EECE 5550
Mobile Robotics

Lecture 16: Partially-Observable Markov Decision Processes (POMDPs)

Derya Aksaray

Assistant Professor

Department of Electrical and Computer Engineering

Northeastern University

Recap

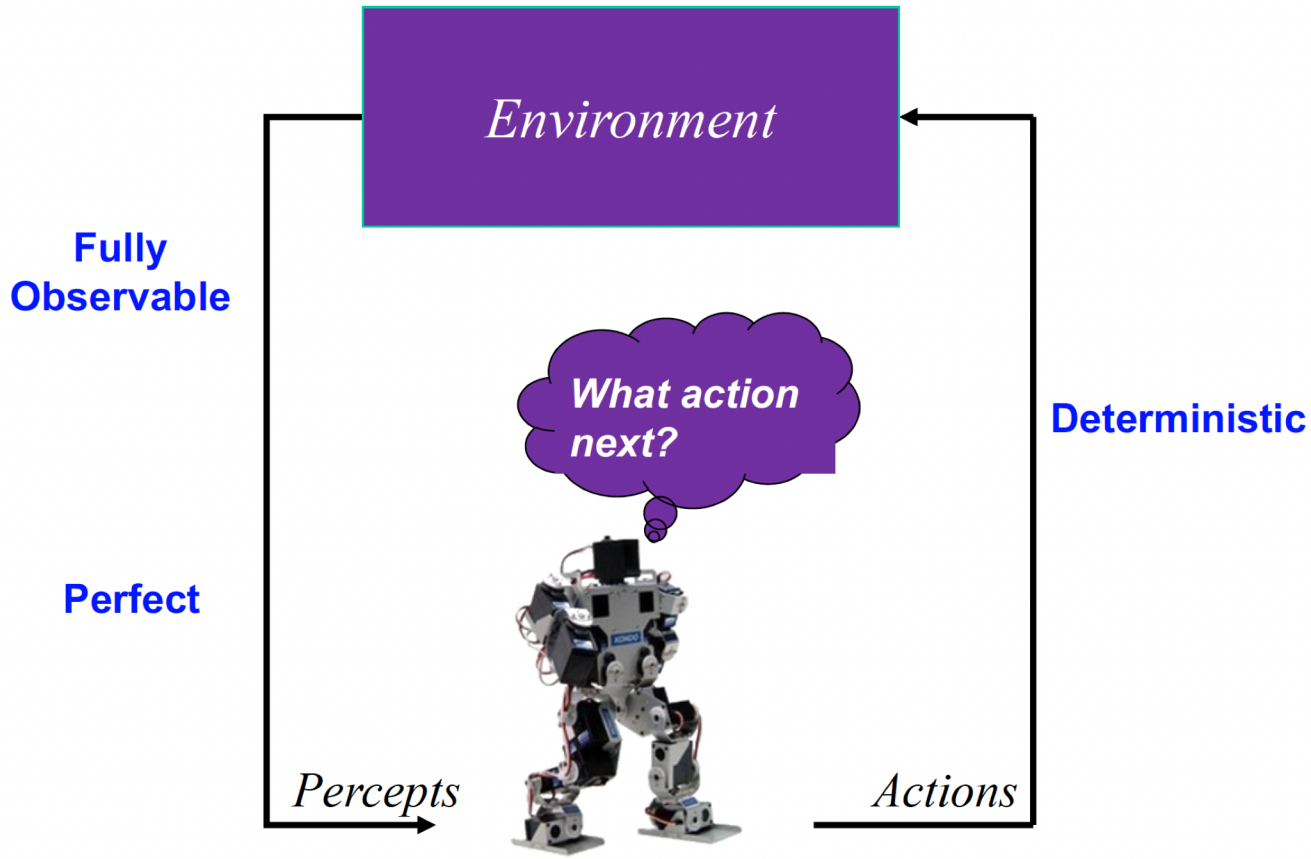
Markov decision process

- Value iteration
- Policy iteration
- Reinforcement learning
- Perfect knowledge about state

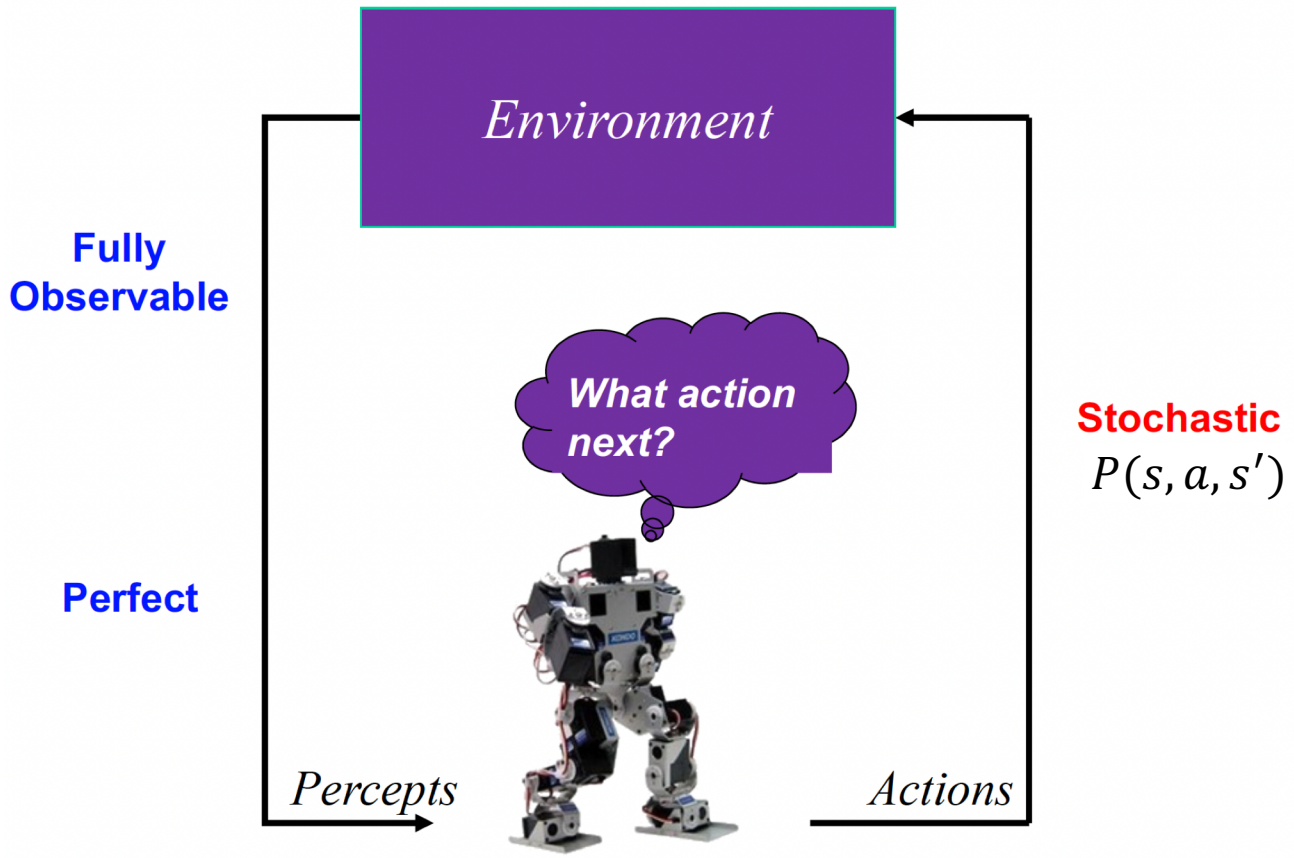
Partially-observable Markov decision process

- State is not fully observable!

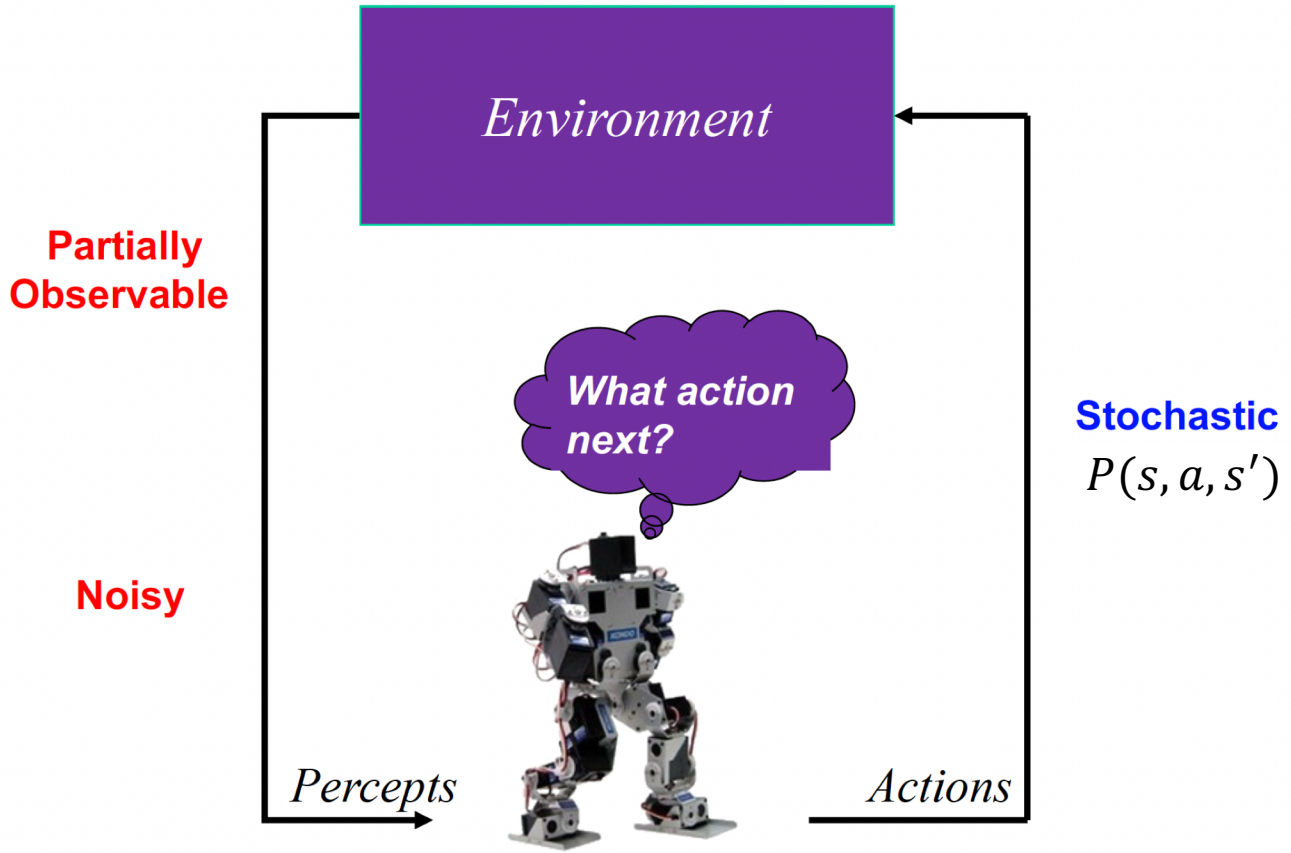
Classical planning



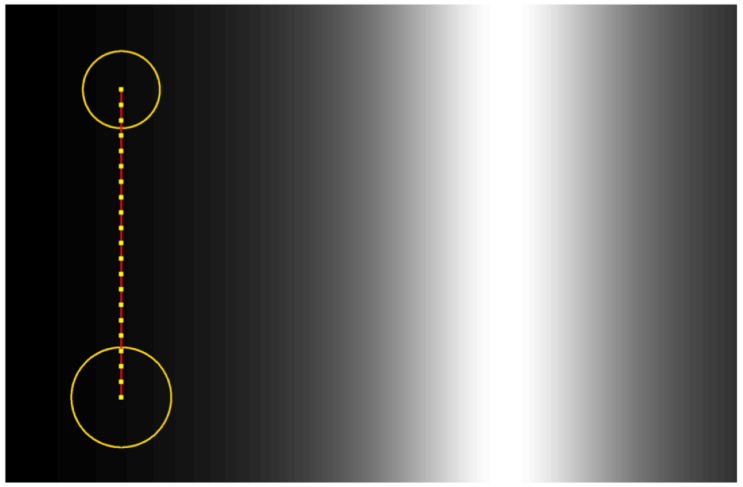
Stochastic (MDPs)

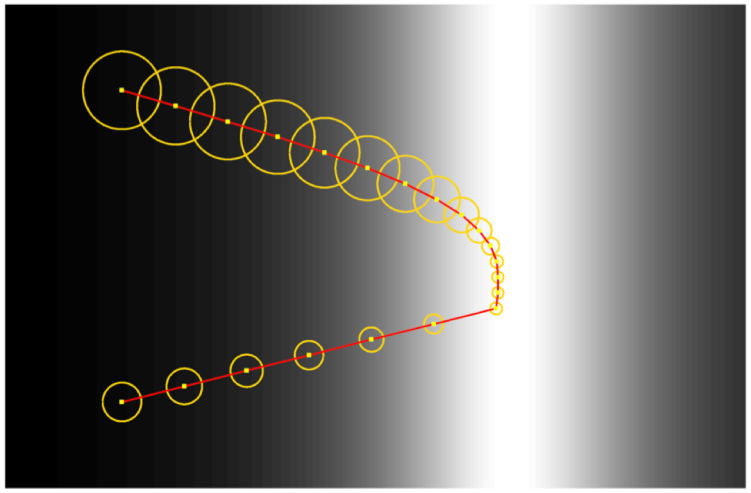


Partially-observable Stochastic (POMDP)



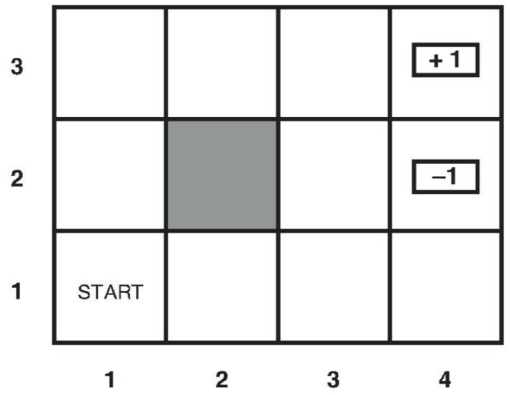
Example

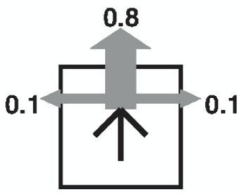




https://groups.csail.mit.edu/robotics-center/public_papers/Platt10.pdf

Example





0.111	0.111	0.111	0.000
0.111		0.111	0.000
0.111	0.111	0.111	0.111

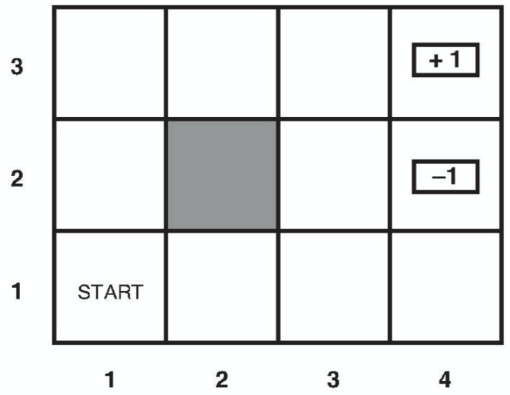
(a)

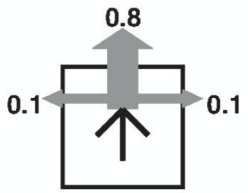
Initial belief

Assumptions:

- The agent does not have any sensor.
- Initially, it does not now where it is.
- Each white cell has -0.04 reward.

Example





Assumptions:

- The agent does not have any sensor.
- Initially, it does not now where it is.
- Each white cell has -0.04 reward.

0.111	0.111	0.111	0.000
0.111		0.111	0.000
0.111	0.111	0.111	0.111

(a)

Initial belief

0.300	0.010	0.008	0.000
0.221		0.059	0.012
0.371	0.012	0.008	0.000

(b)

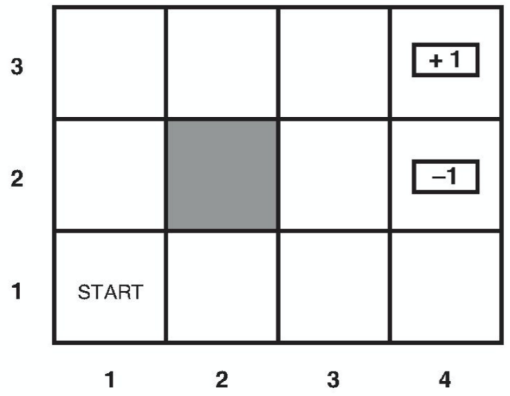
Move Left 5 times

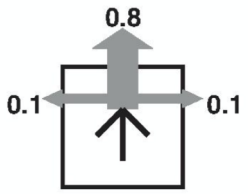
0.622	0.221	0.071	0.024
0.005		0.003	0.022
0.003	0.024	0.003	0.000

(c)

Move Up 5 times

Example





Assumptions:

- The agent does not have any sensor.
- Initially, it does not now where it is.
- Each white cell has -0.04 reward.

Taking actions:

- not only for reaching the goal state
- but also for collecting observations to have a better idea about the state.

0.111	0.111	0.111	0.000
0.111		0.111	0.000
0.111	0.111	0.111	0.111

(a)

Initial belief

0.300	0.010	0.008	0.000
0.221		0.059	0.012
0.371	0.012	0.008	0.000

(b)

Move Left 5 times

0.622	0.221	0.071	0.024
0.005		0.003	0.022
0.003	0.024	0.003	0.000

(c)

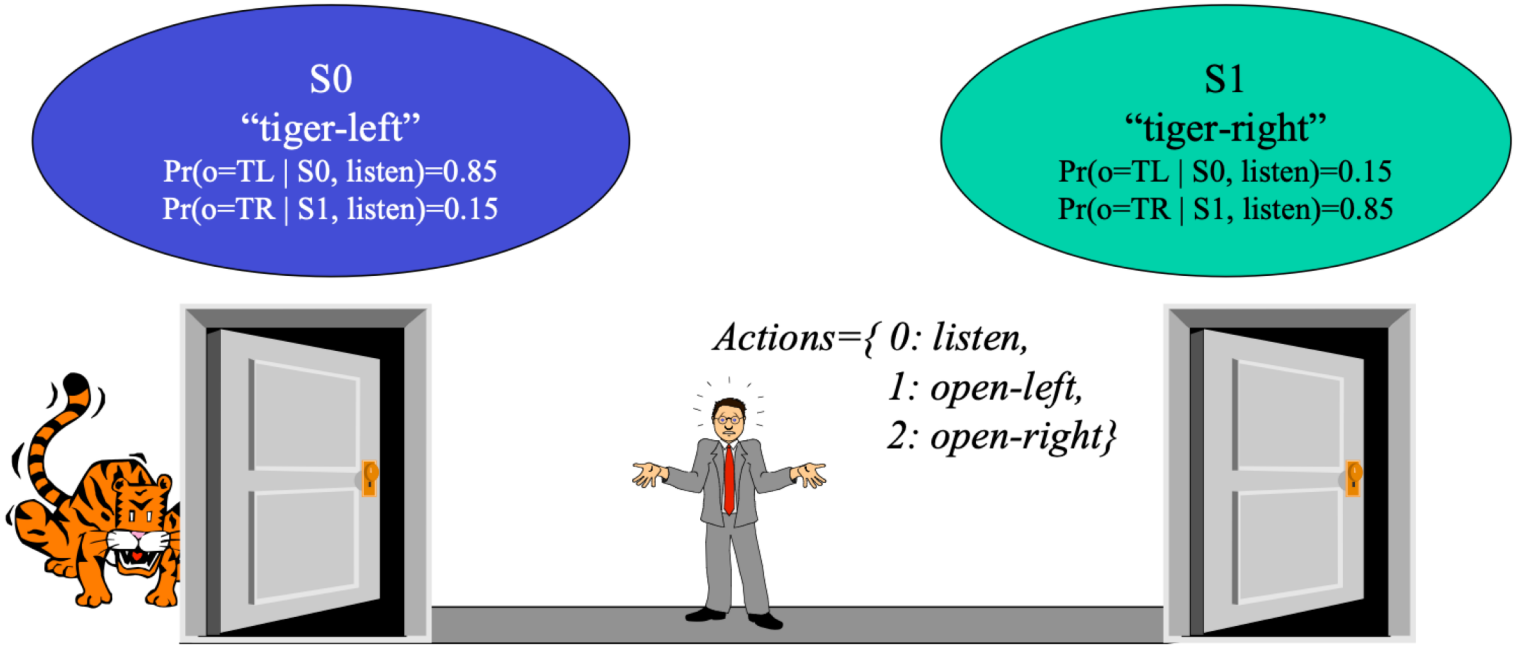
Move Up 5 times

0.005	0.007	0.019	0.775
0.034		0.007	0.105
0.005	0.006	0.008	0.030

(d)

Move Right 5 times

POMDP: Tiger example

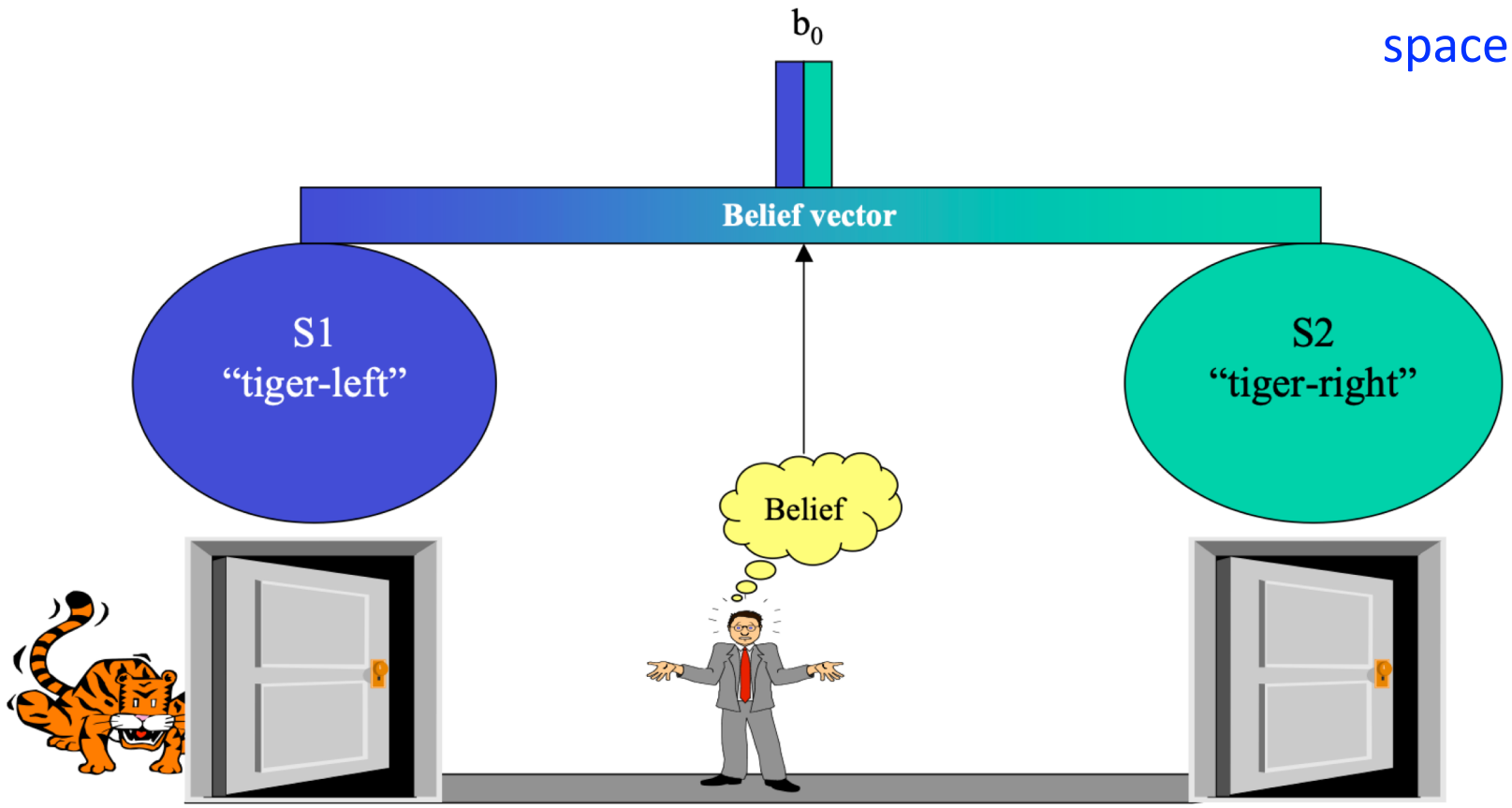


*Once a door is opened, the tiger is randomly placed behind a door, and the belief is reset (0.5,0.5).

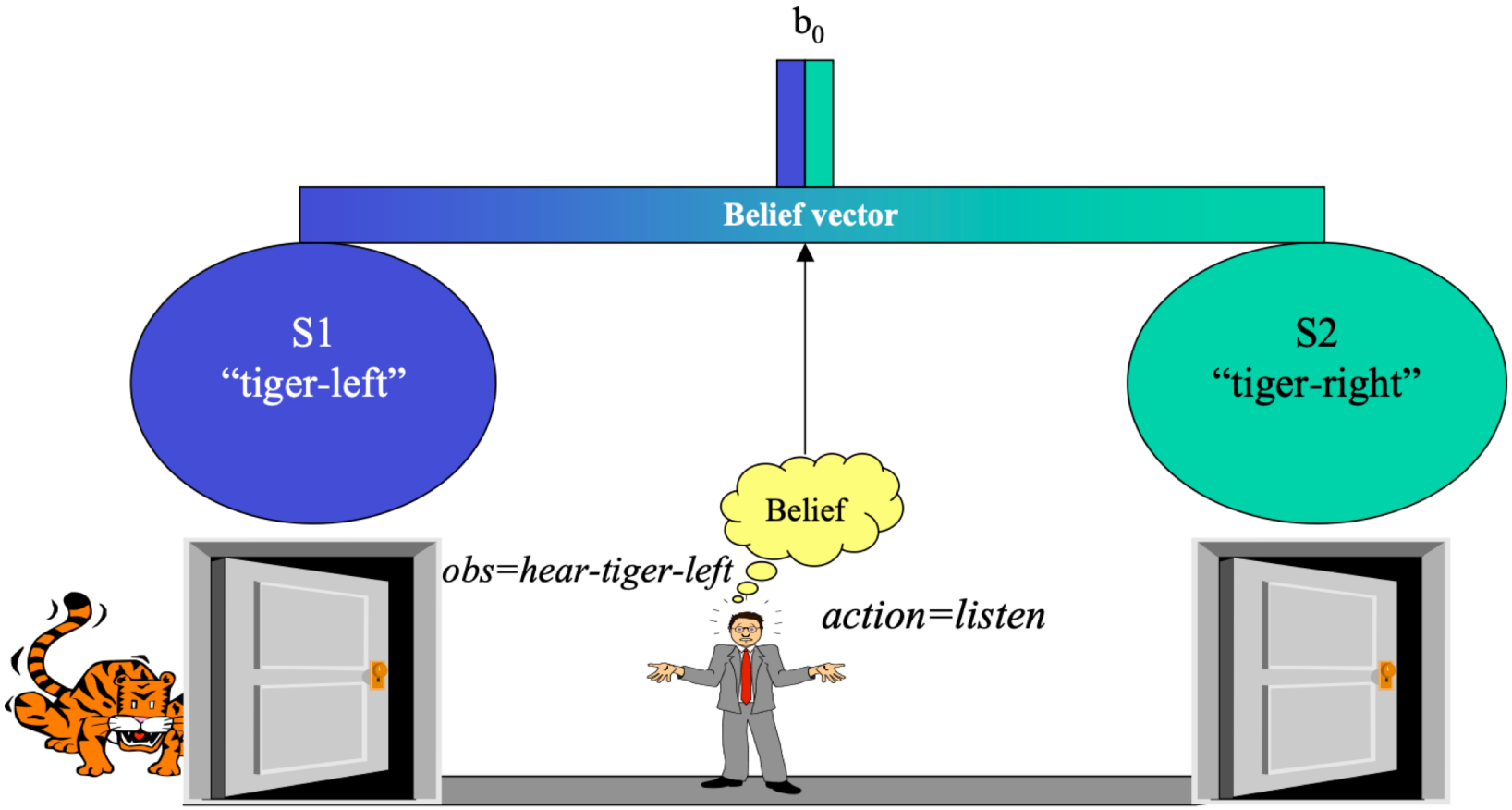
https://people.csail.mit.edu/lpk/papers/aij98-pomdp.pdf

The tiger problem: State tracking

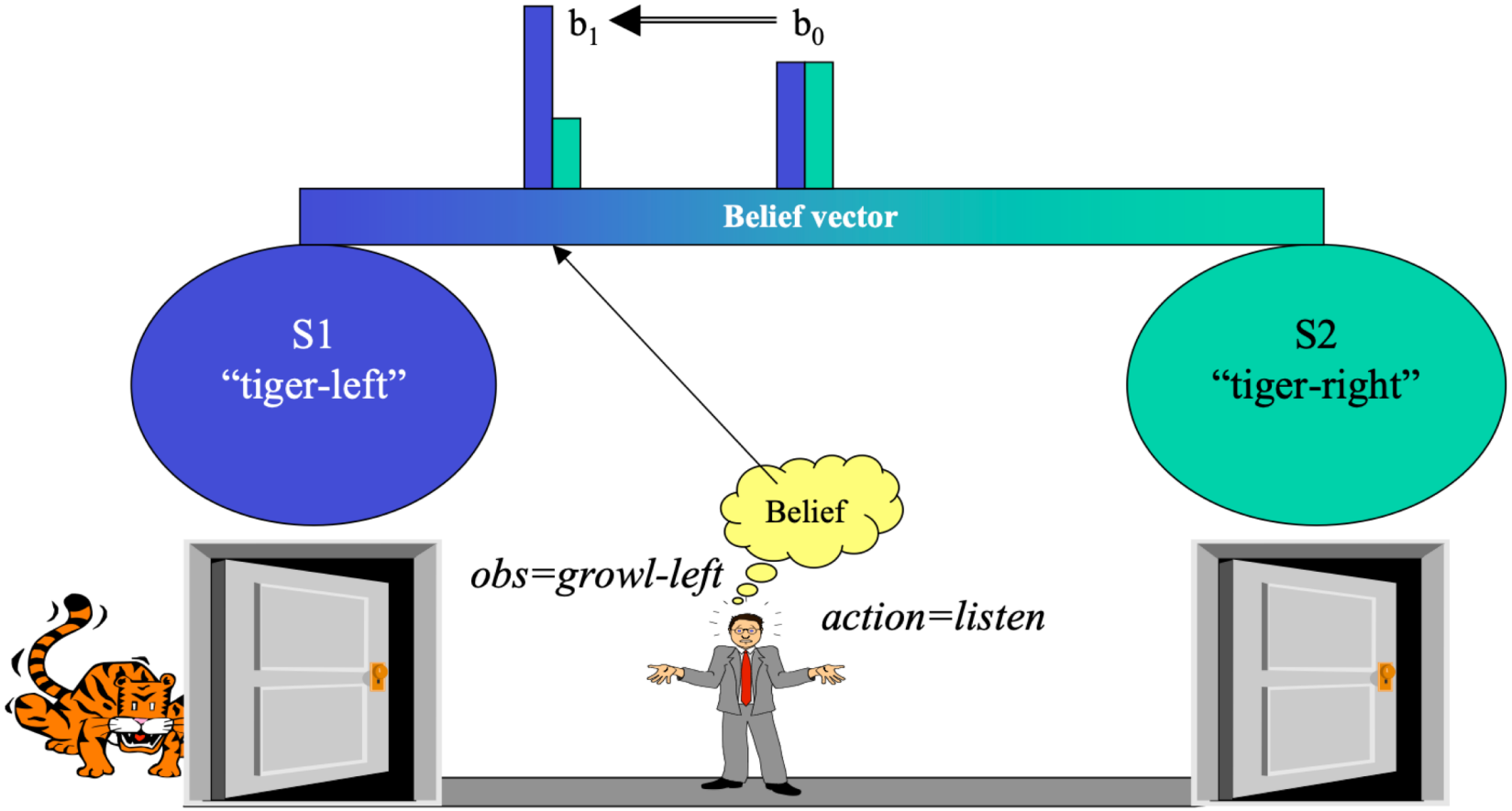
POMDP is a continuous space belief-MDP



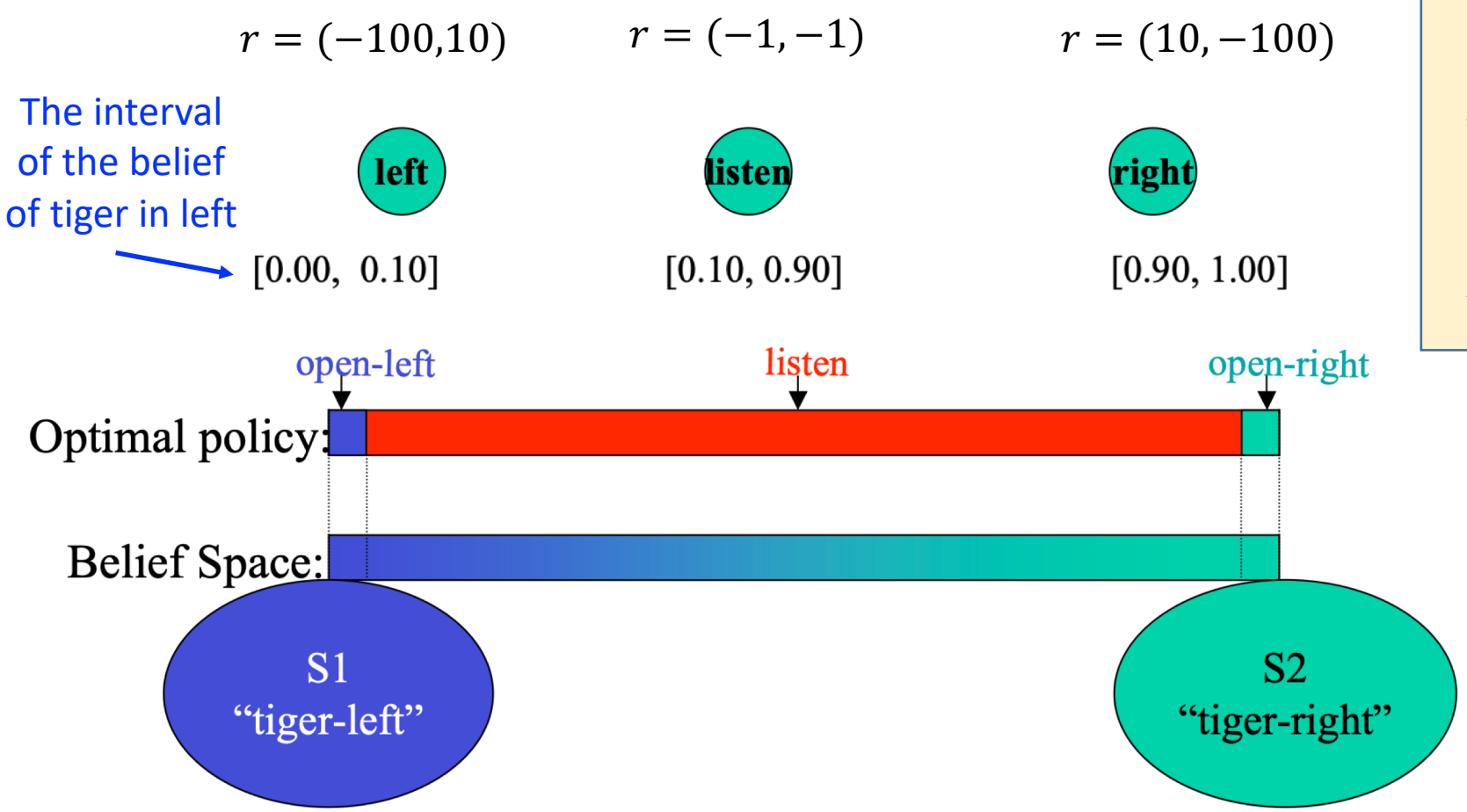
The tiger problem: State tracking



The tiger problem: State tracking



Tiger example – optimal policy for T=1



$$\boldsymbol{b} = (\mathbf{0.5}, \mathbf{0.5})$$

$$E^{Left/Right}[r] = 0.5(-100) + 0.5(10)$$

$$= -45$$

$$E^{Listen}[r] = 0.5(-1) + 0.5(-1) = -1$$

$$\boldsymbol{b} = (\mathbf{0.9}, \mathbf{0.1})$$

$$E^{Left}[r] = 0.9(-100) + 0.1(10)$$

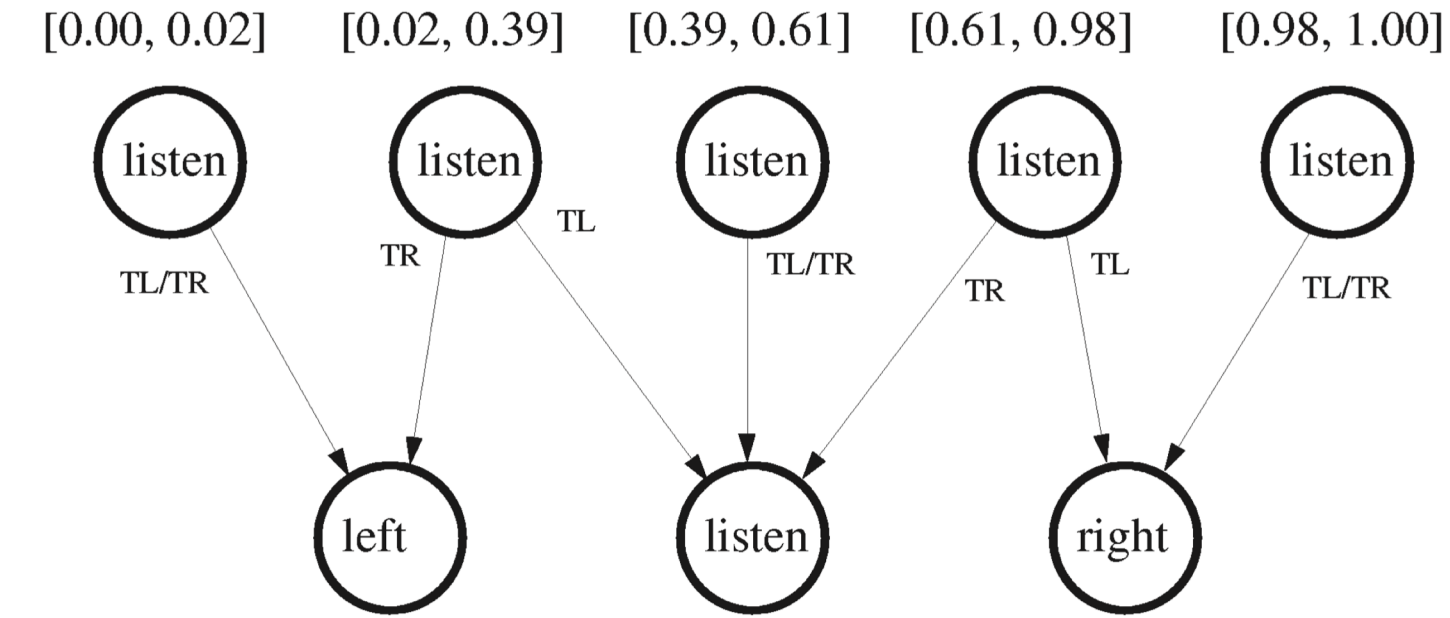
$$= -89$$

$$E^{Right}[r] = 0.9(10) + 0.1(-100)$$

$$= -1$$

Tiger example – optimal policy for T=2

The interval of the belief of tiger in left



- The structure gets richer
- If a door is selected at t=2, then the belief is reset to (0.5,0.5) resulting in listen action in t=1
- Instead listen at t=2, and act accordingly at t=1.

Belief states will change depending on the observations

How to solve POMDPs?

- Since the state is not observable, the agent has to make its decisions based on the belief state which is a posterior distribution over states.
  - $\pi: beliefs \rightarrow actions$
- Let b be the belief of the agent about the state under consideration.
  - b is a probability distribution and continuous.
- POMDPs compute a value function over belief space with horizon T:

$$V_T(b) = \max_u \left[r(b,u) + \gamma \int V_{T-1}(b') p(b'|u,b) db' \right]$$

- For finite worlds with finite state, action, and evidence spaces and finite horizons, however, we can effectively represent the value functions by piecewise linear functions.

Illustrative example

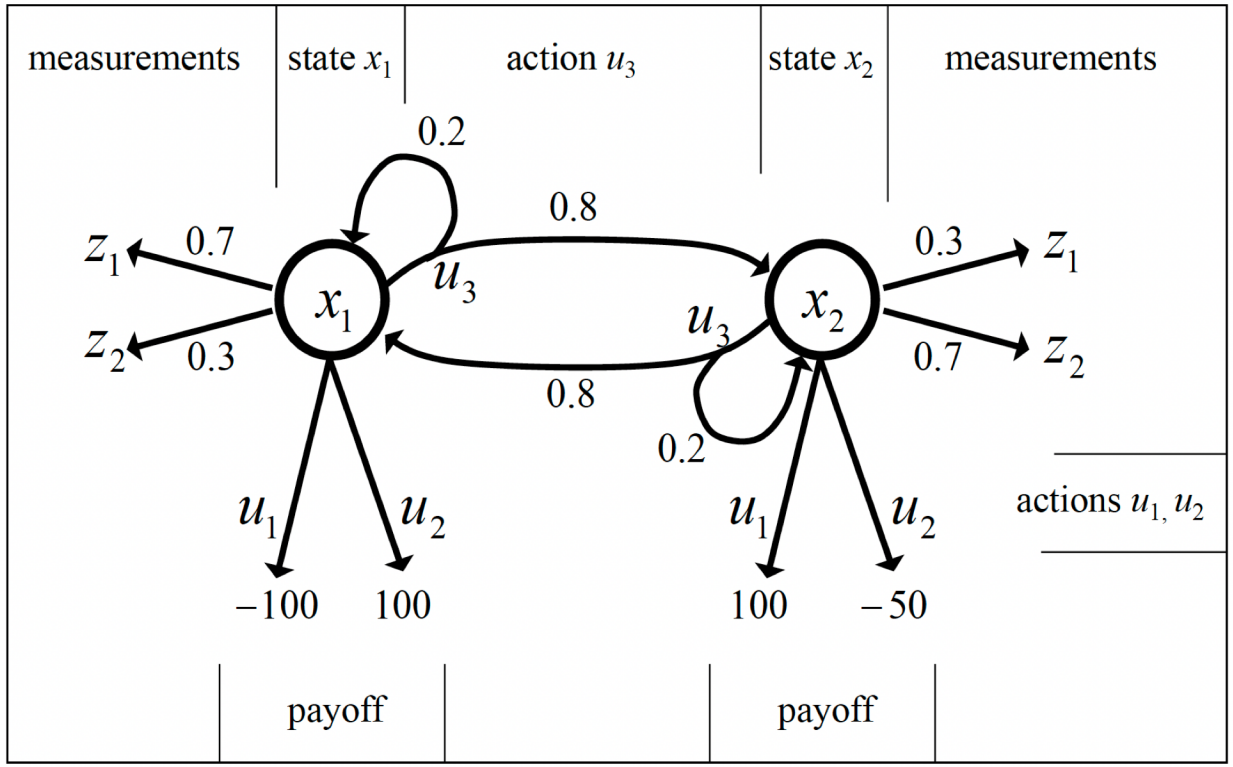
Suppose that the horizon is 1 and $\gamma = 1$.

The actions u_1 and u_2 are terminal actions.

$$
\begin{aligned}
r(x_1, u_1) &= -100 & \qquad r(x_2, u_1) &= +100 \\
r(x_1, u_2) &= +100 & \qquad r(x_2, u_2) &= -50 \\
r(x_1, u_3) &= -1 & \qquad r(x_2, u_3) &= -1
\end{aligned}
$$

$$
\begin{aligned}
p(x_1'|x_1, u_3) &= 0.2 & \qquad p(x_2'|x_1, u_3) &= 0.8 \\
p(x_1'|x_2, u_3) &= 0.8 & \qquad p(z_2'|x_2, u_3) &= 0.2
\end{aligned}
$$

$$
\begin{aligned}
p(z_1|x_1) &= 0.7 & \qquad p(z_2|x_1) &= 0.3 \\
p(z_1|x_2) &= 0.3 & \qquad p(z_2|x_2) &= 0.7
\end{aligned}
$$



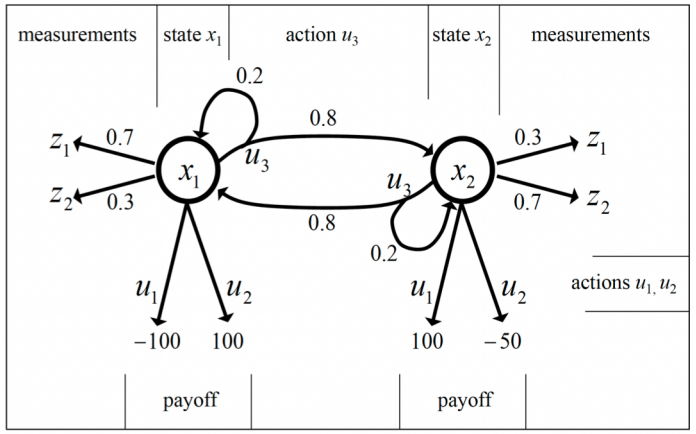
Reward in POMDPs

- In MDPs, the reward depends on the state of the system.
- In POMDPs, however, the true state is not exactly known.
- Therefore, we compute the expected reward by integrating over all states:

$$r(b,u) = E_x[r(x,u)]$$

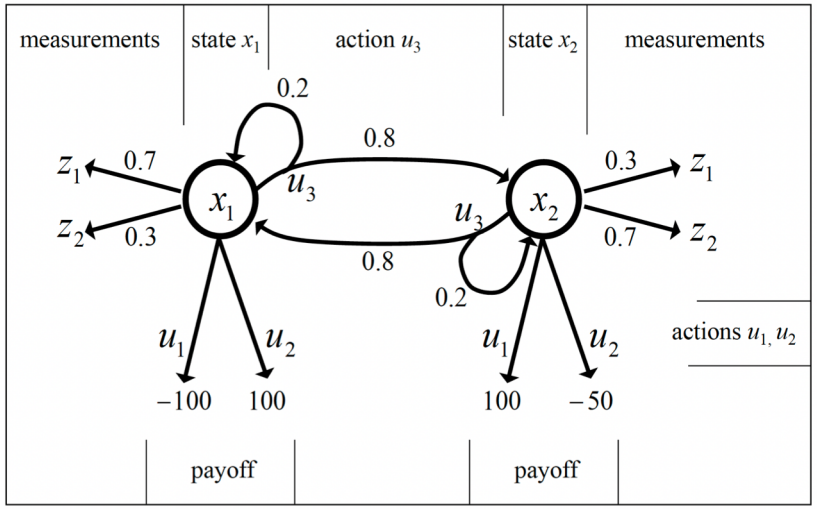
$$= \int r(x,u)p(x)dx$$

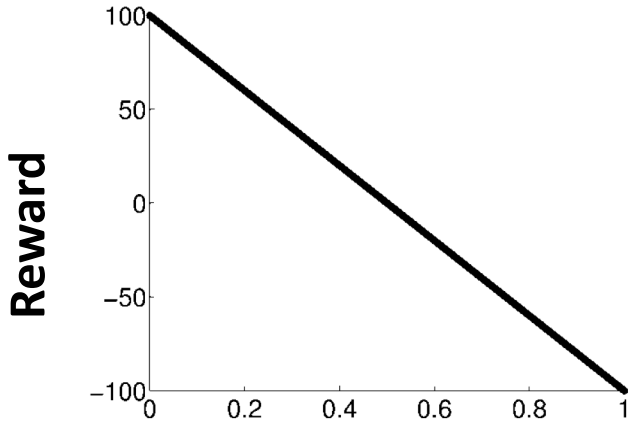
$$= p_1 r(x_1,u) + p_2 r(x_2,u)$$



Rewards in our example

- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100.
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by the probabilities.



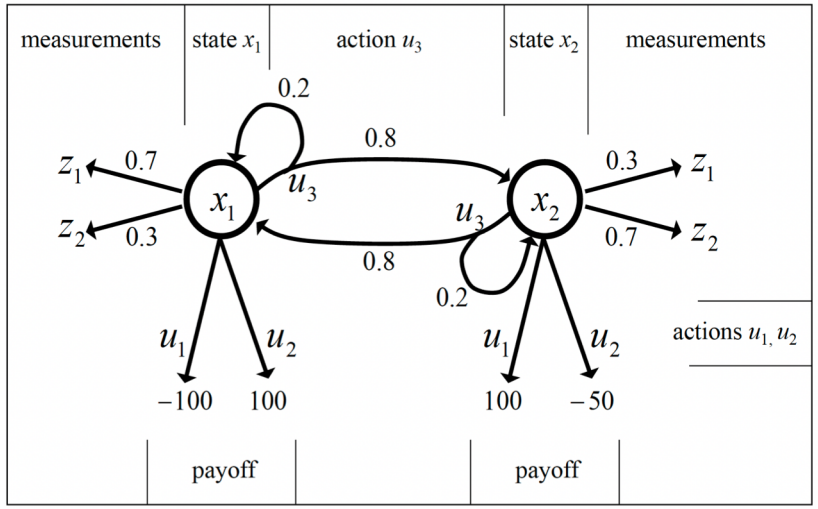


$P_1 = p(state = x_1)$

$$r(b, u_1) = p_1 r(x_1, u_1) + p_2 r(x_2, u_1)$$
$$= -100p_1 + 100p_2$$
$$= -100p_1 + 100(1 - p_1)$$
$$= 100 - 200p_1$$

Rewards in our example

- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100.
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by the probabilities.



$$r(b,u_1) = -100p_1 + 100p_2$$
$$= -100p_1 + 100(1-p_1)$$
$$= 100 - 200p_1$$

$$r(b,u_2) = 100p_1 - 50p_2$$
$$= 100p_1 - 50(1-p_1)$$
$$= 150p_1 - 50$$

$$r(b,u_3) = -1$$

Rewards in our example

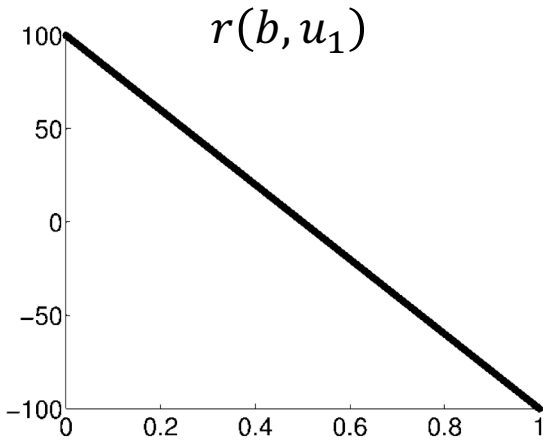
$r(b, u_1) = 100 - 200p_1$

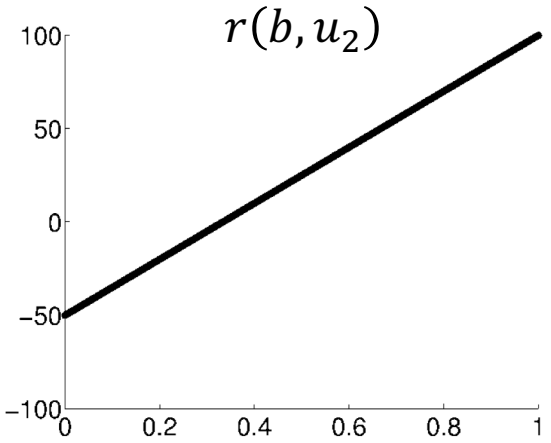
$r(b, u_2) = 150p_1 - 50$

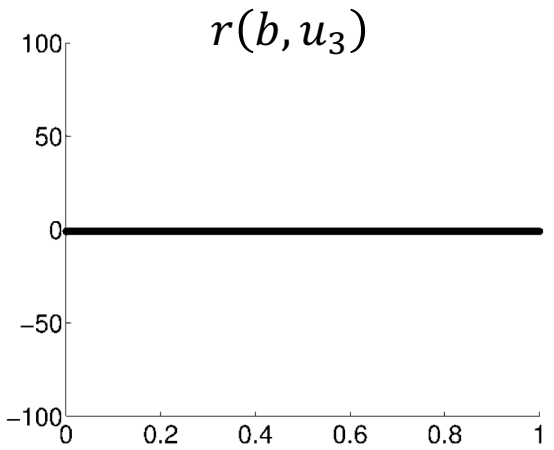
$r(b, u_3) = -1$

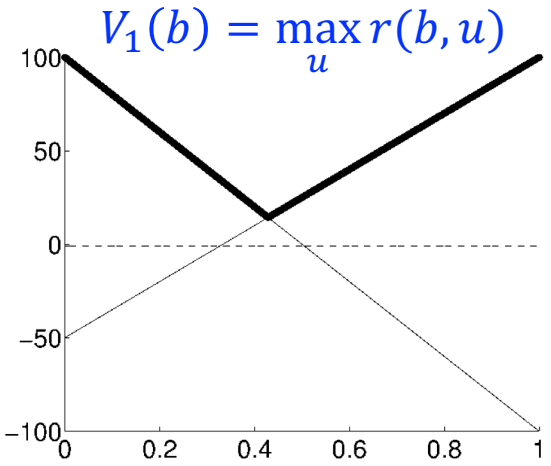
$$V_1(b) = \max_u r(b, u)$$
$$= \max \begin{Bmatrix} 100 - 200p_1 \\ 150p_1 - 50 \\ -1 \end{Bmatrix}$$

it's piecewise linear and convex





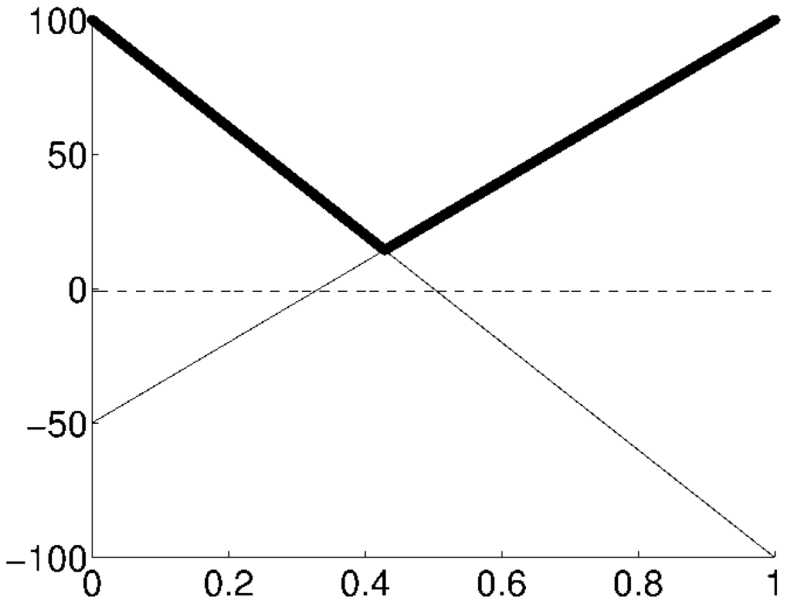




Resulting policy for horizon=1

- Given a finite POMDP with time horizon = 1, use $V_1(b)$ to determine the optimal policy.

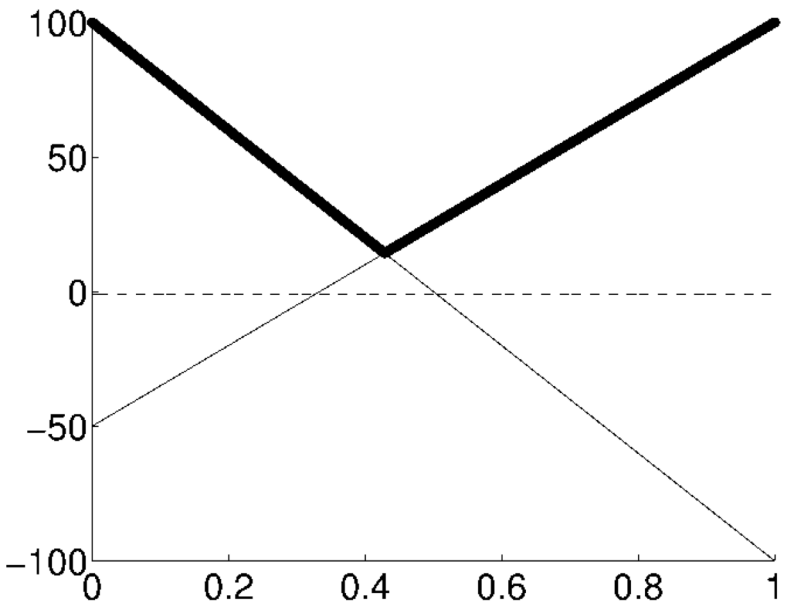
$$\pi_1(b) = \begin{cases} u_1 & if\ p_1 \leq \frac{3}{7} = 0.429 \\ u_2 & if\ p_1 > \frac{3}{7} \end{cases}$$



Pruning

$$V_1(b) = \max\begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 100p_1 - 50(1-p_1) \\ -1 \end{Bmatrix}$$

$$= \max\begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 100p_1 - 50(1-p_1) \end{Bmatrix}$$

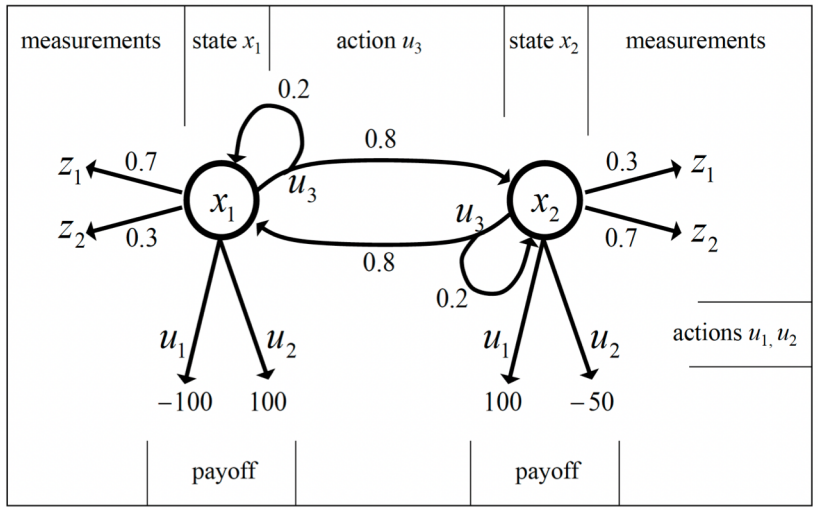


- The first two parts contribute,
- The third element can be pruned.

$$V_1(b) = \max\begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 100p_1 - 50(1-p_1) \end{Bmatrix}$$

Incorporating observations

- $p(z_1|x_1) = 0.7$
- $p(z_1|x_2) = 0.3$
- Given z_1, we update the belief using Bayes rule.



$$p(x_1|z_1) = p_1' = \frac{p(z_1|x_1)p(x_1)}{p(z_1)} = \frac{0.7p_1}{p(z_1)} \qquad \text{where}$$

$$p(z_1) = 0.7p_1 + 0.3(1-p_1) = 0.4p_1 + 0.3$$

$$V_1(b|z_1) = \max\begin{Bmatrix} -100\dfrac{0.7p_1}{p(z_1)} + 100\dfrac{0.3(1-p_1)}{p(z_1)} \\ 100\dfrac{0.7p_1}{p(z_1)} - 50\dfrac{0.3(1-p_1)}{p(z_1)} \end{Bmatrix} = \frac{1}{p(z_1)}\max\begin{Bmatrix} -70p_1 + 30(1-p_1) \\ 70p_1 - 15(1-p_1) \end{Bmatrix}$$

Expected value after measuring

- But, we do not know in advance what the next measurement will be, so we must compute the expected belief:

$$\bar{V}_1(b) = E_z[V_1(b|z)] = \sum_{i=1}^{2} p(z_i) V_1(b|z_i)$$

$$V_1(b|z_1) = \frac{1}{p(z_1)} \max \begin{Bmatrix} -70p_1 + 30(1-p_1) \\ 70p_1 - 15(1-p_1) \end{Bmatrix}$$

$$= p(z_1) V_1(b|z_1) + p(z_2) V_1(b|z_2)$$

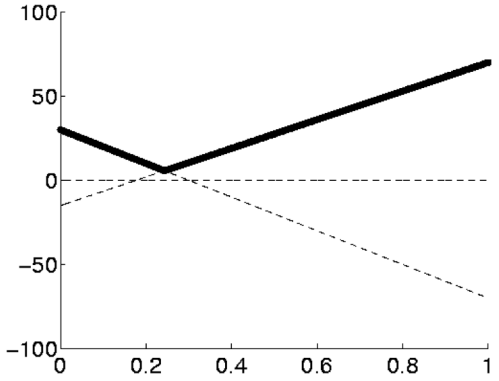
$$= \max \begin{Bmatrix} -70p_1 + 30(1-p_1) \\ 70p_1 - 15(1-p_1) \end{Bmatrix} + \max \begin{Bmatrix} -30p_1 + 70(1-p_1) \\ 30p_1 - 35(1-p_1) \end{Bmatrix}$$

Resulting value function

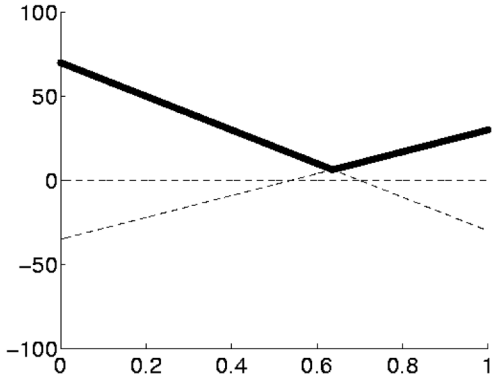
$$\bar{V}_1(b) = \max\begin{Bmatrix} -70p_1 + 30(1-p_1) \\ 70p_1 - 15(1-p_1) \end{Bmatrix} + \max\begin{Bmatrix} -30p_1 + 70(1-p_1) \\ 30p_1 - 35(1-p_1) \end{Bmatrix}$$

$$= \max\begin{Bmatrix} -70p_1 + 30(1-p_1) - 30p_1 + 70(1-p_1) \\ -70p_1 + 30(1-p_1) + 30p_1 - 35(1-p_1) \\ 70p_1 - 15(1-p_1) - 30p_1 + 70(1-p_1) \\ 70p_1 - 15(1-p_1) + 30p_1 - 35(1-p_1) \end{Bmatrix}$$

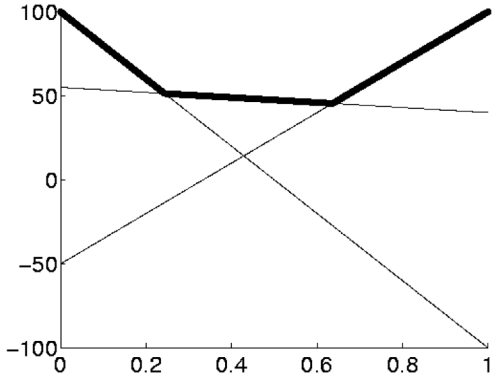
$$= \max\begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 40p_1 + 55(1-p_1) \\ 100p_1 - 50(1-p_1) \end{Bmatrix}$$



$p(z_1)V_1(b|z_1)$

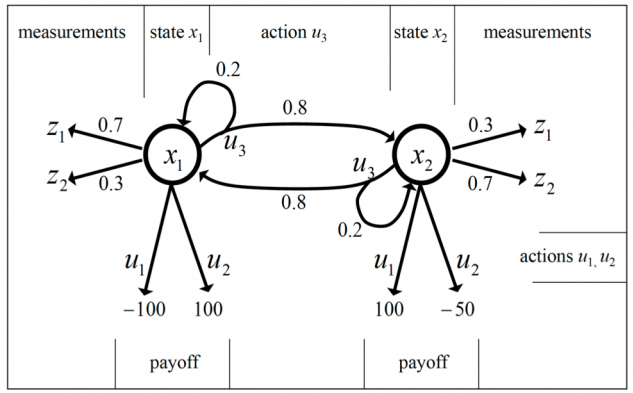


$p(z_2)V_1(b|z_2)$



$\bar{V}_1(b)$

Increasing the time horizon



- When the agent selects u_3, its state may change.
- When computing the value function, these potential state changes should be considered.

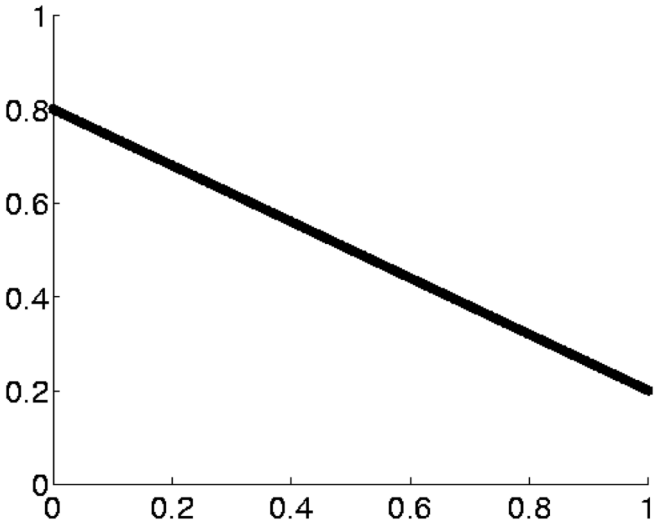
$$p_1' = E_x[p(x_1|x, u_3)]$$

$$= \sum_{i=1}^{2} p(x_1|x_i, u_3)p_i$$

$$= 0.2p_1 + 0.8(1 - p_1)$$

$$= 0.8 - 0.6p_1$$

P($x = x_1$ after executing u_3)



P($x = x_1$ originally)

Resulting value function after executing $\boldsymbol{u_3}$

$$p_1' = 0.8 - 0.6p_1$$

Probability of being at x_1 after executing u_3

$$\bar{V}_1(b) = \max \begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 40p_1 + 55(1-p_1) \\ 100p_1 - 50(1-p_1) \end{Bmatrix}$$

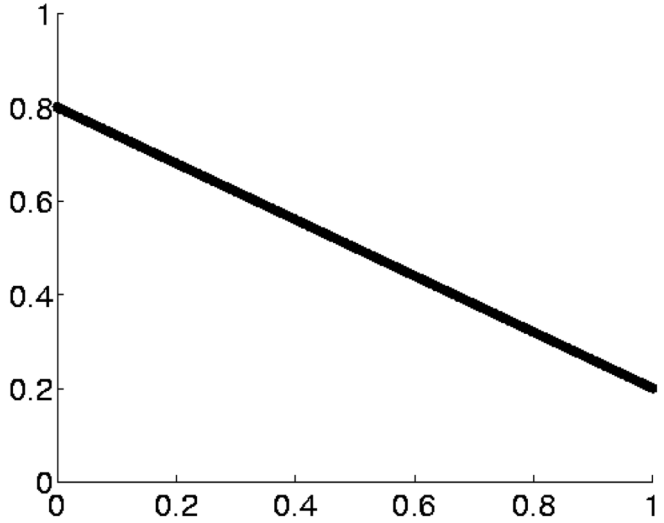
Expected value after measurement

$$\bar{V}_1(b|\mathrm{u}_3) = \max \begin{Bmatrix} 60p_1 - 60(1-p_1) \\ 52p_1 + 43(1-p_1) \\ -20p_1 + 70(1-p_1) \end{Bmatrix}$$

Takes into account the state transitions

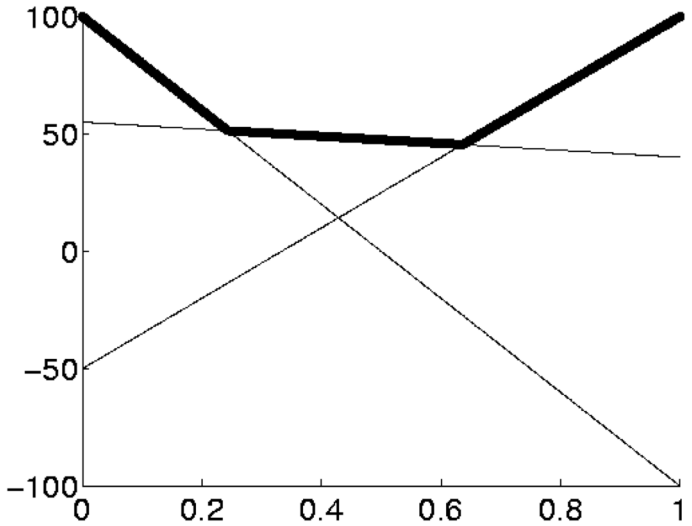
Value function after executing $\boldsymbol{u_3}$

P($x = x_1$ after executing u_3)

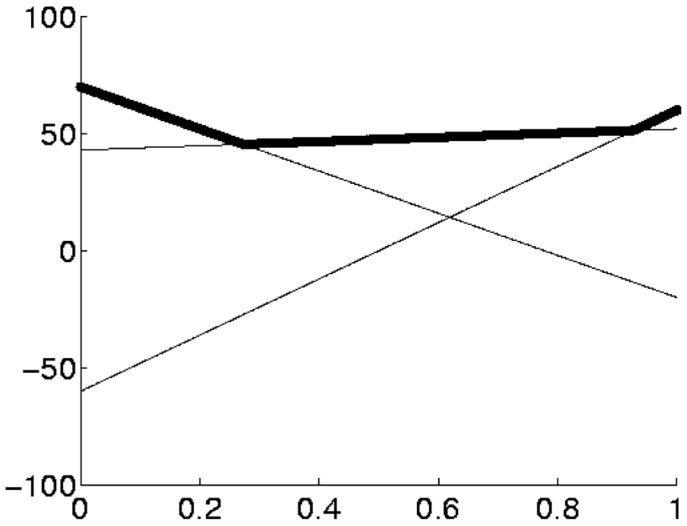


P($x = x_1$ originally)

$\bar{V}_1(b)$



$\bar{V}_1(b|u_3)$

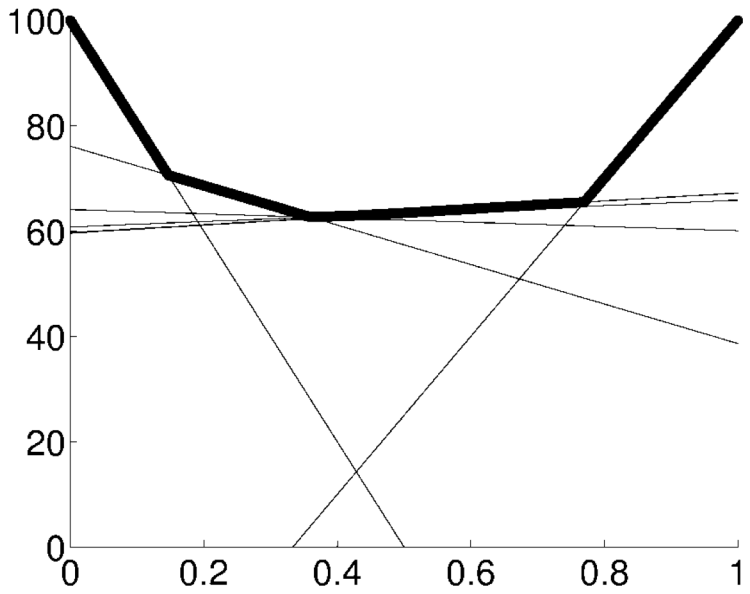


$$\bar{V}_1(b) = \max\begin{Bmatrix} -100p_1 + 100(1-p_1) \\ 40p_1 + 55(1-p_1) \\ 100p_1 - 50(1-p_1) \end{Bmatrix}$$

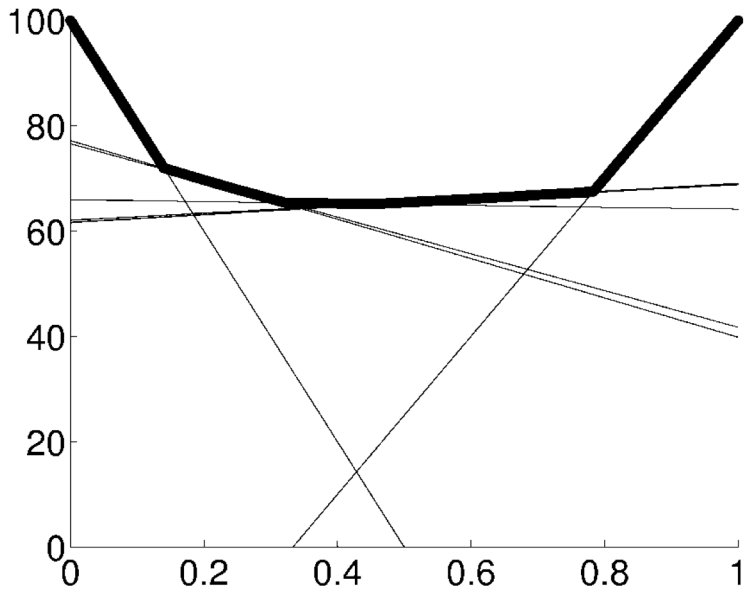
$$\bar{V}_1(b|\mathrm{u}_3) = \max\begin{Bmatrix} 60p_1 - 60(1-p_1) \\ 52p_1 + 43(1-p_1) \\ -20p_1 + 70(1-p_1) \end{Bmatrix}$$

For longer horizons

- We know how to update the value of a belief under action execution and measurements.



T=10

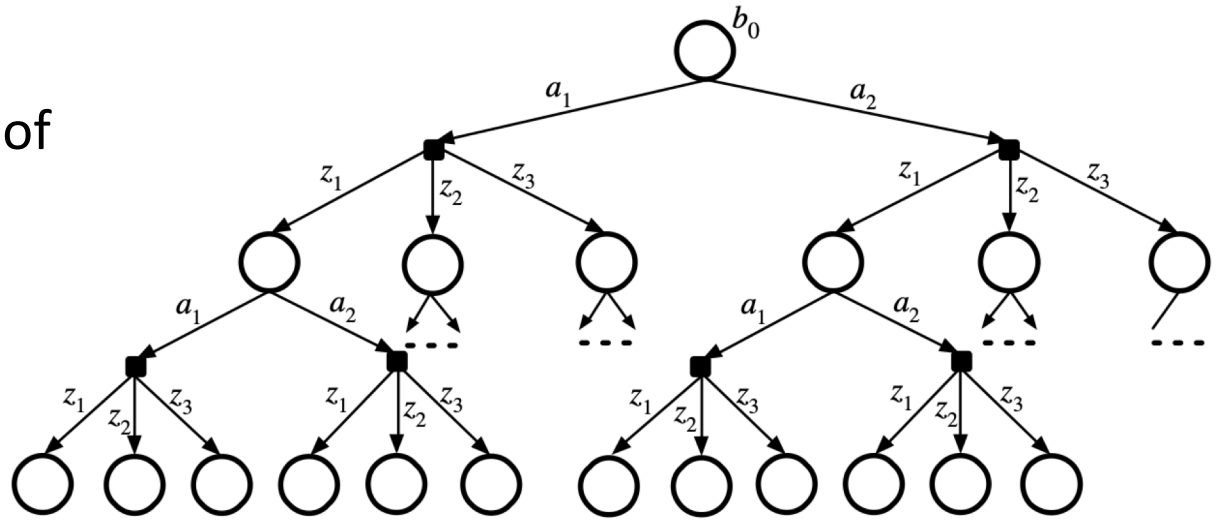


T=20

Pruning is crucial!

Solving POMDP for longer horizons and larger problems

- Branching over actions and observations
- Curse of dimensionality: complexity of planning grows exponentially with the size of state space and the planning horizon
- Approximate solutions
  - Search over sequences of actions with limited look-ahead horizon
  - Monte Carlo tree search



Summary

- **MDP:** States fully observable
- **POMDP:** States partially observable
  - POMDP state: **belief** (probability distribution over the states)
- Exact solutions for POMDPs are not possible!
  - Approximate solutions: finite look-ahead horizon, sampling some scenarios
- Applications:
  - Planning
    - The states are the robots location, surroundings and internal state,
    - the actions are the available actuators,
    - the observations are the outputs of the sensing equipment, and
    - the immediate reward function encodes the robot's general goals.
  - Estimation