COMP4620 – Advanced Topics in AI
Partially Observable Markov Decision Processes (POMDP) 2/3

Hanna Kurniawati

http://users.cecs.anu.edu.au/~hannakur/

hanna.kurniawati@anu.edu.au

Australian National University | RESEARCH SCHOOL OF COMPUTER SCIENCE

Topics

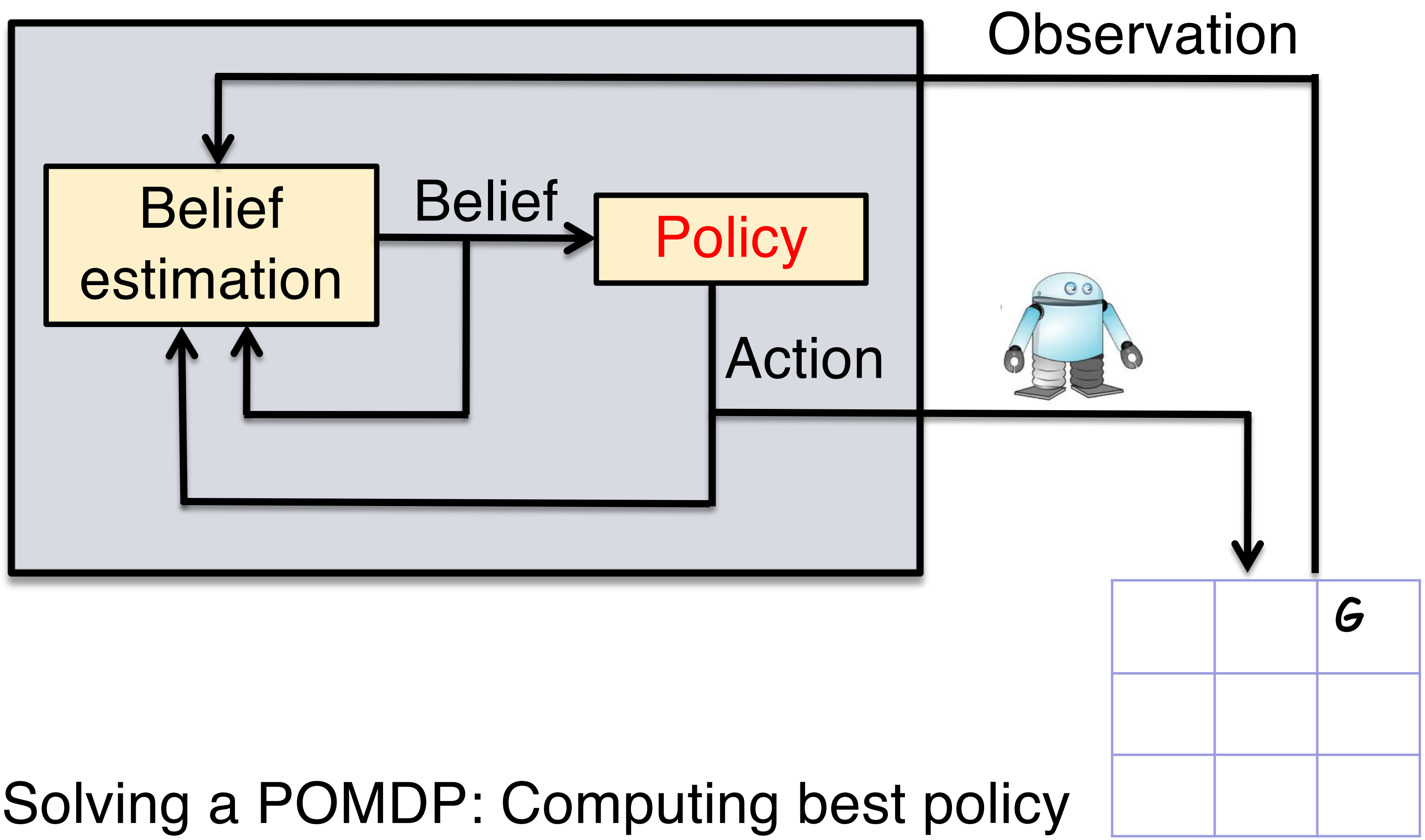
✓Lecture 1: What is POMDPs?

- Lecture 2: How do we solve POMDPs?
- Lecture 3: Applications of POMDPs in Robotics & Cyber

How to solve POMDPs?

- What does solving a POMDP means?
- Difficulty
- Approximate Solvers: Sampling-based
  - Offline Solvers
  - Online Solvers

Partially Observable Markov Decision Processes (POMDPs)



- Solving a POMDP: Computing best policy
- Policy: mapping from beliefs to actions.

“Best” policy

- Maps each belief to an action that satisfies the following objective function

$$V^*(b) = \max_{a \in A}\left(\underbrace{\sum_{s \in S} R(s,a)b(s)}_{\text{Expected immediate reward}} + \gamma \underbrace{\sum_{o \in O} P(o|b,a)V^*(b')}_{\text{Expected total future reward}}\right)$$

b': next belief after the system at belief b performs action a and observes o

γ: discount factor, $(0,1)$

A bit more formal about policy

- Usually denoted as π, it is a function that maps beliefs to actions.
  - Note that here, we focus on deterministic policy: The policy maps a belief to a single action, i.e., $\pi(b) \in A$
- Each policy π has an associated value V_π :

$$V_\pi(b) = \boxed{\sum_{s \epsilon S} R\big(s, \pi(b)\big) b(s)} + \gamma \boxed{\sum_{o \epsilon O} P(o|b, \pi(b)) V_\pi(b')}$$

- The best policy, π^*, is one that maximises the value at each belief, i.e.: $\pi^*(b) = argmax_{\pi \epsilon \Pi}\, V_\pi(b)$
  - Π: The set of all possible policies
- Solving a POMDP problem means finding π^*

How to solve POMDPs?

✓What does solving a POMDP means?

- Difficulty
- Approximate Solvers: Sampling-based
  - Offline Solvers
  - Online Solvers

POMDP as Belief MDP

- POMDP can be viewed as MDP, but in the belief space
  - This MDP is often called Belief MDP
- Belief MDP:
  - S: The POMDP belief space (ie., the set of all possible beliefs)
  - A: The POMDP action space
  - Transition: Note that given a pair of action—observation, the next belief is deterministic

    $\tau(b, a, b') = P(o|b, a)$ if $b' =$ next belief after a is performed from b and o is perceived and 0 otherwise

    - Recall from last Wednesday: $b'(s') = \frac{P(o|s', a,) \sum_s P(s'|a, s) b(s)}{\sum_{s''} (P(o|a, s'') \sum_s P(s''|a, s) b(s))}$ and

    $P(o|b, a) = \sum_{s''} (P(o|a, s'') \sum_s P(s''|a, s) b(s))$
  - R: $R(b, a) = \sum_{s \in S} R(s, a) b(s)$

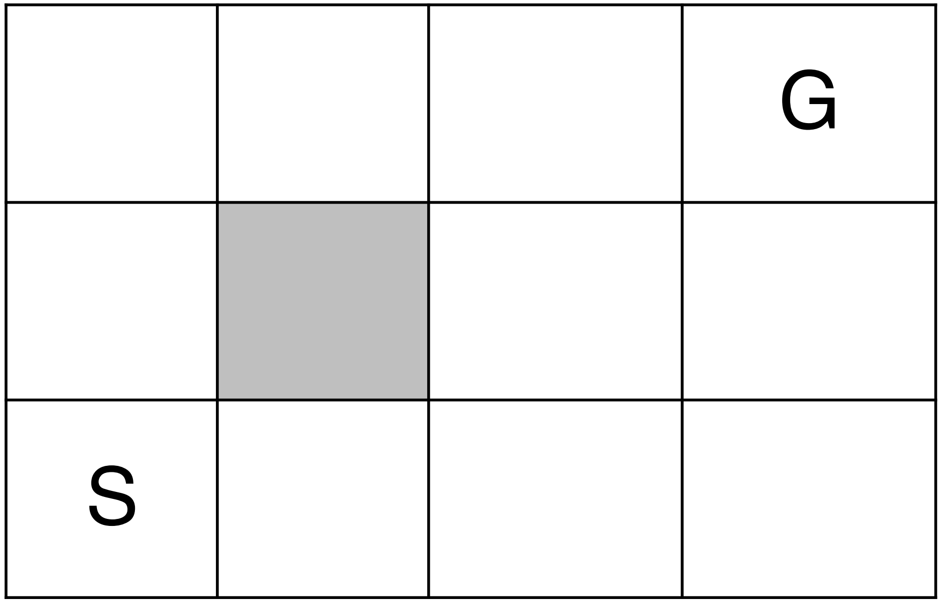
So …

- Solving POMDP is the same as solving MDP
- Well, there's one issue: We need to solve MDP with uncountable state space
  - Belief space is continuous
- Turns out, this is not that easy

Difficulty

- Solving MDP is a P problem – P: Problems that can be solved using polynomial time algorithm
- Solving POMDP is a PSPACE-hard problem – PSPACE: Problems that can be solved using polynomial amount of space.
  - Remember NP? P ⊆ NP ⊆ PSPACE
  - A problem X is Y-hard: All problems in Y is reducible to X

For a long time …



POMDP is viewed as impractical and abandoned

How to solve POMDPs?

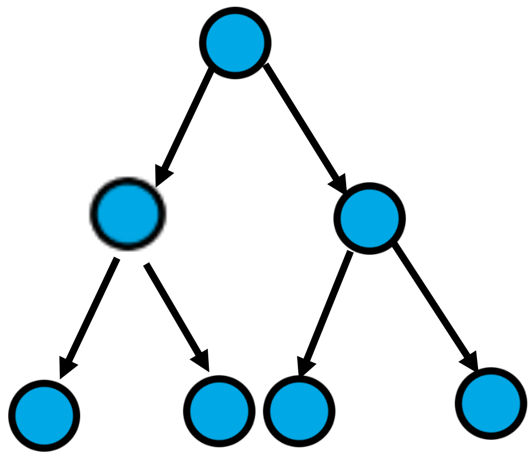
✓What does solving a POMDP means?

✓Difficulty

- Approximate Solvers: Sampling-based
  - Offline Solvers
  - Online Solvers

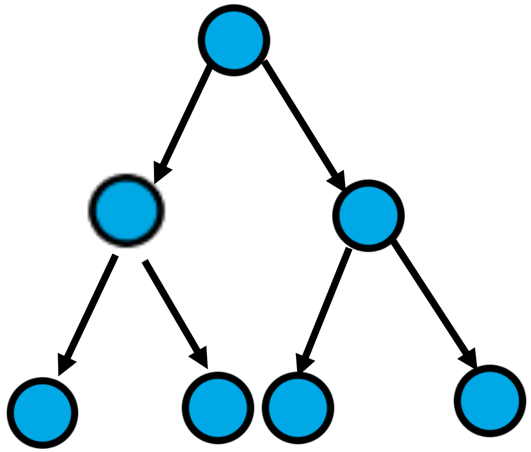
Approximate the optimal value

- Using optimal value for a finite step discounted value function
  - Based on a finite step policy, represented as a policy tree
  - Policy tree: A tree where the nodes is associated with actions and the edges are labelled with observations



Policy Tree

- Given a belief b:
  - The agent starts to execute the action associated with the root node
  - Suppose the agent then perceives an observation o, it will then execute the action associated with the child node of n following the edge labelled with o
  - The process repeats



The Value

- The value of executing a policy tree p from a belief can be computed as

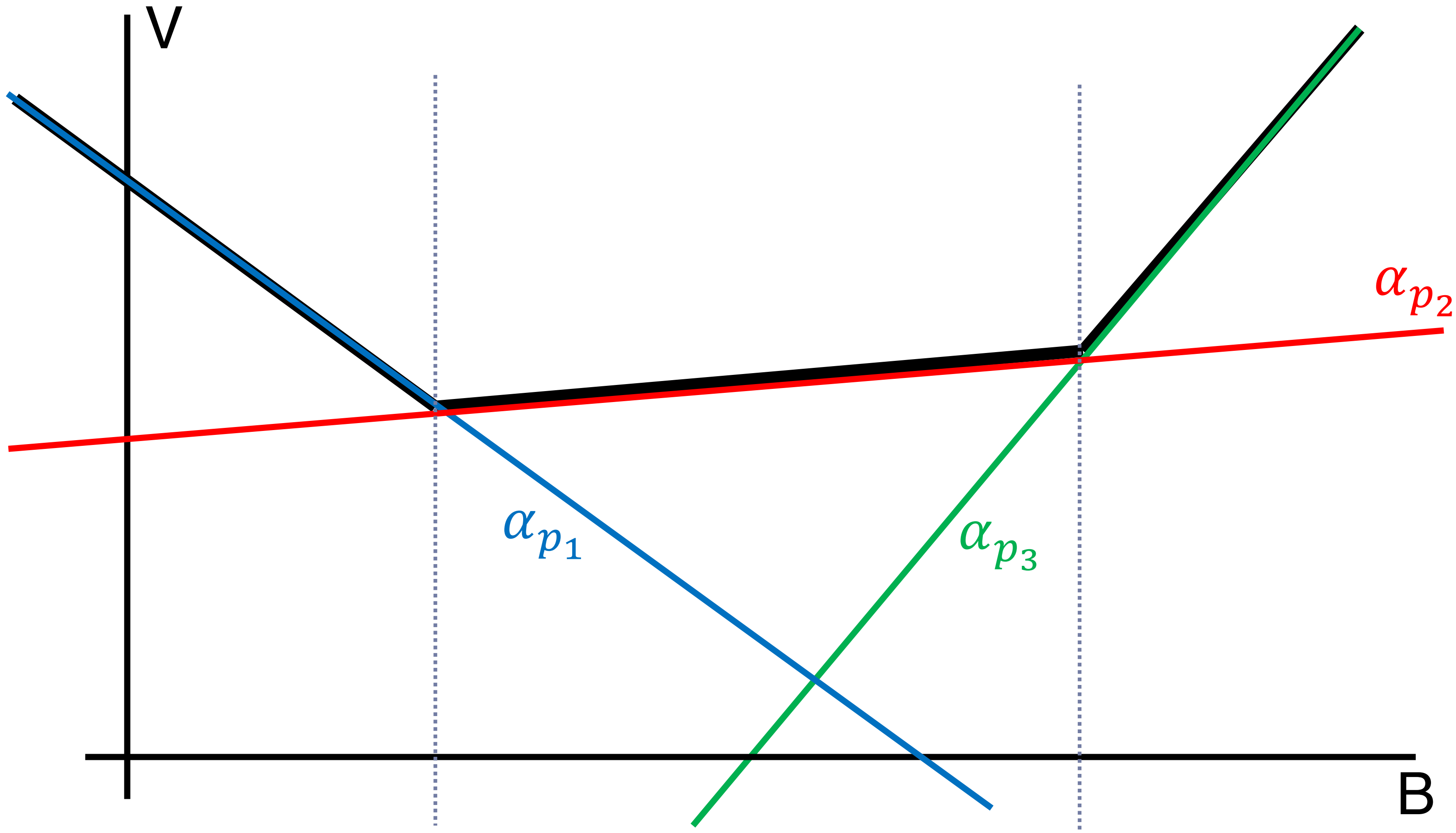
$V_p(b) = \sum_{s \epsilon S} V_p(s) b(s)$ where

$$V_p(s) = R(s, p(a)\,) + \sum_{o \epsilon O} \sum_{s' \epsilon S} Z(s', p(a), o) T(s, p(a), s') V_{p(a,o)}(s')$$

- A more famous name: α-vector

$$V_p(b) = \sum_{s \epsilon S} \boxed{V_p(s)} b(s) = \alpha_p . b$$

Geometrically …



The value of executing a policy tree is linear over the belief space. The optimal value function is the upper envelope of the values of policy trees

Policy Representation: α-vector

- The policy is represented as a set Γ of α-vectors
- Given a belief b, the agent finds the α-vector that maximizes the value function of b, i.e.: $V^*(b) = \max_{\alpha \epsilon \Gamma} \alpha . b$
- Suppose this α-vector is associated with a policy tree *p*, then the agent will execute the action associated with the root node of *p*

Constructing the policy
(aka the set of α-vectors)

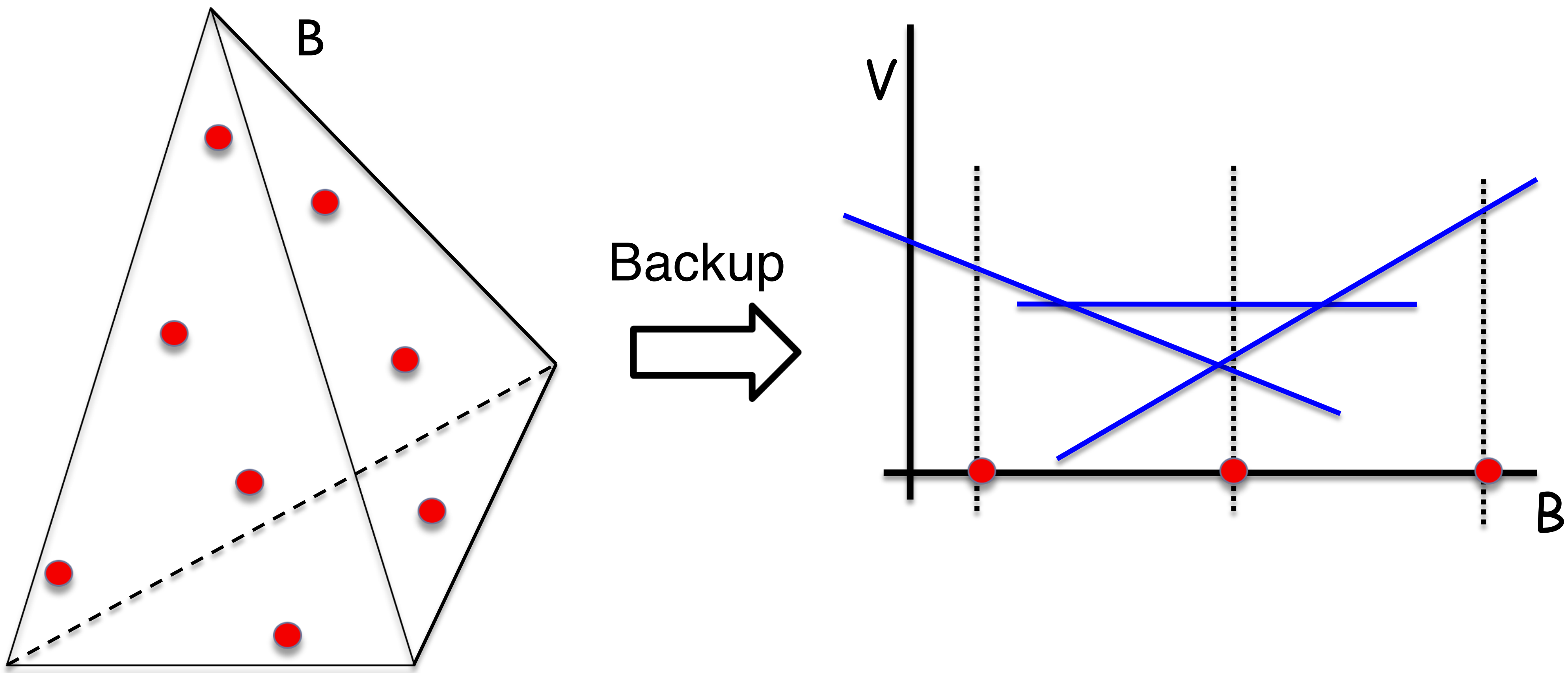
- Given the current set of α-vectors and its associated policy trees, construct a new policy tree by
  - Selecting an action to be associated with a root node
  - Add edges to this root node, an edge per observation
  - For each edge, select the current policy tree to be the descendent of the root node via the edge
- Bellman update: which action & policy tree?

$$V(b) = \max_{a \epsilon A} \left[\begin{array}{c} \sum_{s \epsilon S} R(s,a)b(s) + \\ \gamma \sum_{o \epsilon O} \max_{\alpha \epsilon \Gamma} \sum_{s \epsilon S} \sum_{s' \epsilon S} Z(s',a,o)T(s,a,s')\alpha(s')b(s) \end{array} \right]$$

(Naïve) Bellman Update with α-vector Representation

- Would like to find best action for each belief
- In the worst case, each iteration can generate $|A||\Gamma|^{|O|}$ new policy trees and hence α-vectors
- No surprise that both computational time & memory requirements are massive

Point-based POMDP



Point-based Value Iteration (PBVI)

- An anytime algorithm, meaning: If the algorithm stops at any point in time, it will give a solution.
  - Of course the quality of the solution will generally be better if more time is given
- Idea:
  - Sample a set of points from the belief space
  - Each belief is associated with a single α-vector
- Therefore the number of α-vectors is limited to the number of samples
- 1st to generate good solution for an 870 states POMDP problem (tag), though it takes 50 hours

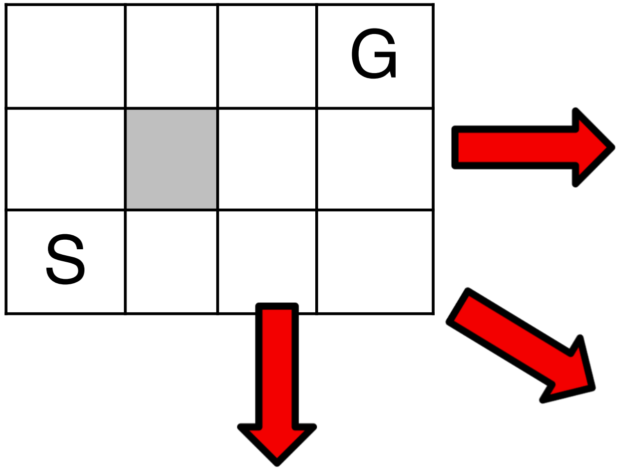
Joelle Pineau, Geoff Gordon, and Sebastian Thrun. "Point-based value iteration: An anytime algorithm for POMDPs." *International Joint Conference on Artificial Intelligence.* 2003.

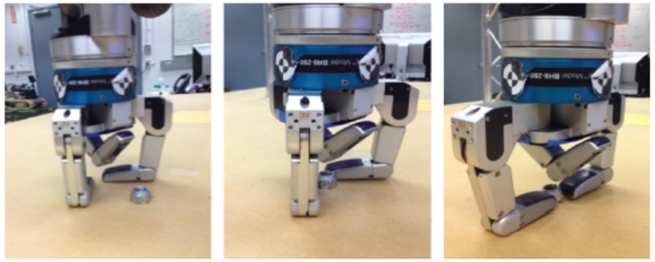
Successive Approximations of the Reachable Space under Optimal Policies (SARSOP)

- Improve sampling strategy of PBVI
  - Represent the set of beliefs reachable from the initial belief as a belief tree
  - Maintain upper & lower bound
  - Sampling a belief = expanding a node of the tree
    - Sample a node, sample an action and an observation, compute the next belief. This next belief is the new sample.
  - Action selection: Predict optimal action
  - Observation selection: Choose the observation that reduces the gap between upper & lower bound (to improve future value estimate)
- Can get better policy than PBVI on the 870 states tag problem after 6 seconds and beyond…

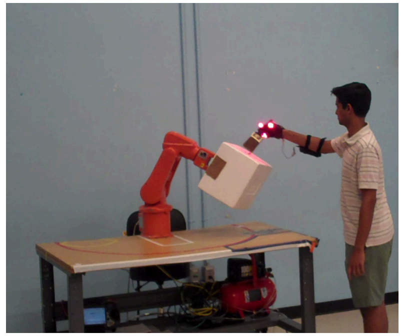
Hanna Kurniawati, David Hsu, and Wee Sun Lee. "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces." *Robotics: Science and systems*. 2008.

Some Progress

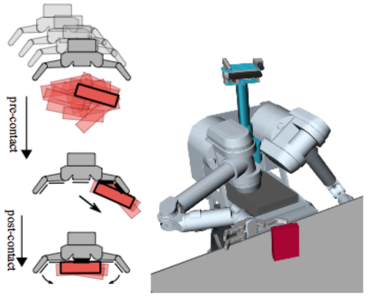




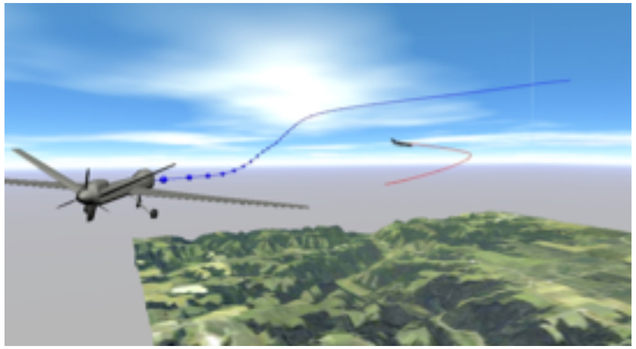
Horowitz &
Burdick (ICRA'13)



Nikolaidis, et.al. (HRI'15)



Koval, et.al. (RSS'14)



Temizer, et.al. (Lincoln Lab TR'09)
Improve safety of TCAS by 20X

Wang, et.al. (ICAPS'15)
Learn interaction model of
bees with reduced data

Bandyopadhyay, et.al. (early work
leading to nuTonomy)

How to solve POMDPs?

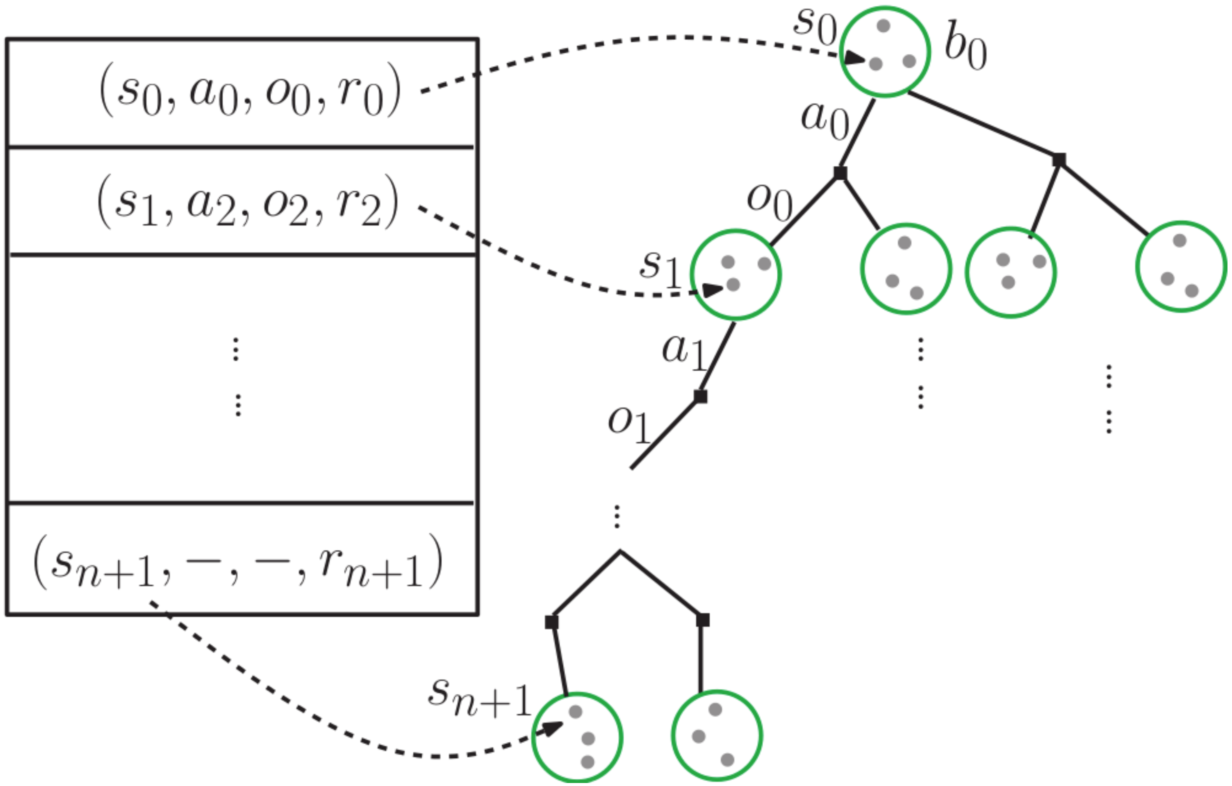
- ✓ What does solving a POMDP means?
- ✓ Difficulty
- Approximate Solvers: Sampling-based
  - ✓ Offline Solvers
  - Online Solvers

Basic Structure

- Online: Interleave policy computation & execution
  - At each step, compute the best action to perform from the current belief, execute this action, perceive observation, update the belief, and the process repeats
- Anytime
- Policy representation: Belief tree
  - A tree where the nodes are beliefs and an edge from node b to b' means there is an action—observation pair (a, o), such that the belief associated with b' is the subsequent belief after a is executed from the belief associated with b and o is perceived

Basic Structure

- Sample: History, usually MCTS style
- Action selection: UCB



Some notes

- A belief is a sufficient statistics of the entire history of actions—observations
  - A POMDP policy accounts for the entire history of actions – observations
  - Sometimes, policy is represented as a mapping from this history of actions – observations (rather than beliefs) to the subsequent action

The Problems & Some of Our Solutions

- Large state space [Kurniawati, et.al. (RSS’08)]
- Large observation space [Kurniawati, et.al. (RSS’11, Auro’12 invited)]
- Long planning horizon [Kurniawati, et.al. (ISRR’09, IJRR’11 invited)]
- Model may change [Kurniawati & Patrikalakis (WAFR’12), Kurniawati & Yadav (ISRR’13)]
- Large action space [Seiler, et.al. (ICRA’15, best paper award finalist), Wang, et.al. (ICAPS’18)]
- Complex dynamics [Hoerger, et.al. (WAFR’16), Hoerger et.al. (ISRR’19)]

Implementation: http://rdl.cecs.anu.edu.au/software

How to solve POMDPs?

- ✓What does solving a POMDP means?
- ✓Difficulty
- ✓Approximate Solvers: Sampling-based
  - ✓Offline Solvers
  - ✓Online Solvers

Next: Applications of POMDPs
+ What's Next?