Aalto University
School of Electrical Engineering

ELEC-E8125 Reinforcement Learning
Large POMDPs

Joni Pajarinen

23.11.2021

Today

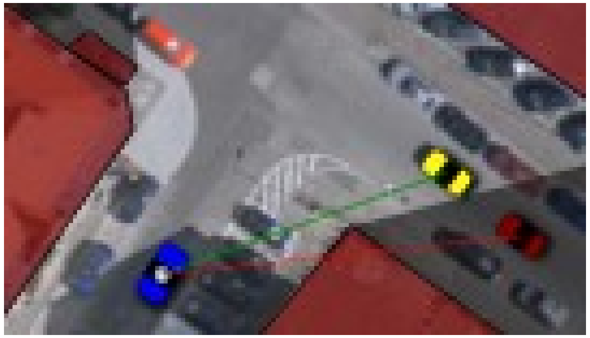
- POMDPs towards real world problems

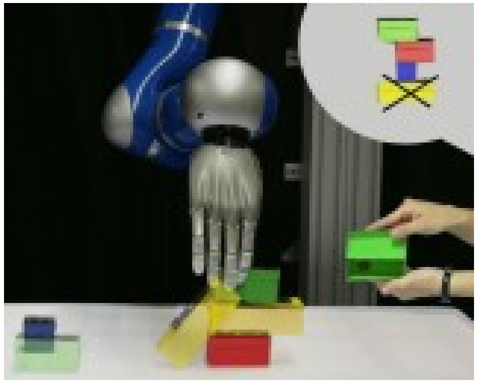
Learning goals

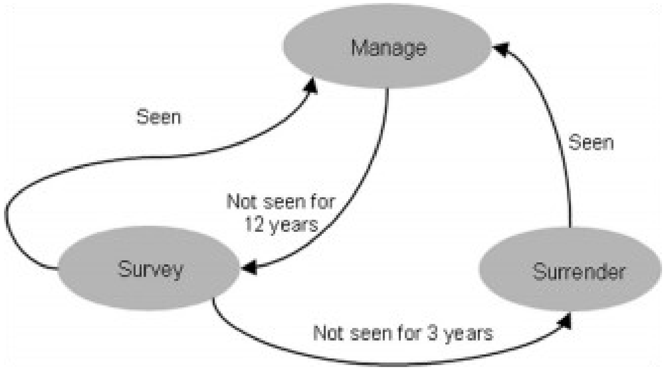
- Different ways for solving discrete valued POMDPs
    - Classic POMDP methods
    - Treating solution process as search
- POMDPs with very large observation and action spaces

Motivation: POMDP application examples

- Autonomous driving
- Human-robot interaction
- Tiger reservation
- Robotic manipulation
- Teaching systems
- Target tracking
- Localization and
  Navigation

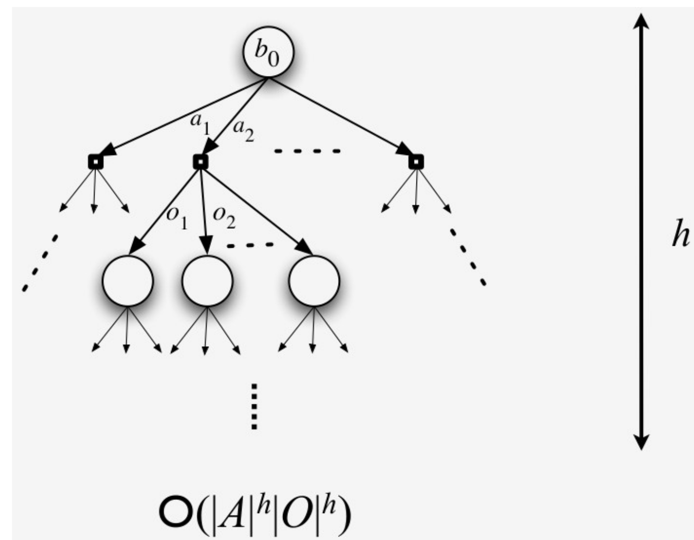






"Curses" of POMDP

- Curse of dimensionality
  - Number of states exponential in number of state variables
  - Complexity of accurate discretization exponential in belief dimensionality, that is, number of states
- Curse of history
  - Complexity exponential in length of history

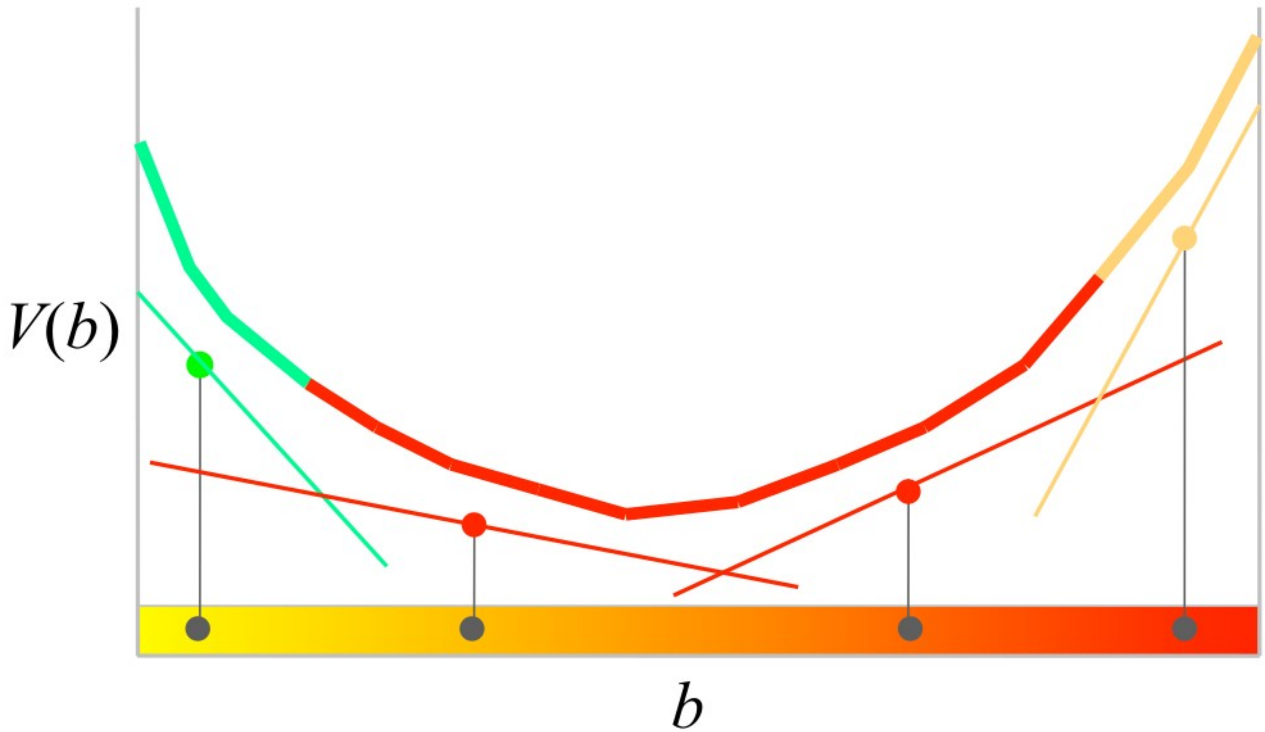


Is it possible to solve these kind of challenges?

- How to solve complex POMDPs
- Discrete valued POMDPs
  - Point based POMDP
    - Approximating value function
    - Considering only part of belief space
  - Treating solution process as search
- Continuous / complex action and observation POMDPs
  - Reinforcement learning with function approximation

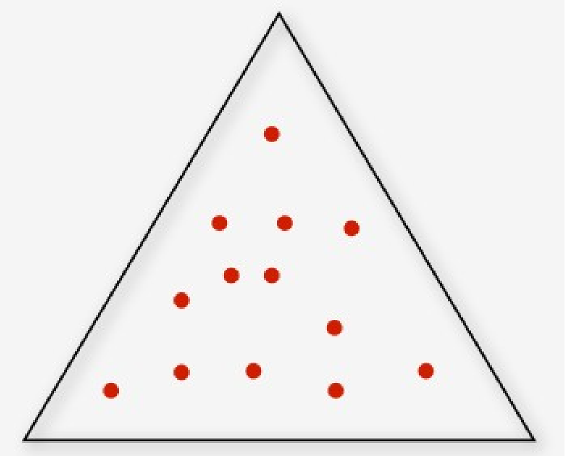
Approximating the value function

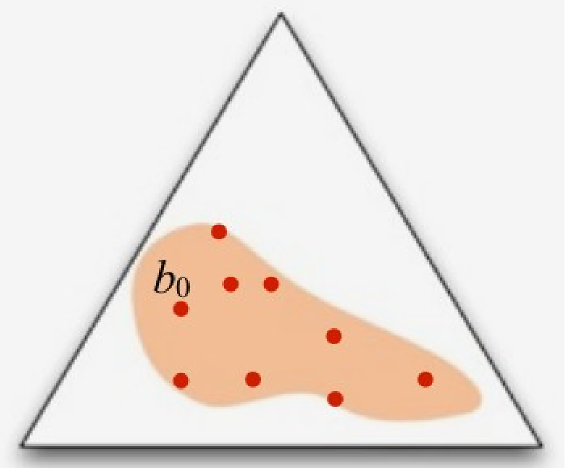
- Fixed number of beliefs (point-based approximation)
(e.g. Point-based value iteration, Pineau 2003)



Belief-space sampling

- Instead of calculating back-ups for whole belief space, use a set of points to approximate
- Instead of using points uniformly, use a set of points reachable from a starting belief





Point-based POMDP approaches

- PBVI, Pineau et al., 2003
  - Sample reachable points under arbitrary policy
- HSVI, Smith et al., 2004, SARSOP, Kurniawati et al., 2008
  - Use value function to sample reachable points
- Point-based methods help with larger belief spaces

Can we find an even better way to concentrate on the most relevant part of belief space?

On-line approaches

- Idea: Search reachable beliefs from current belief

- Basic algorithm
  - Plan starting from current belief
  - Execute first step
  - Update belief
  - Repeat

Similar idea to receding horizon optimal control!

Off-line vs on-line approaches

Off-line

- Plan for all beliefs
- High computational cost
- Fast online execution
- Significant implementation effort
- Cannot handle changing environment

On-line

- Plan for current belief
- Lower computational cost
- Slower online execution
- Easier to implement
- Can handle changing environments

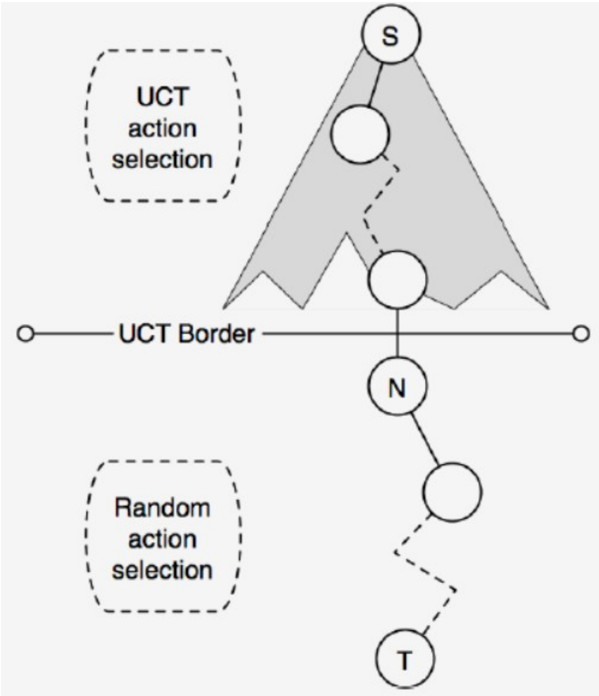
On-line planning with tree search

- Build a search tree from current belief
  - Start from a tree with one node corresponding to current belief
  - Choose a node to expand
  - Choose an action based on (optimistic) heuristic
  - Choose an observation based on another heuristic
  - Expand tree and backup back to root
  - Repeat
- Execute the best action
- Update belief
- Repeat

Does search sound familiar?
Have we seen something similar on the course?

Reminder: Monte-Carlo tree search

- From start node *S* choose actions to walk down tree until reaching a leaf node
- Choose an action and create a child node *N* for that action
- Perform a **random** roll-out (take random actions) until end of episode (or for a fixed horizon)
- Record returns as value for *N* and back up value to root



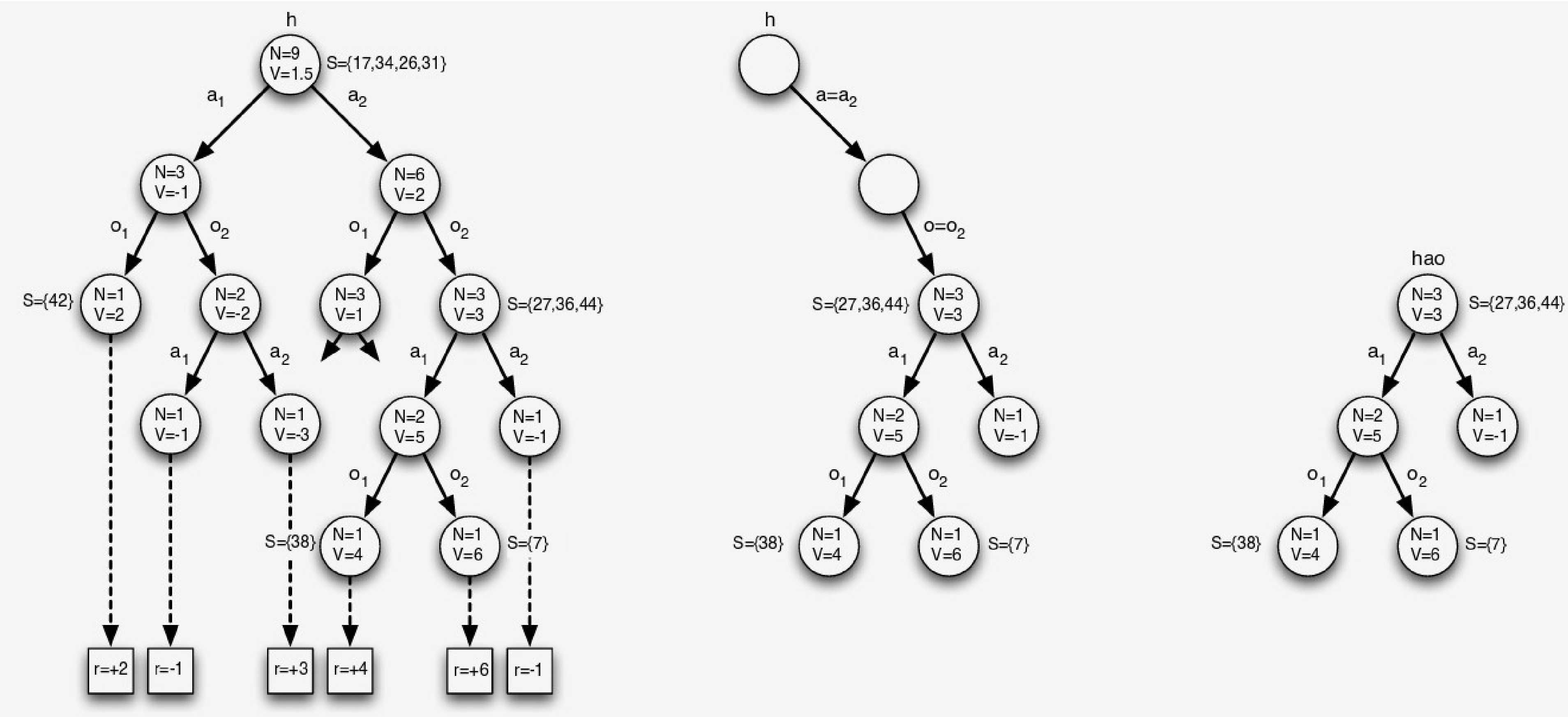
Remember MDPs!

In POMDPs, we do not know the state
→ how to use MCTS with POMDPs?

From MCTS to POMCP (Silver&Veness, 2010)

- Extension of MCTS to POMDPs
- Search tree represents *histories* (actions and observations) instead of states
- Belief state approximated by a *particle filter*
  - After taking an action, update belief by sampling particles by using simulation and keeping ones with true observation
- Each node has visitation count, mean value and particles (states)

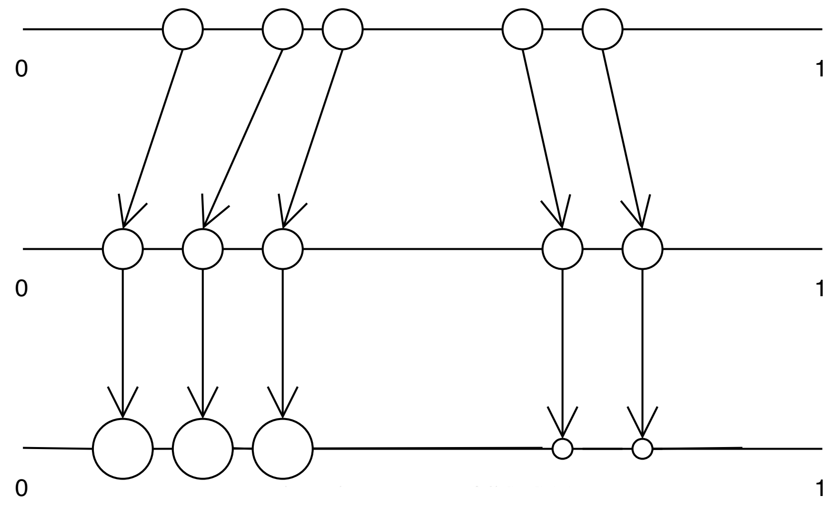
POMCP example



Silver&Veness, 2010

Particle filter for belief updates

Main idea: update *belief*, represented by a *finite set of states (= particles)*, using *action* and *observation*



Using *action* sample next states from current belief

Weight sampled states using observation probabilities and normalize weights

If desired, resample particles to get rid of particles with very small probabilities

POMDPs with large action and observation spaces

- How to handle POMDPs with continuous observations and actions?
- How to handle POMDPs with high-dimensional, e.g. image, observations?
- Possible solutions:
  - Kalman filter + optimal control
  - Discretization / simplification of continuous / complex values
  - Policy gradient / value iteration / actor-critic (Lectures 1 – 6) but how?

Reinforcement learning with POMDPs

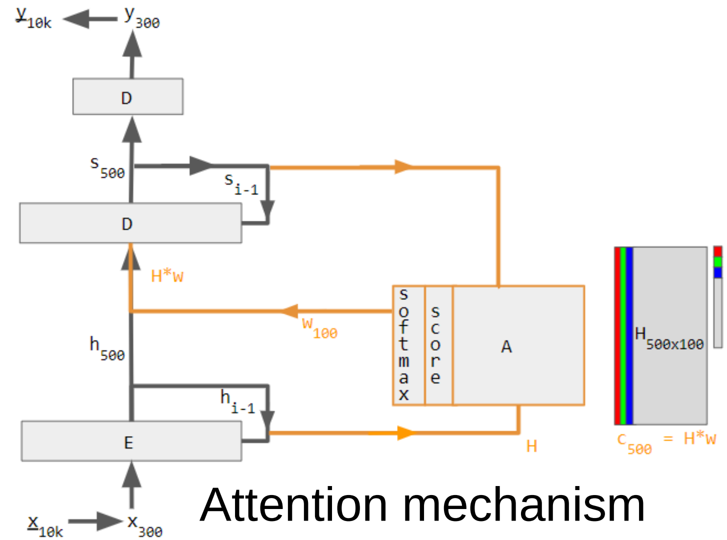
- Sufficient statistics for optimal decision making in POMDPs:
  - Belief, a probability distribution over states $b(s)$
  - Full history of actions and observations $a_1, z_1, \ldots, a_t, z_t$
- Problems:
  - Belief computation requires dynamics/observation model
  - History grows with each time step
- Solution:
  - Put history into a "*memory representation*" m
  - Replace $\pi(s), V(s), Q(s,a)$ with $\pi(m), V(m), Q(m,a)$ and apply policy gradient, value iteration, actor-critic, or other methods

Full history may be too long?

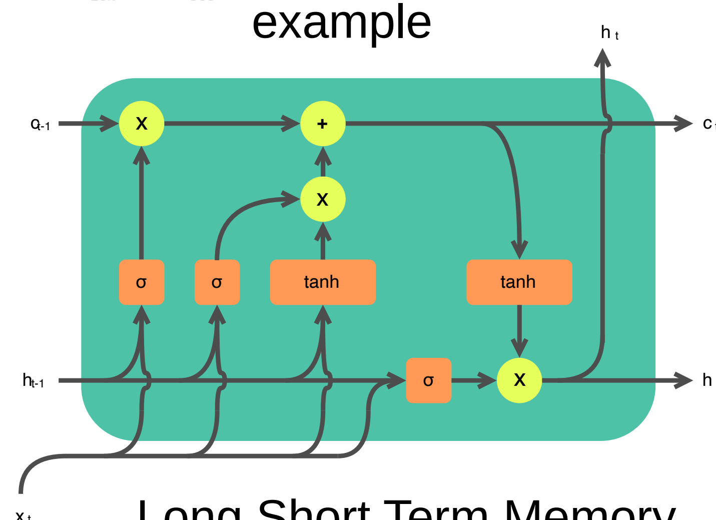
Memory representations

- Direct mapping: $m_t = f(a_1, z_1, \ldots, a_t, z_t)$
  - Truncated history

    $m_t = f(a_{t-N}, z_{t-N}, \ldots, a_t, z_t)$
  - Look at only parts of the history: *attention*
- Recurrent memory: $m_t = f(a_t, z_t, m_{t-1})$
  - Memory state part of neural network
  - External memory state



Attention mechanism example



Long Short Term Memory

Summary

- Key to more efficient POMDP solutions is to consider only parts of belief space
  - Off-line approaches sample over reachable beliefs
  - On-line approaches sample over currently reachable beliefs
- Real-world problems are complicated and solutions require approximations
  - Careful choices in modeling are important

Current directions in reinforcement learning (RL)

- Challenges: sample efficiency, computational efficiency, safety
- Offline RL
- Model-based RL
- Exploration in RL
- Multi-agent RL
- Safe RL
- POMDPs
- Deep RL
- Combining different approaches: offline/online, model-free/model-based, planning
- Many other topics