Planning Algorithms in AI and Robotics

Prof. Gonzalo Ferrer

Skoltech, 27 October 2025

Presentation: Who are we?

Instructor: Prof. Ferrer (name Gonzalo)
(g.ferrer@skoltech.ru)

Teaching Assistants: Sergei Bakulin
Ruslan Babakyan

Telegram chat: https://t.me/+TCQKjgy_xMNiN2Ey

Mobile Robotics Lab: Path planning, Robot Navigation in dynamic environments, Pedestrian Motion prediction, Sensor fusion of Lidar, camera, IMU, etc., SLAM, Localization, Mapping, etc

https://sites.skoltech.ru/mobilerobotics/

What is planning? Robotics

Robot converts high-level specification of tasks into low-level descriptions of how to move.

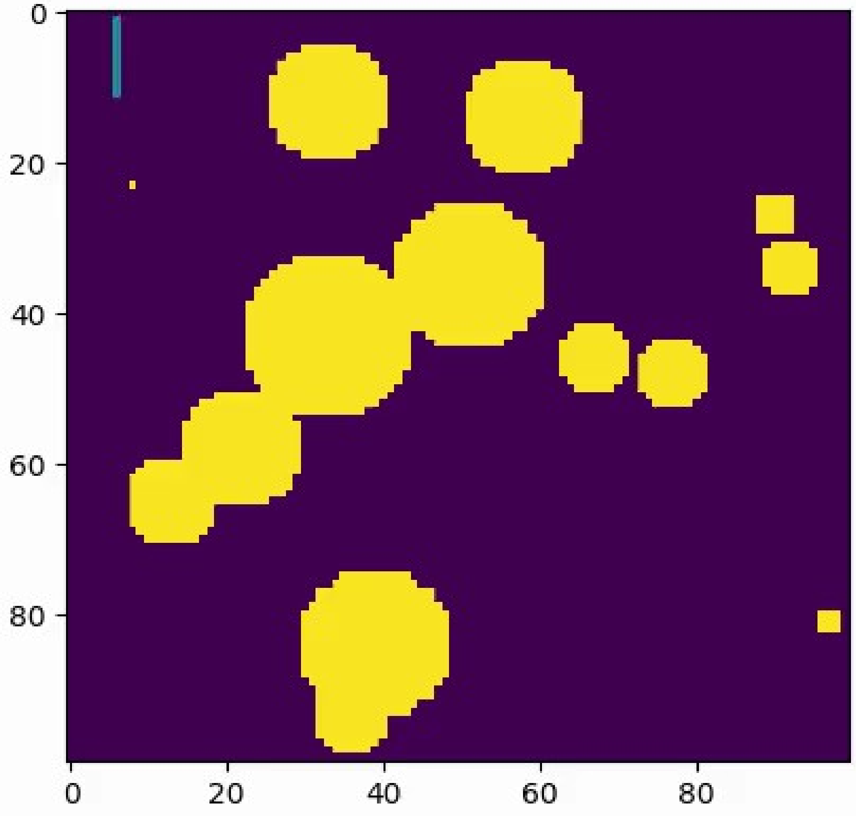
Mostly finding a plan is known as **motion planning** or **path planning**.

Example: The *piano movers problem*: how to move one piano from one room to the next room.

What is planning? Robotics

Other examples, moving an object from a **starting configuration** to g**oal configuration** while avoiding obstacles.

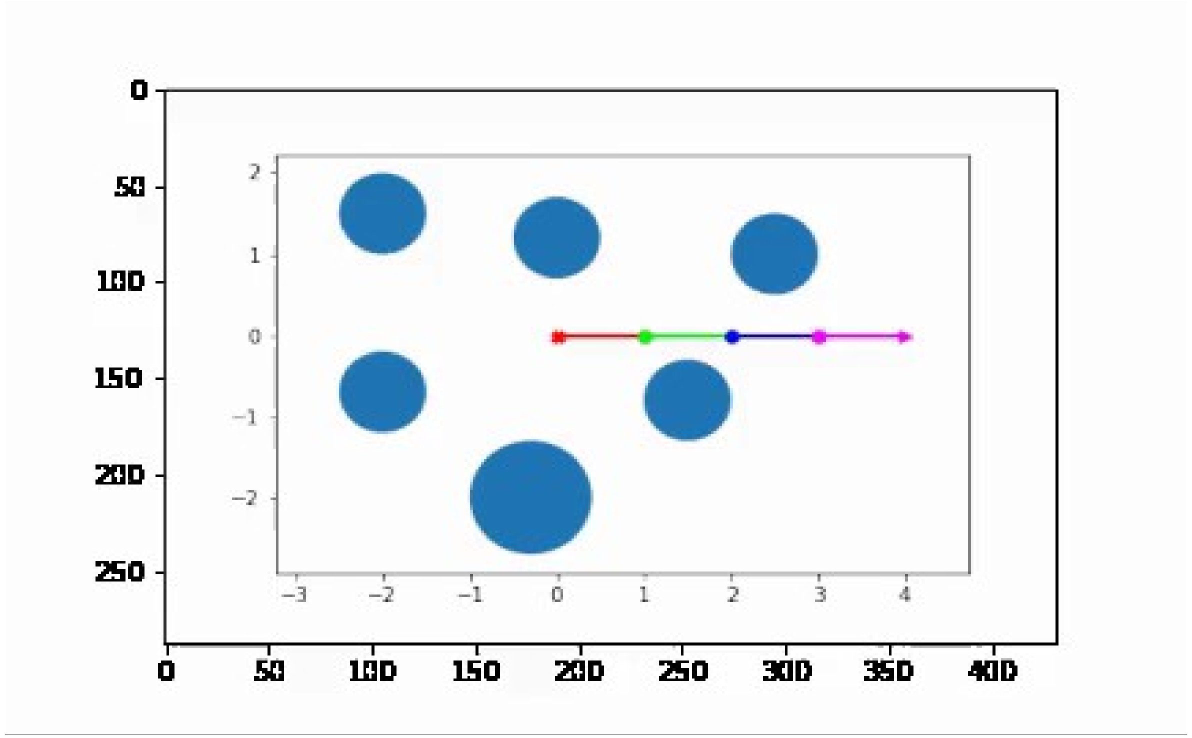
On this first view we only consider **feasibility**, although later we will consider optimality and uncertainty.



What is planning? Robotics

This is part of the PS2, where you will implement a sampling based algorithm (RRT) to solve the 5R manipulator.

We will investigate how sampling-based planning can help to alleviate the curse of dimensionality.

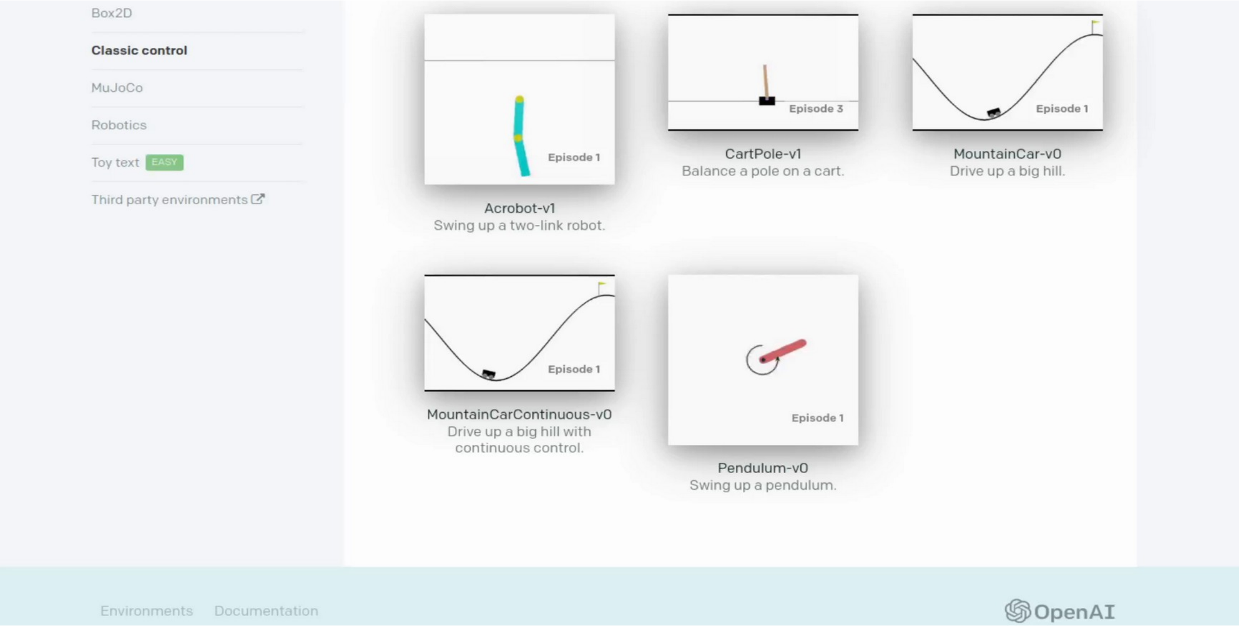


What is planning? Control Theory

Control Theory typically considers physical systems and differential constrains.

The **controller** designs **feedback policies** or **control laws**.

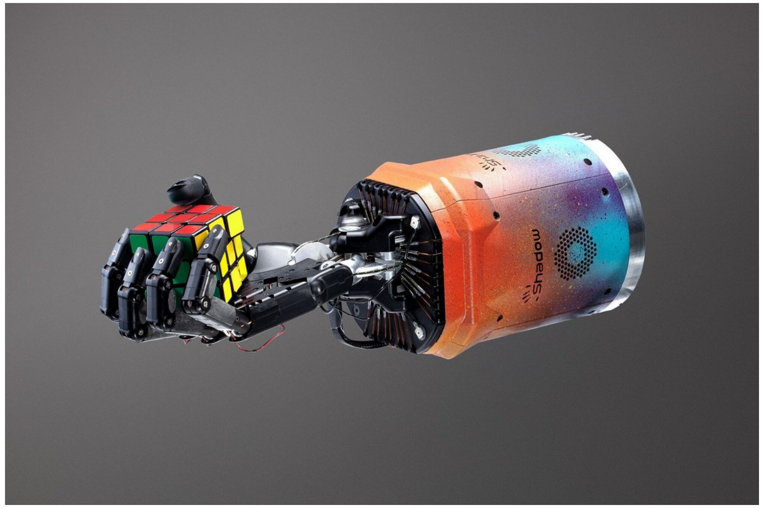
Controls usually focuses on <u>optimality</u> and stability.



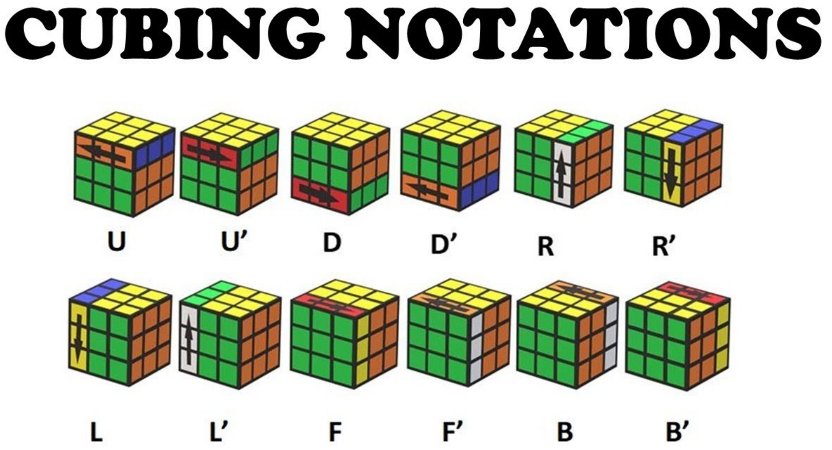
What is planning? Artificial Intelligence

Agents or decision-makers in AI do planning or **Problem Solving.**

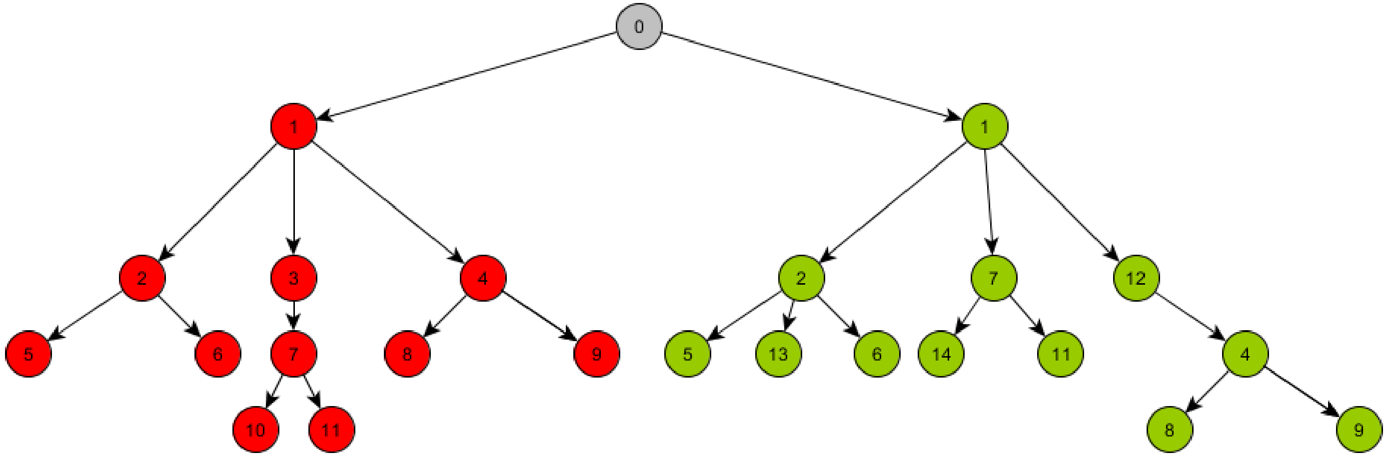
Usually discrete spaces, leading to combinatorial solutions.



Rubik cube (with robot)



Rubik cube's actions

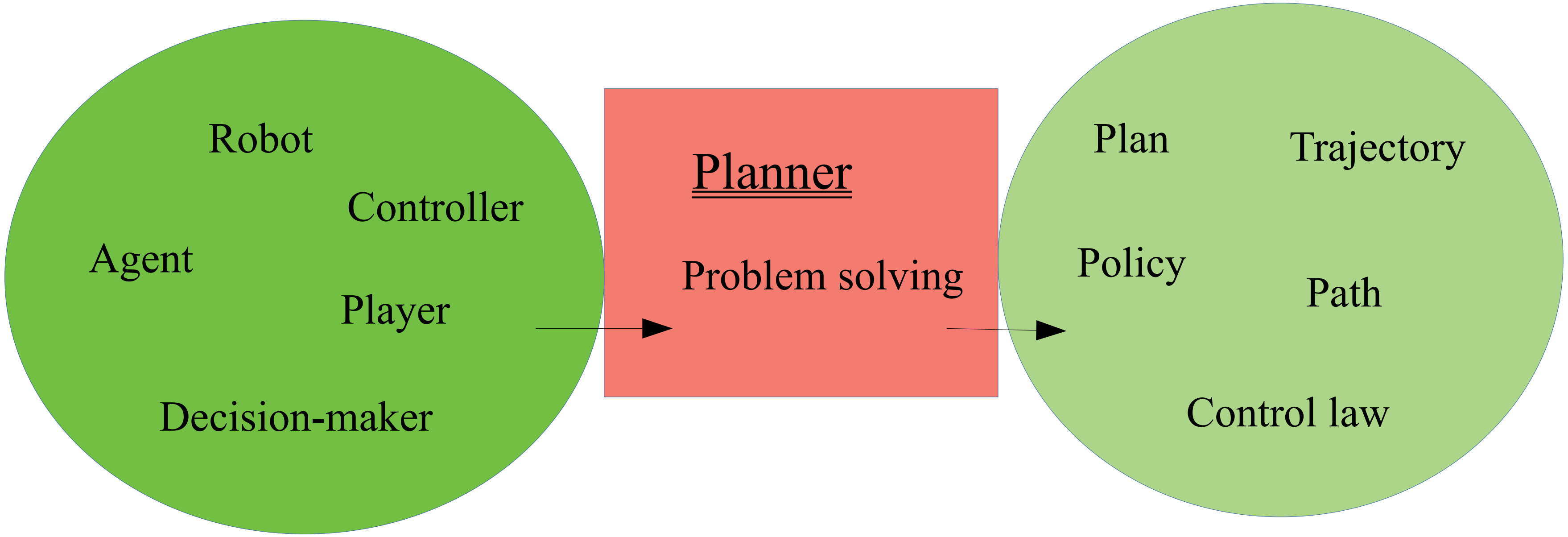


Example of Graph of possible action-state

Planning to plan

Planning means different things in different disciplines: robotics, artificial intelligence and control theory.

In this course we will present a unified view:



This view is from LaValle's book, Ch.1

The ingredients of planning

- **State**: All possible situations that could arise.
- **Action**: Change the states. Also known as *inputs, controls, operations*, etc.
- **Initial and Goal states**: A planning problem involves these 2 states.
- **Criterion**: The desired outcome of a plan:
  - Feasibility: arrival at a goal state, regardless of efficiency.
  - Optimality: Find a feasible plan and optimize some function.
- **Plan**: A specific strategy of behavior. It could be a sequence, a policy, etc.

The ingredients of planning

Example: the Piano movers problem.

State: 3D poses | Actions: 3D (relative) poses

Initial state: the corridor | Goal: my office

Planner: RRT (L04 algorithm) | Plan: sequence of actions

Example: How to arrange the furniture in your new house

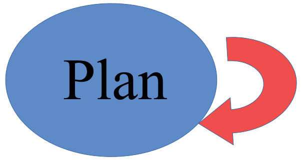
State: Position of all objects | Actions: Move objects

Initial state: Empty space | Goal: Ikea picture

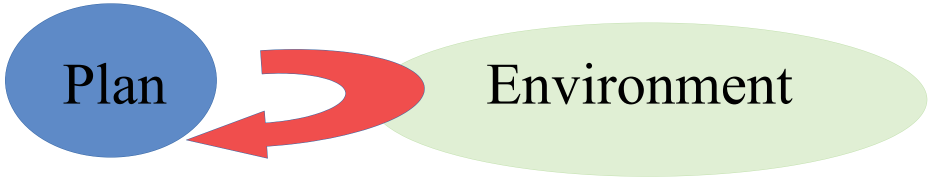
Planner: Yourself | Plan: sequence of actions.

Taxonomy of Planning

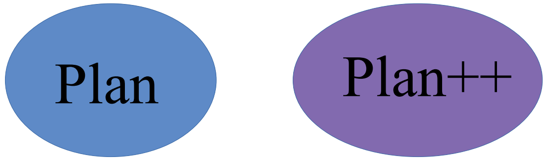
There are different ways to use the plan calculated by the planner



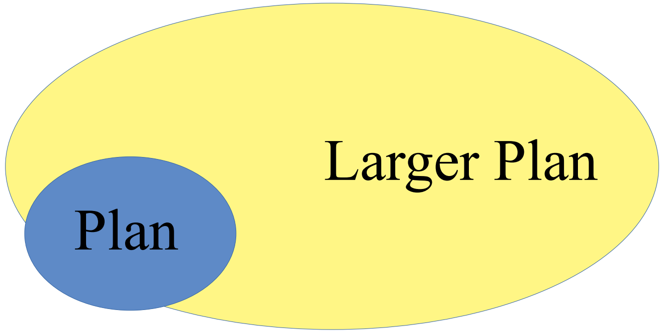
Open loop execution



Feedback



Refinement



Hierarchical

Course Goals

Mastering a set of core planning algorithms, covering a wide variety of planning problems, such as <u>discrete planning</u>, <u>continuous planning</u>, <u>decision-making</u>, planning under <u>uncertainty</u>, <u>learning-based</u>, etc.

For any give task, select the right planning algorithm.

Prerequisites

Basic programming skills in Python

Probability, introductory course.

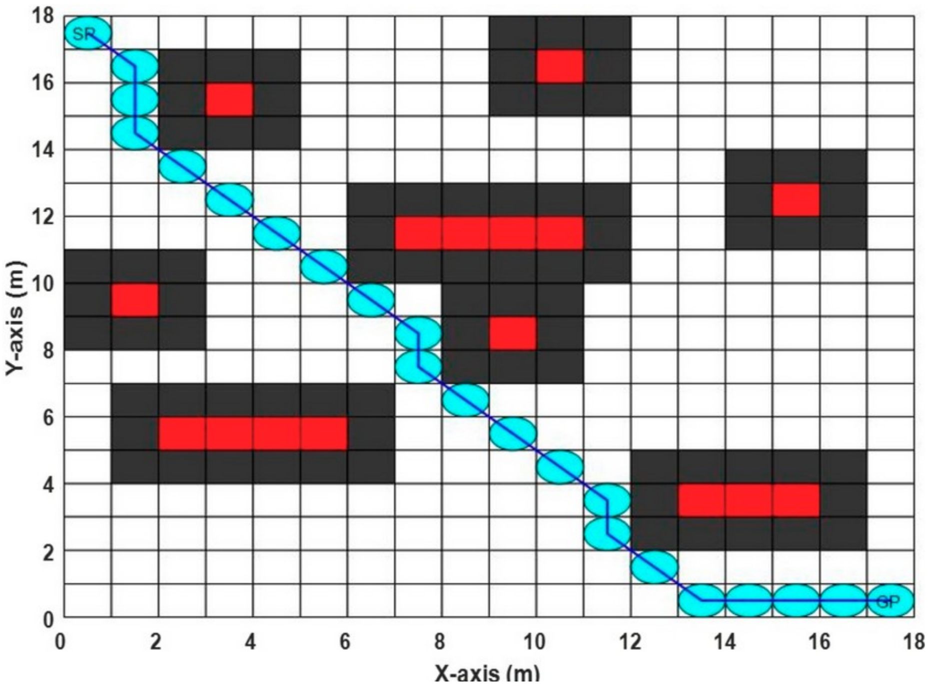
Next steps -> Perception in robotics in T3

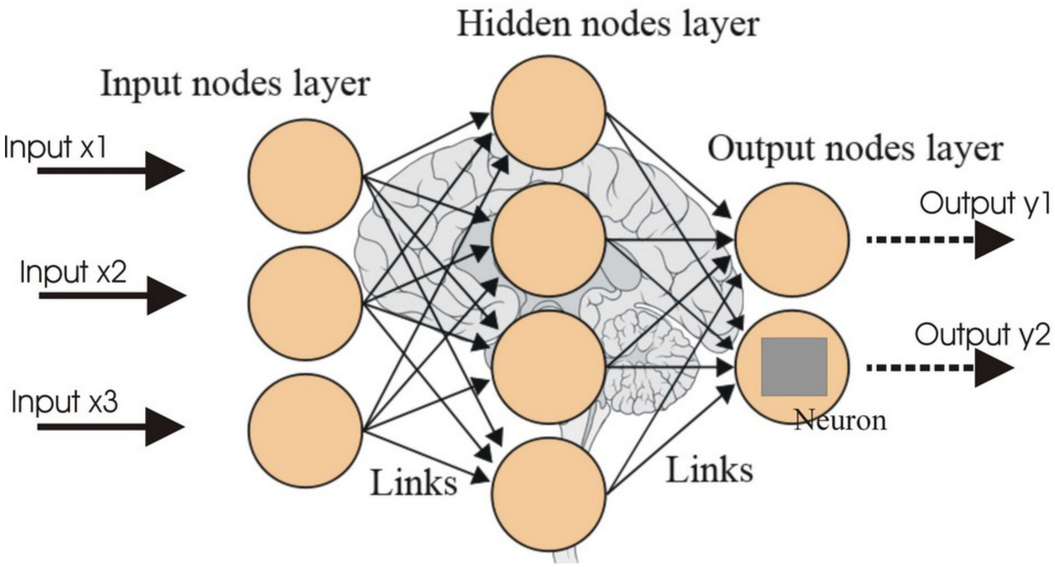
https://github.com/g-ferrer/Perception-in-Robotics-2025

Why study planning in the era of AI?

It is a valid question, are techniques such as A* and RRT obsolete?

I just train a policy and it can solve many planning problems!? Right?





Why study planning in the era of AI? Answer

Answer: No. Planning is more relevant than ever, not as replacements, but in different schemes.

1.) Guarantees & trust:

Classical Planners
✅ **Optimality:** Guarantees the shortest path (e.g., A∗).
✅ **Completeness:** Will find a solution if one exists.
✅ **Predictable & Verifiable:** Essential for **safety-critical** systems (autonomous vehicles, surgery robots).

AI/Deep Learning Models
❌ **Probabilistic:** Finds a "good" path, not necessarily the best.
❌ **Data-Dependent:** May fail on unseen scenarios (edge cases).
❌ **"Black Box":** Difficult to debug and certify for high-stakes applications.

Other reasons to consider planning

2.) Interpretability & Debugging
- We need sometimes to know why!

3.) Data & Computational Efficiency
- Robotic data is very expensive to obtain, sometime it is not possible to train properly with small amounts of data
- Some neural methods include current VLM foundation models and add on top of it action (plan) to the network.

Why study planning in the era of AI? Powerful hybrid examples

AI-Enhanced Heuristics: Use a neural network to predict a better heuristic for A*, making it much faster while keeping its optimality guarantee.

Learning Cost Maps: An AI learns from human data to create a cost map for "socially aware navigation" or "rough terrain." The Classical Planner (A*) uses this intelligent cost map to find the best path.

Hierarchical Planning: AI/ML provides scene understanding a high level goals while the classical clanner provides optimal path (MPC for instance).

Learning Outcomes

- The student will acquire **theoretical knowledge** and a rich set of **practical skills** to design their own planning and decision-making algorithms, applied to any kind of problem related to AI, Robotics, etc.
- The student will be able to **analyze** problems under the perspective of planning, and provide a more diverse set of tools for problem solving.
- The student will get **experience** on different planning techniques such as discrete planning, continuous planning, decision-making, planning under uncertainty, learning-based. We will prepare materials, seminars and problem sets that will serve as a first step into each of these particular topics.

Course Structure

Lectures (on campus B2-3007)	Monday 16:00-19:00 Friday 16:00-19:00
75% Problem Sets	PS1: Discrete planning (13-Nov-2025) PS2: Sampling-based planning (27-Nov-2025) PS3: Decision Making (7-Dec-2025)
25% Final Group Project	(more in canvas and later in class)

Grading system

A: 90

E.g. 86 means that students performing at 86%-100% would be graded with A.

B: 80

E.g. B set to 76 means that students performing at 76%-85% would be graded with B.

C: 70

D: 60

E: 50

F: 0

E.g. F set to 0 means that students performing at 0%-45% would be graded with F.

Course Material

- Lecture slides, uploaded after every class.
- Videos from lectures (in principle).
- Books:
  - S. LaValle, “Planning Algorithms”. Cambridge university press, 2006
  - Sutton and Barto “Reinforcement Learning: an Introduction”, MIT press 2018
- Canvas, selected papers
- Telegram channel

Class Structure

Lectures will be imparted live on campus.

Students are encouraged to participate and discuss in class.

Videos from previous years could be used in case of missing the class, but the content is slightly outdated.

Recommendations:

Participate as much as possible in class!!

Make questions, be engaging, learn more.

Problem Sets

- Problem Sets (PS) will be written in Python
- PSs are substantial work and should be worked on during the full allotted time period (each is a 25% of your grade).
- There will be an upper bound in the max possible grade of -0.5%/hour up to 60% of the grade (after 4 days).

  Grade = min (your grade, max possible grade)
- Late submission is based on the last update to canvas.
- Students are encouraged to discuss on PS. Copying code is forbidden, if caught, the PS will be invalidated.
- All PS’s must be submitted in order to pass the course.

Course Policies

Attendance

Attendance is required. This will be your chance to ask, discuss and learn as we progress. Later it will be too late.

PS Regrade Policy

If you believe we graded a problem-set or an exam of yours incorrectly, you can submit a regrade request no later than one week after the graded work is originally returned.

Academic Integrity

Reference to Skoltech's policy (see canvas)

Final Group Project

- Topic (related to the course): Extend a state of the art algorithm, or paper reproduction or implementation on your own settings. We will upload to canvas a list of selected papers for inspiration.
- 3-5 students / group
- Presentation (December 19) 12’+3’ questions (TBD)
- Final project document, on an IEEE template.

Past projects

Agile Opponent Overtaking in Drone Racing

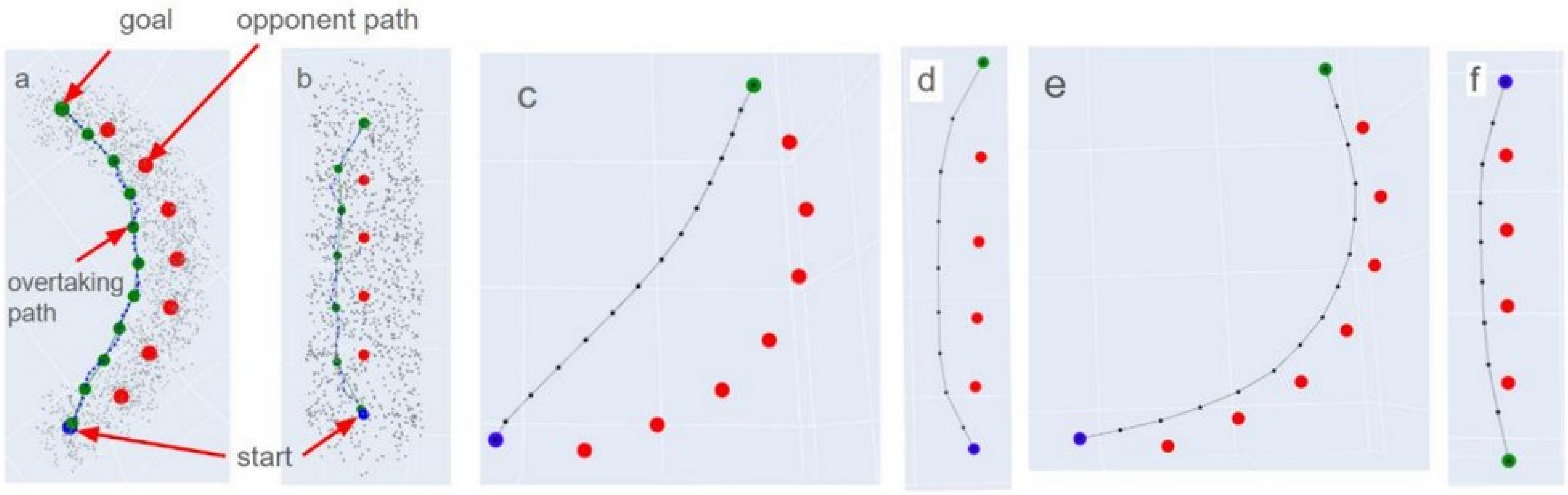


Figure 6. Generated path to overtake Leader drone path (red points). a – RRT curve, b – RRT line, c – A* curve, d – A* line, e – CHOMP curve, f – CHOMP line.

Past projects

Path planning: modern approaches

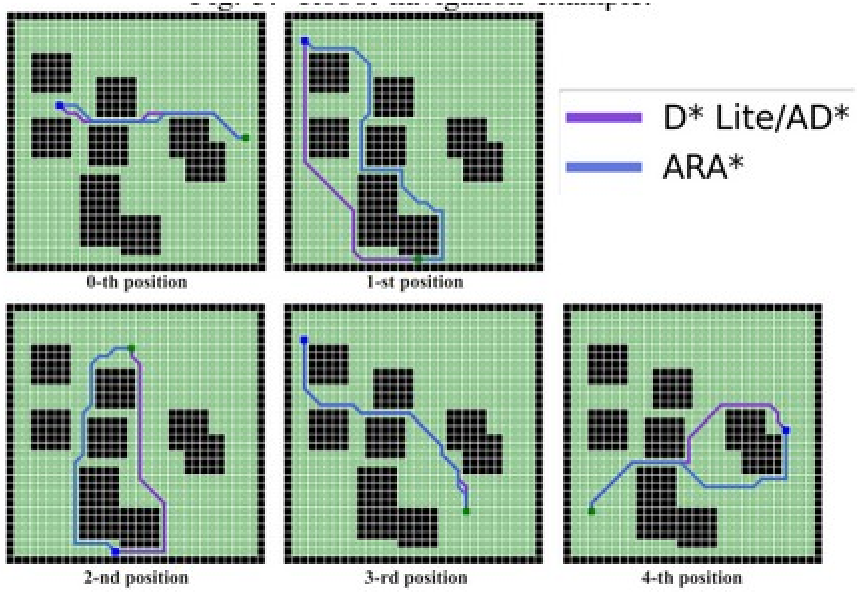


Figure 5 shows the final routes for the robot that the algorithms found. We can note that in our cases, algorithms D* Lite and AD* (Anytime Dynamic A*) always found the same routes.

Lazy collision checking in sampling based planning

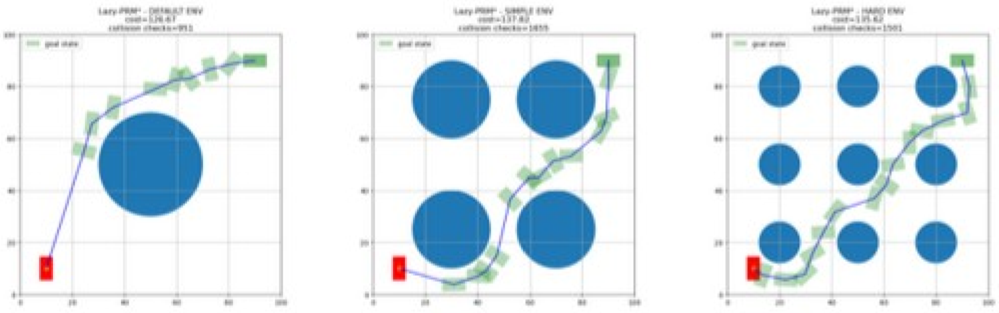


Fig. 3. Lazy-PRM* performance

Algorithm	Avg. collision checks	Avg. path costs
PRM*	32450	146.74
Lazy-PRM*	1457	134.88

TABLE I
COMPARISON OF PRM* AND LAZY-PRM*

Past projects

Evader-pursuer zero-sum game by MCTS

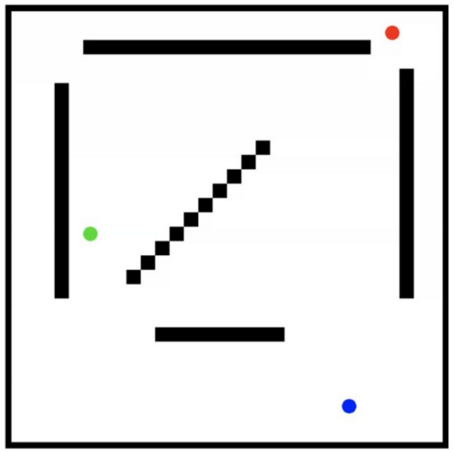


Fig. 2. Environment. Red circle corresponds to the pursuer position, blue to the evader position, green to the goal and black regions correspond to the walls.

Comparison of different algorithms for Pacman environment: MCTS, DQL, etc

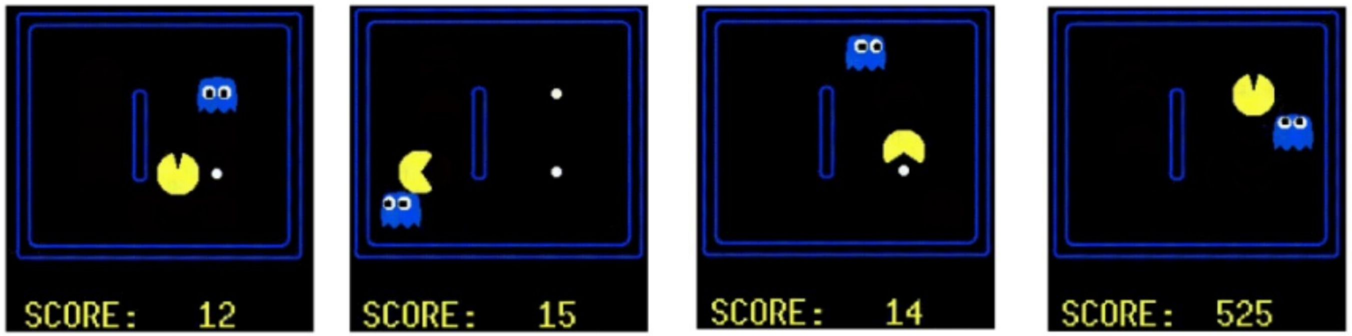


Fig. 6. Experiments with Q-learning on small environment