UPF Report

Damien Allonsius & Anders Jonsson

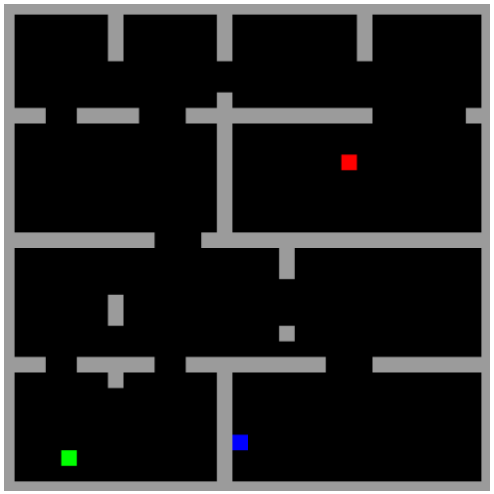
30 April 2019

Outline

Outline

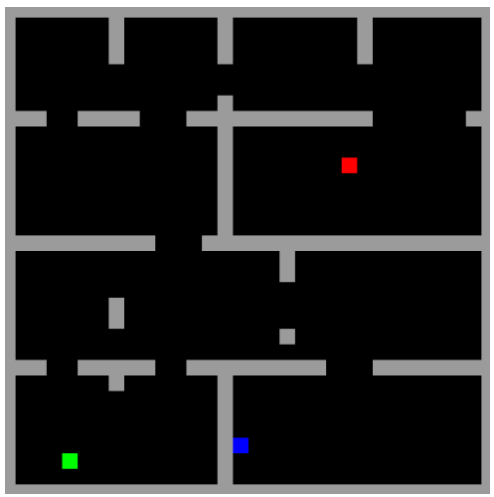
Reinforcement learning

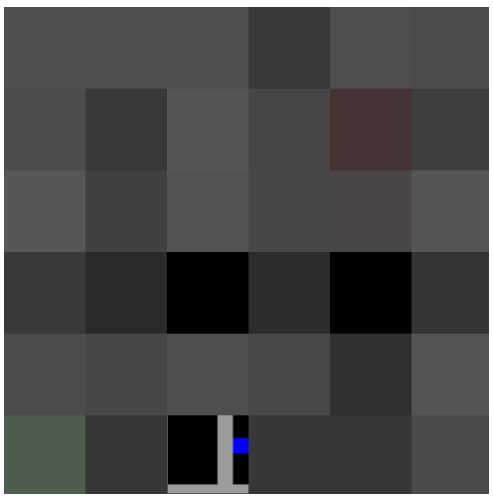
- An agent learns by experience to perform actions that achieve a certain task
- Already challenging for individual tasks
- Example : navigation task where the task is to pick up a key to open a door



Lifelong learning

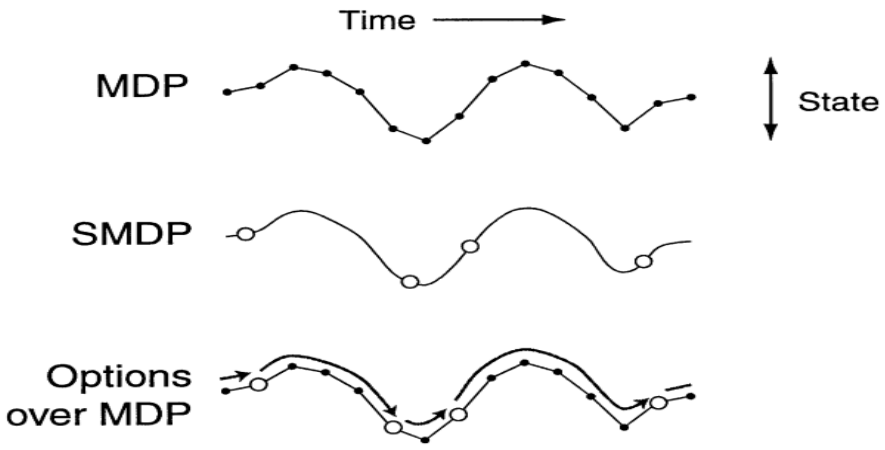
- No initial model of the environment
- Only available information is what agent observes right now
- Agent may face new tasks later
- The environment dynamics may change over time



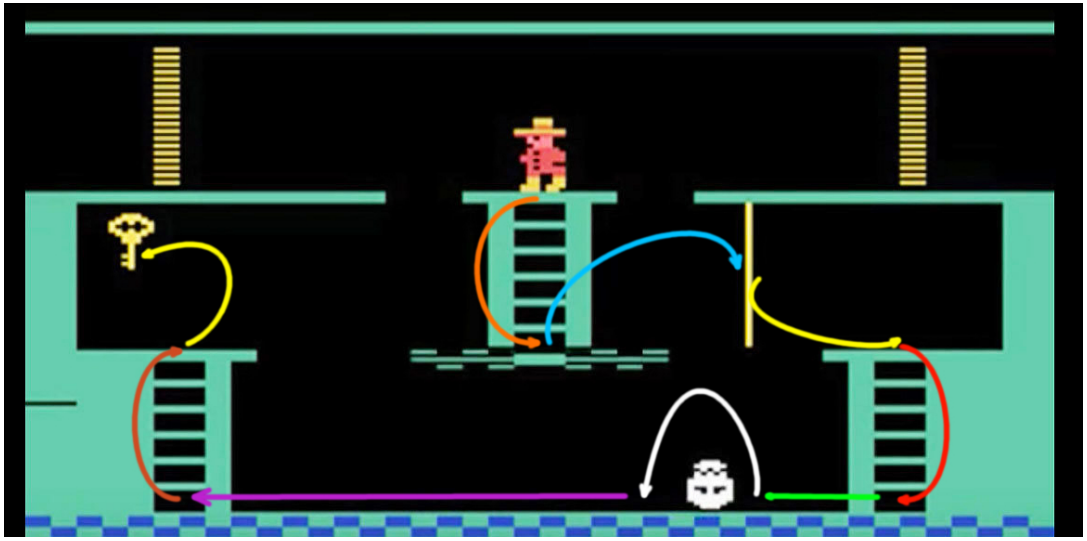


Hierarchical reinforcement learning

- Decompose task into subtasks, each with its own policy
- Actions are performed at different time scales



Example : Montezuma's Revenge



Why hierarchical RL ?

- Exploration and planning at different levels of abstraction
- Capture subtasks that are relevant for a range of tasks
- Planning horizon is shorter for subtasks (easier to solve)
- Drawback : more learning problems !

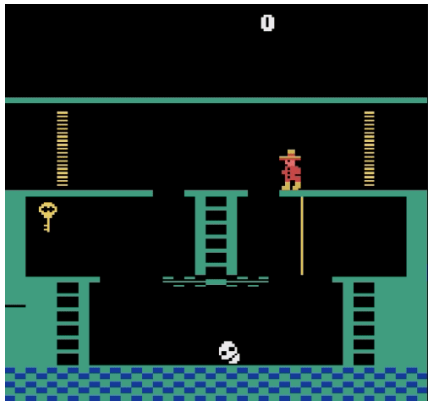
Outline

Setup

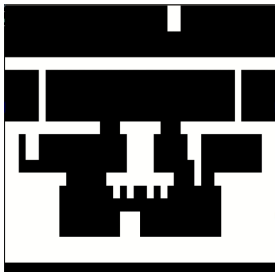
- Assume that the state space is partitioned into regions
  (Formally, need a function $\Phi : S \to \mathbb{N}$ mapping states to partition IDs)
- Initially the agent is in a single region and has no options
- Two phases :
  1. Exploration : find transitions among regions
  2. Learning : introduce options that perform transitions between regions
- Abstract MDP : states are regions, actions are options

Example : Montezuma's Revenge

Partition : downsampled images of game screen
Option policies retain more detailed states



Normal view



Options' view

Agent's View

Our Strategy

- At the agent level :
  1. Exploration strategy with tree search.
  2. When a reward is found : Q-Learning.

- At the option level, correct transitions can be learned with
  1. Q-Learning
  2. DQN (ongoing implementation).

- A reward (resp. penalty) is given when the option performs good (resp. bad) transitions decided by the agent.

Advantages

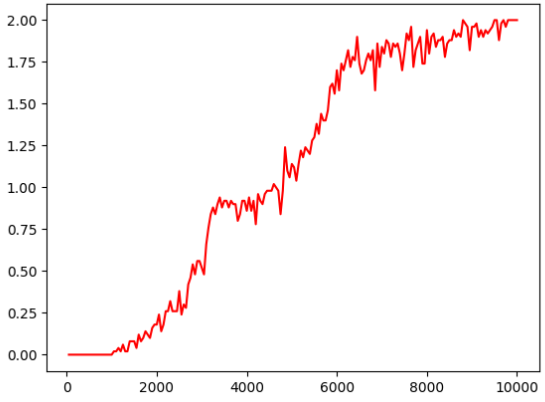
- Can be combined with planning since agent maintains a transition model
- Flexible, can be combined with any strategy for identifying subtasks
- Partitions can be formed even when there is no clear task structure
- Transition options can potentially be reused (transfer learning)
- Options are only applicable in a single region (action elimination)

Outline

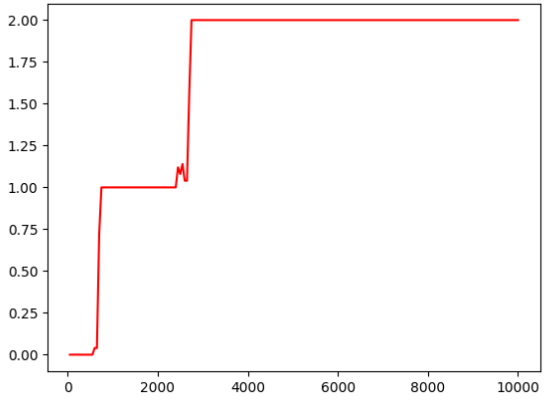
Preliminary results

Comparison between our strategy and Q-Learning.

GRIDWORLD ENVIRONMENT



Q Learning



Hierarchy

MONTEZUMA'S REVENGE (ATARI)

We restart every time the agent picks the key.
With our strategy we get the key 19 times in 3000 iterations.

Outline

Planned work

Apply the hierarchical approach to the microgrid simulator :

1. Downsample the state space by discretizing the state of charge of batteries (and the consumption and production of energy).
2. Make options to go from a level of charge to another.

Let's talk about this together !