Skoltech
Skolkovo Institute of Science and Technology

Final Remarks on Planning: Learning and Beyond

Planning Algorithms in AI

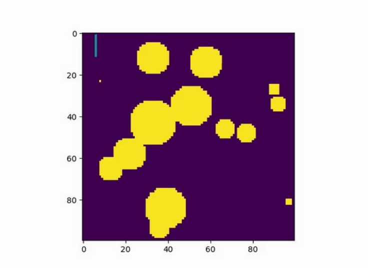
Gonzalo Ferrer

Skoltech, 09 December 2022

How to choose a Planning algorithm?

First an most important question is to think about the **state space** and **action space**.

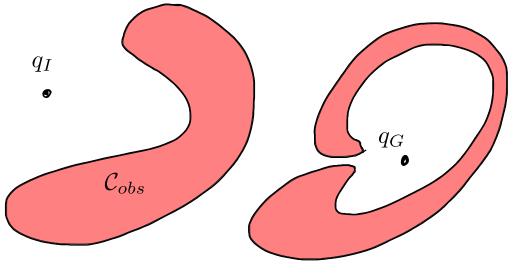
Can the state space be discretized?



Yes, and the dimensionality remains relatively low (3-4)

Yes, but dimensions are high

No, irregular regions, complicated spaces, etc.



Discrete planning (L02)

- Forwards search
- Dijkstra, A* if cost

Sampling (L04) (LaValle)

- Variants of RRT or PRM
- RRT* if we want optimality.

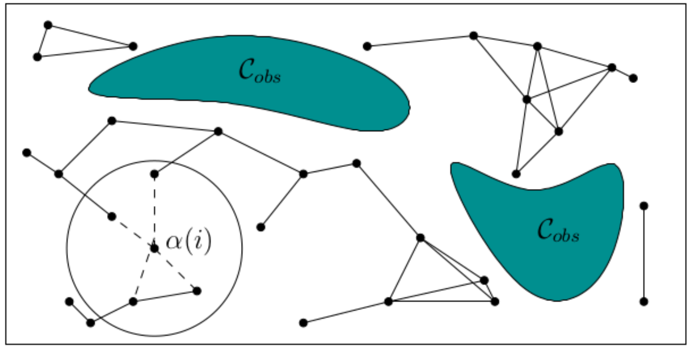
How to choose a Planning algorithm?

Do we want a single plan or a **multiple query** algorithm?

Yes, plan can be as complex as we want

Discrete planning (L02),
RRT (L04),
Traj. Optimization (L06)

No, we will execute the plan multiple times



PRM (L04)
Discrete Planning (L02) (it might be still more efficient)
Multiple-VI (L05) (only if memory is not a problem)

How to choose a Planning algorithm?

There will be perturbation while executing the plan? (**policy**)

Yes, and we also want the optimal plan

Value Iteration (L05)
Wavefront
Max clearance

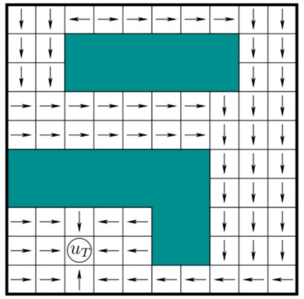
Continuous state values?

Interpolated VI (L06)

Discretization+NavFunction

Yes, only feasible plan

Navigation Functions (L05)



Potential Fields (L06)

Cell Decomp, Funnels

Not needed

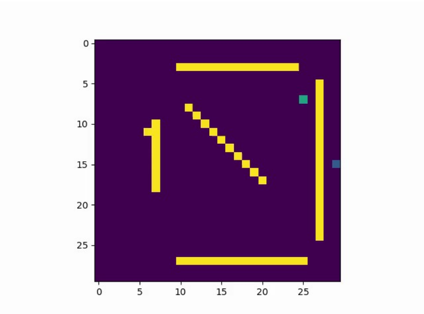
Methods from L1-4

How to choose a Planning algorithm?

Is this a game? → gigantic state space

State space is "small" enough for building a decision tree

State space is monstrous but the action space is relatively small



Alpha-Beta Pruning (Russel&Norvig)

It allows to almost double the depth wrt naive decision tree and the solutions are guaranteed to be optimal

Monte-Carlo Tree Search

The evaluation is approximated by sampling and this allows to explore depth of the tree unthinkable for exhaustive search

How to choose a Planning algorithm?

Is there **uncertainty** of any kind?

Yes

MDP (L08) (Sutton&Barto)

- It works well for finite state spaces
- Requires known transitions

Yes, but the environment might be unknown

Reinforcement Learning

- It only interacts with the environment and does not require **complete** knowledge.

No, states are observable and transitions are deterministic

Planning methods from L1-6

How to choose a Planning algorithm?

Are any parameters of the environment **unknown**?

Yes, such as transition function, action execution, etc.

Learning Based approaches:

- **Supervised learning**
- **Reinforcement Learning**

No, we have total control on all elements

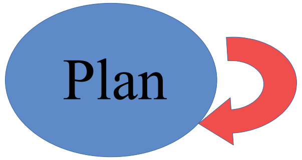
Planning L1-4, discrete, sampling, policies, etc.

MDP (L08) For discrete and low dimensional state spaces

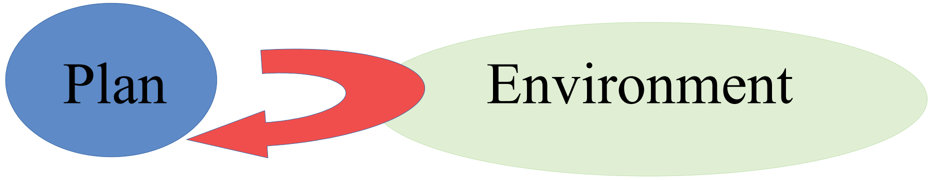
Dimensionality is high, we need function NN approximators

Planning Taxonomy

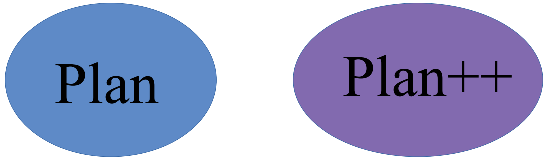
Now we see the power of combining plans and a hierarchical approach.



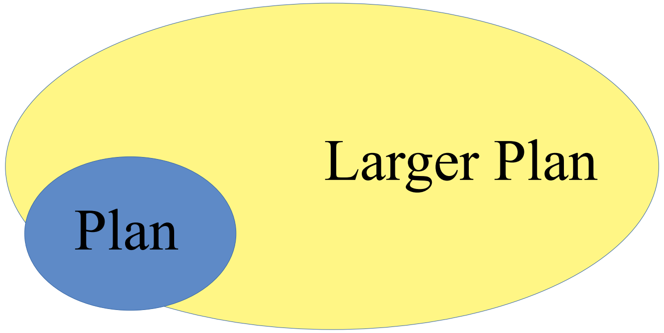
Open loop execution



Feedback



Refinement



Hierarchical

Learning in planning: Limits and Potentials

- Some problems do not require learning and planning methods can be enough, so it is better if we ask first: does it make sense for your problem?
- We must always think what is the learning algorithm doing:
  - Policy approximated by a parametrized function
  - Compressing a value function over a continuous state space.
  - Heuristics for a distance function
- Second we must think about the state space:
  - Position, Pose, Image, a sequence of values, etc.
- The equivalent in CV to classification and regression is not always applicable since the agent, as a result of a plan, executes an action in an environment, whose results are uncertain as well.

Learning in planning: Limits and Potentials

- On the other hand, learning-based approaches **allow** to solve planning under impossible conditions for planning, such as unknown dynamics, changes in parameters and many other artifacts.
- **Combining** both approaches can be a great way to go. Examples are MCTS with learned policies, which also combines approximation to evaluation functions.
  - In RRT, learning sampling distributions, A* learning heuristics, Potential functions learned, etc.

Sünderhauf et al. (2018). The limits and potentials of deep learning for robotics.
The International Journal of Robotics Research