Aalto University
School of Electrical Engineering

ELEC-E8125 Reinforcement Learning
Large POMDPs

Ville Kyrki

26.11.2019

Today

- POMDPs towards largish real world problems.

Learning goals

- How to solve complex POMDPs by
  (i) approximating value function,
  (ii) considering only part of belief space, and
  (iii) treating solution process as search.

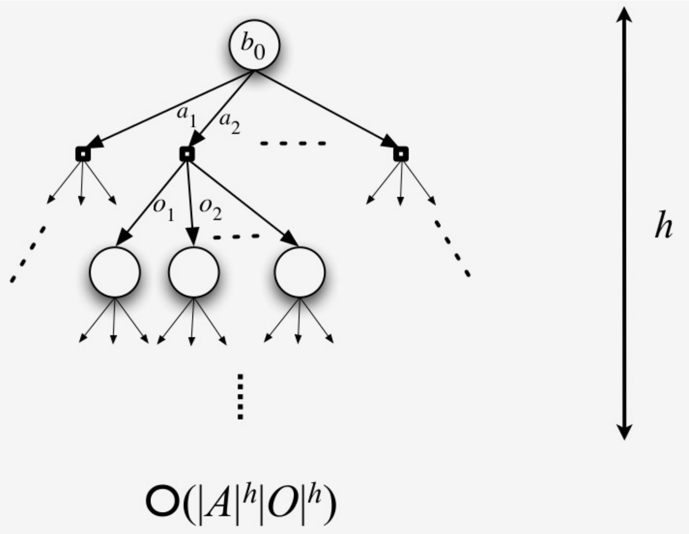
POMDP application examples

- Intention-aware planning for autonomous vehicles (Bai et al., 2015)
- Grasping (Hsiao et al. 2007, Horowitz et al. 2013)
- Manipulation of multiple objects (Pajarinen&Kyrki 2015)

“Curses” of POMDP

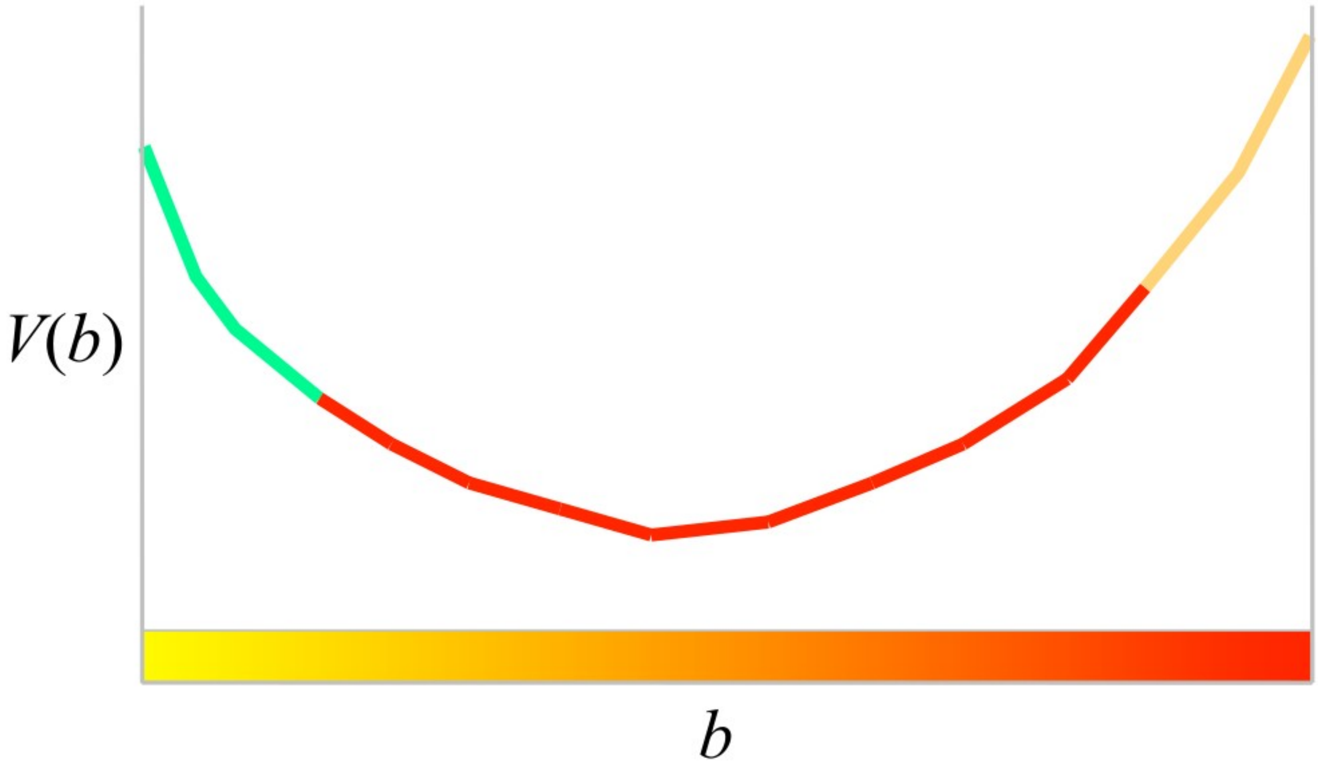
- Curse of dimensionality
  - Complexity exponential in number of states
  - Double exponential in dimensionality of state space

- Curse of history
  - Complexity exponential in length of history



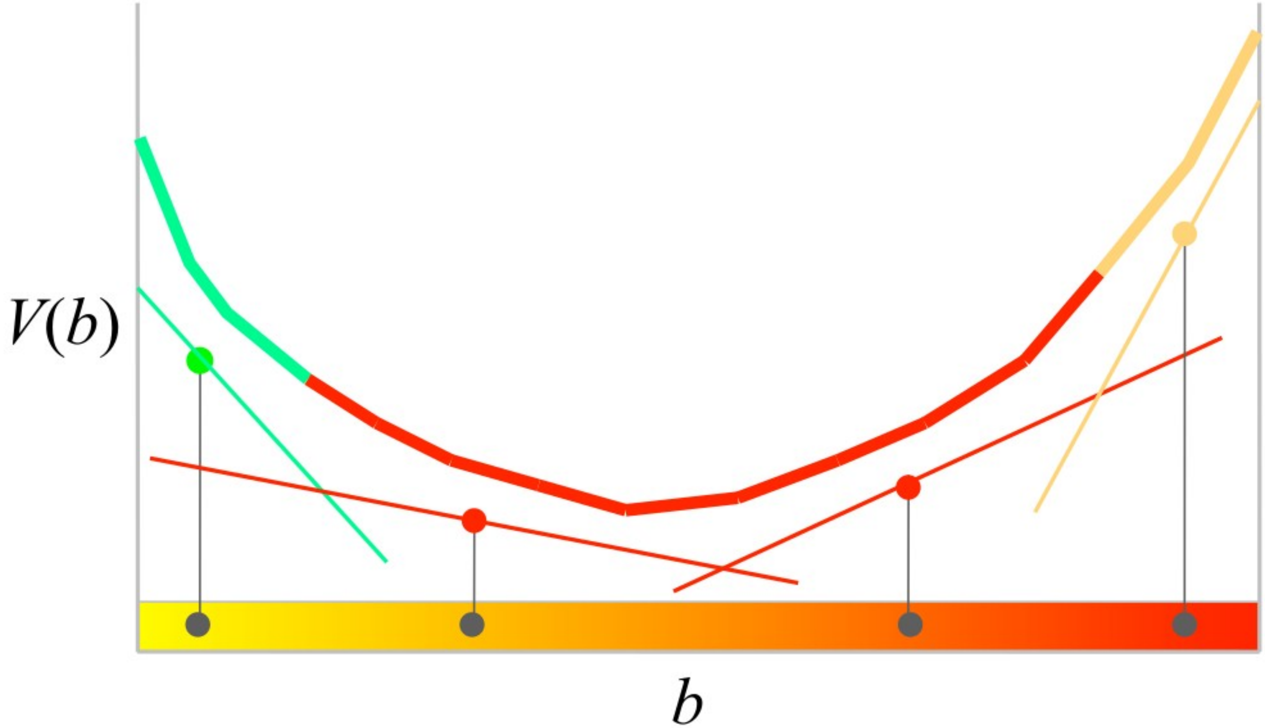
Curse of history with value iteration

- Number of possible policies is exceedingly high.



Approximating value function

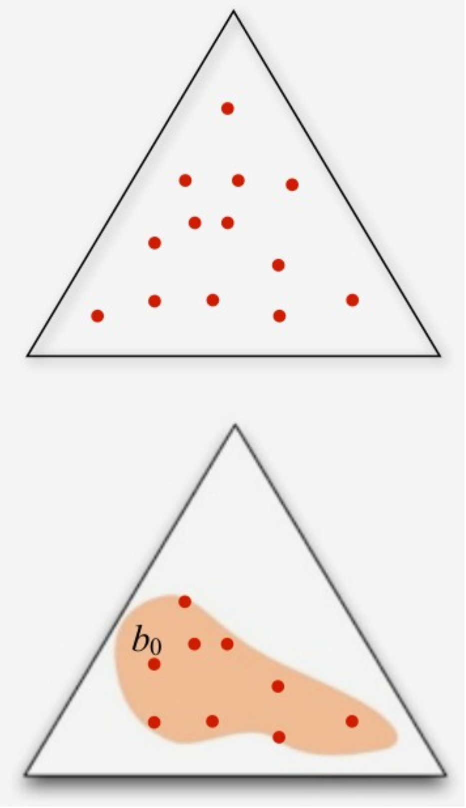
- Point-based approximation (e.g. Point-based value iteration, Pineau 2003)



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.
- SARSOP, Kurniawati et al., 2008
  - Sample reachable points under optimal policy.
- 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 state.

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

On-line planning equates to 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.

Forget partial observability for now.

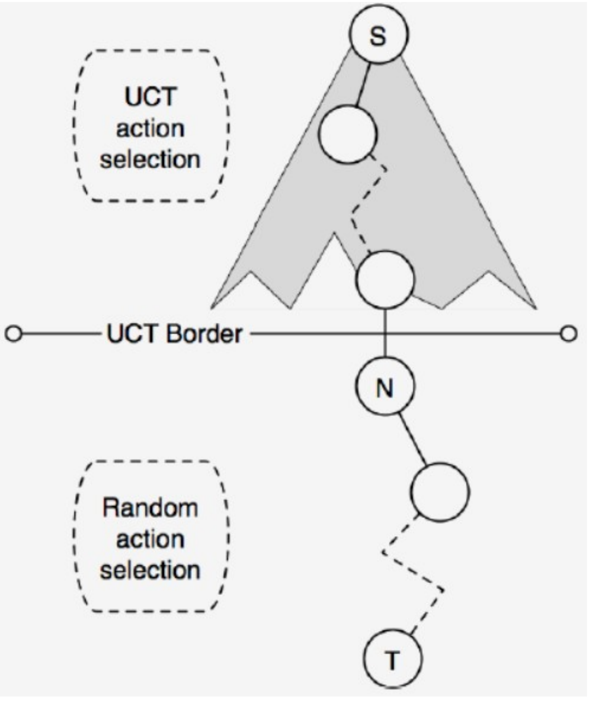
A step back: Monte Carlo tree search

- Search method for optimal decision making.
- State-of-the-art for playing games (e.g. Alpha Go).

- Iteratively builds a search tree.
- Phases:
  - Selection: Choose a promising node to expand.
  - Expansion: Add a new node.
  - Simulation: Simulate value for new node.
  - Backpropagation: Back-up value to root (update values for parents).

MCTS operation

- 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!

Node selection in MCTS

- Node selection has to balance exploration and exploitation (note difference to RL, here x&x is made only in computation).
- First choose
- Upper confidence bound 1 (UCB1) on trees (UCT).
  - A bound for value of a node (Kocsis&Szepesvari, 2006).

$$Q^{+}(x,u)=Q(x,u)+c\sqrt{\frac{\log N(x)}{N(x,u)}}$$

Positive exploration constant

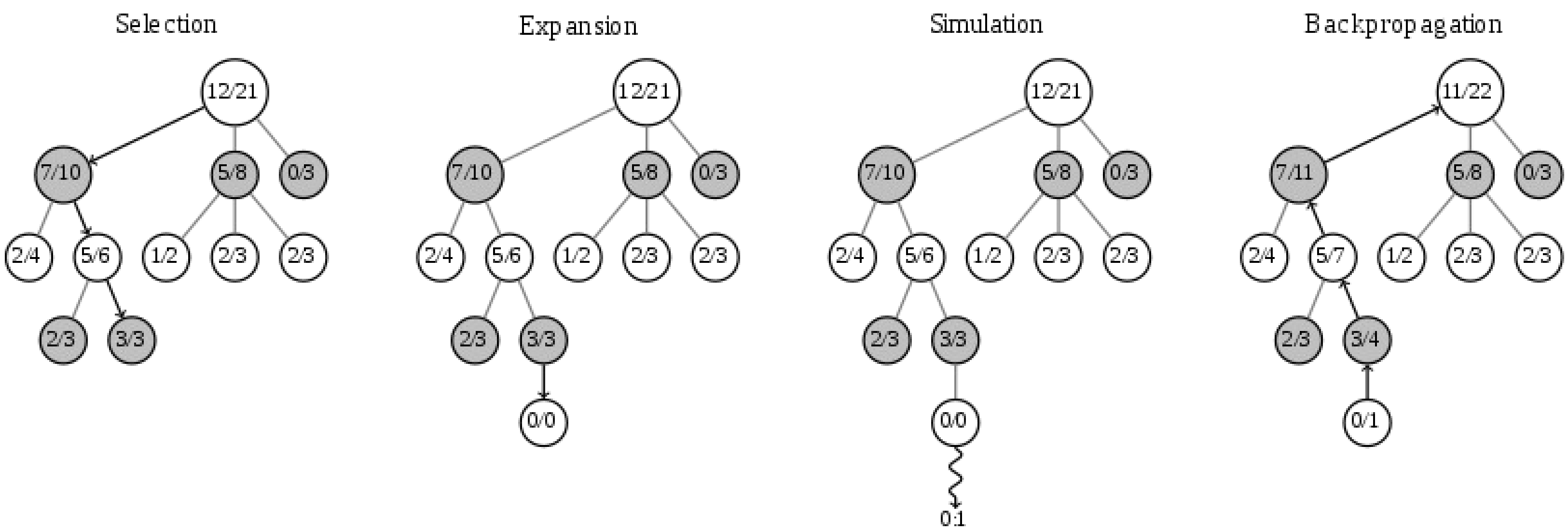
Visitation count

MCTS simulation phase

- Perform (one or) several roll-outs from leaf node using random action selection.
- Stop at terminal state or until a discount horizon is reached.
- Estimate value of state as mean return of the *N* simulations: $V(x) = \frac{1}{N} \sum_i R_i$

MCTS: Example in game playing

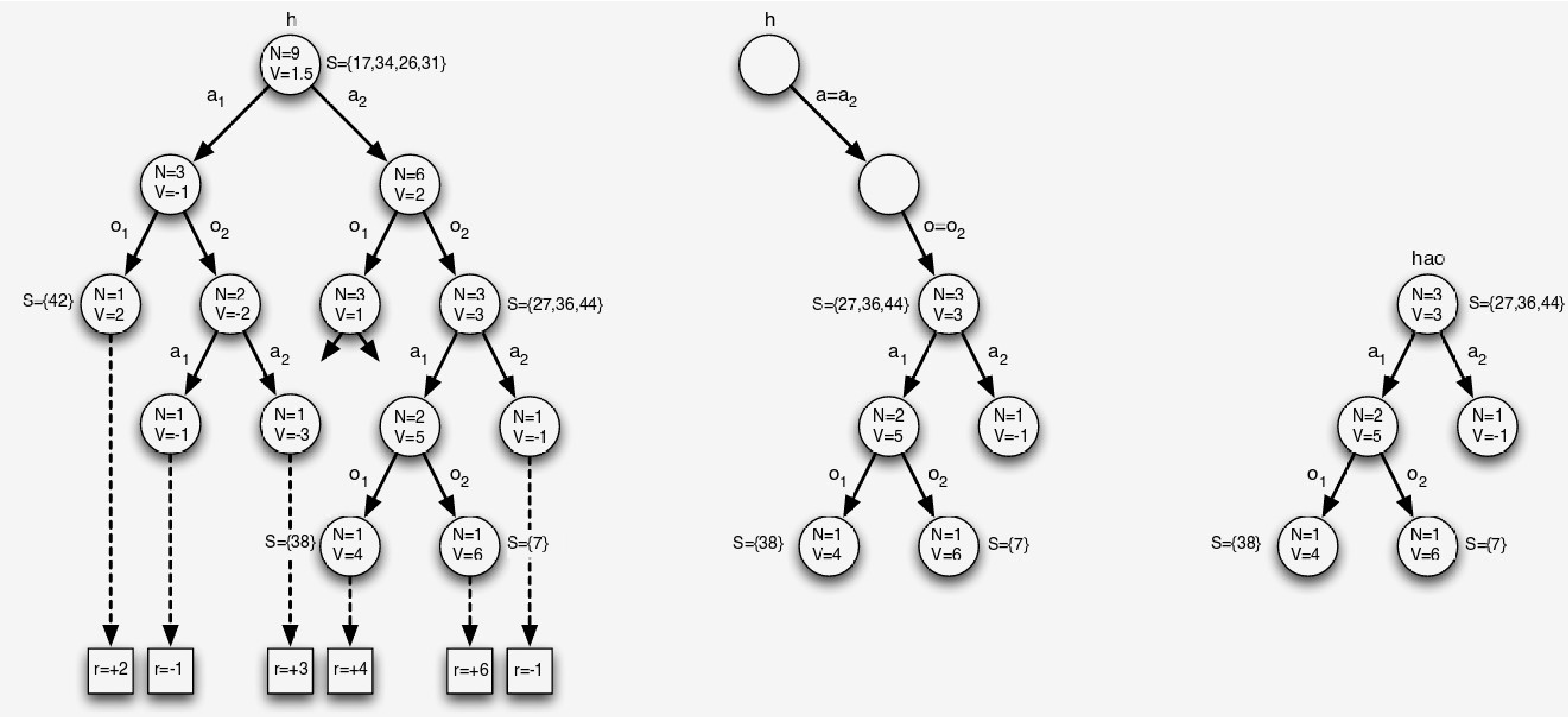
- Value number of won games.



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.

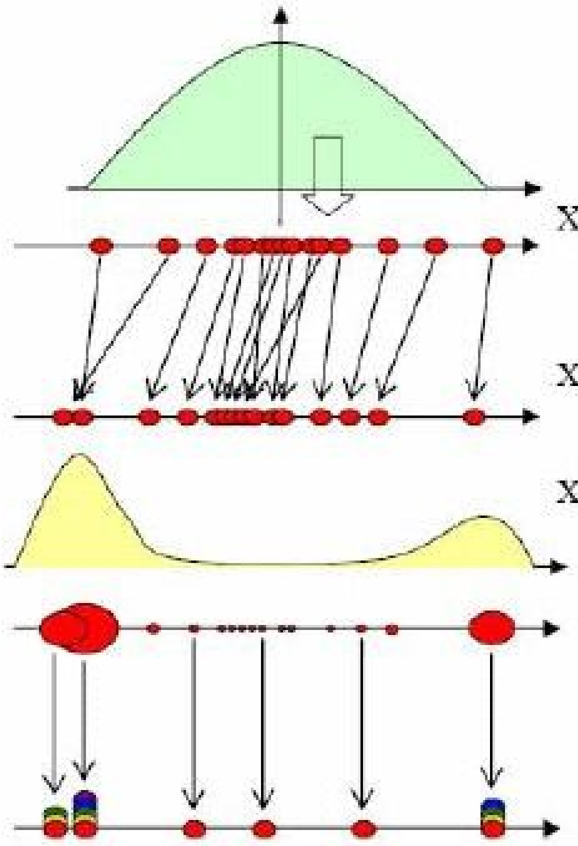
POMCP example



Silver&Veness, 2010

Recap (hopefully): Particle filter

- Starting from current belief, sample future.
- Calculate weights depending on observation probability.
- Resample according to weights.



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 environment

We didn't cover

- Other on-line approaches available, e.g. DESPOT (Somani et al., 2013).

- Current work towards combining off-line and on-line approaches.
  - E.g. using precomputed macro-actions.

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.