CSE-571
Robotics

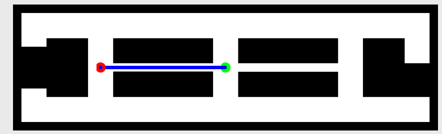
Planning and Control:

Markov Decision Processes

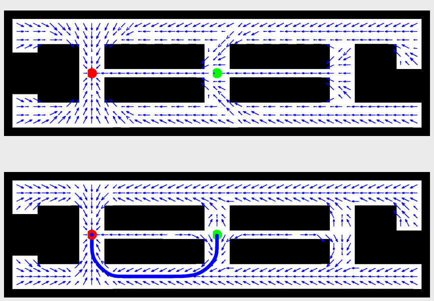
Problem Classes

- Deterministic vs. stochastic actions
- Full vs. partial observability

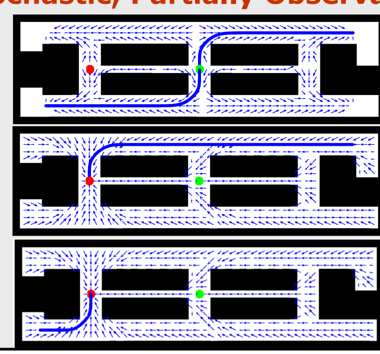
Deterministic, fully observable



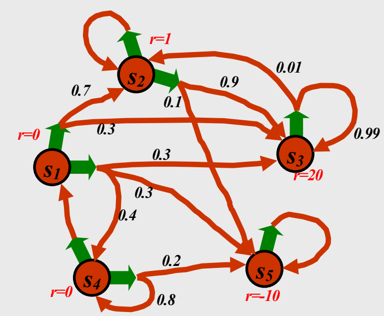
Stochastic, Fully Observable



Stochastic, Partially Observable



Markov Decision Process (MDP)



Markov Decision Process (MDP)

- **Given:**
- States x
- Actions u
- Transition probabilities $p(x'|u,x)$
- Reward / payoff function $r(x,u)$

- **Wanted:**
- Policy $\pi(x)$ that maximizes the future expected reward

Rewards and Policies

- Policy (general case):

$$\pi: \quad z_{1:t-1}, u_{1:t-1} \rightarrow \quad u_t$$

- Policy (fully observable case):

$$\pi: \quad x_t \rightarrow \quad u_t$$

- Expected cumulative payoff:

$$R_T = E\left[\sum_{\tau=1}^{T} \gamma^\tau r_{t+\tau}\right]$$

  - T=1: greedy policy
  - T>1: finite horizon case, typically no discount
  - T=infty: infinite-horizon case, finite reward if discount < 1

Policies contd.

- Expected cumulative payoff of policy:

$$R_T^\pi(x_t) = E\left[\sum_{\tau=1}^{T} \gamma^\tau r_{t+\tau} \mid u_{t+\tau} = \pi\left(z_{1:t+\tau-1} u_{1:t+\tau-1}\right)\right]$$

- Optimal policy:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \quad R_T^\pi(x_t)$$

- 1-step optimal policy:

$$\pi_1(x) = \underset{u}{\operatorname{argmax}} \quad r(x,u)$$

- Value function of 1-step optimal policy:

$$V_1(x) = \gamma \max_u \, r(x,u)$$

2-step Policies

- Optimal policy:

$$\pi_2(x) = \underset{u}{\operatorname{argmax}} \quad \left[r(x,u) + \int V_1(x') p(x' \mid u,x) dx'\right]$$

- Value function:

$$V_2(x) = \gamma \max_u \quad \left[r(x,u) + \int V_1(x') p(x' \mid u,x) dx'\right]$$

T-step Policies

- Optimal policy:

$$\pi_T(x) = \underset{u}{\operatorname{argmax}} \quad \left[r(x,u) + \int V_{T-1}(x') p(x' \mid u,x) dx'\right]$$

- Value function:

$$V_T(x) = \gamma \max_u \quad \left[r(x,u) + \int V_{T-1}(x') p(x' \mid u,x) dx'\right]$$

Infinite Horizon

- Optimal policy:

$$V_\infty(x) = \gamma \max_u \quad \left[r(x,u) + \int V_\infty(x') p(x' \mid u,x) dx'\right]$$

- Bellman equation

- Fix point is optimal policy

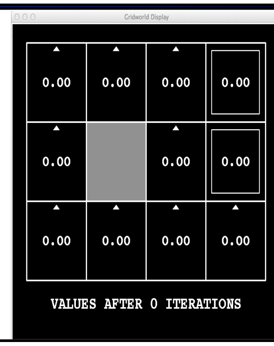
- Necessary and sufficient condition

Value Iteration

- for all x do
  $$\hat{V}(x) \leftarrow r_{\min}$$
- endfor
- repeat until convergence
  - for all x do
    $$\hat{V}(x) \leftarrow \gamma \max_u \left[r(x,u) + \int \hat{V}(x') p(x' \mid u,x)\, dx' \right]$$
  - endfor
- endrepeat

$$\pi(x) = \operatorname*{argmax}_u \left[r(x,u) + \int \hat{V}(x') p(x' \mid u,x)\, dx' \right]$$

k=0



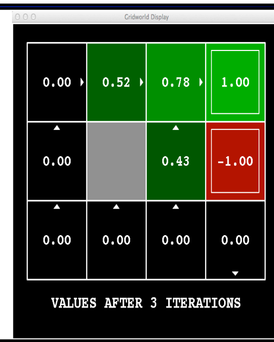
Noise = 0.2
Discount = 0.9
Living reward = 0

k=1

Noise = 0.2
Discount = 0.9
Living reward = 0

k=2

k=3

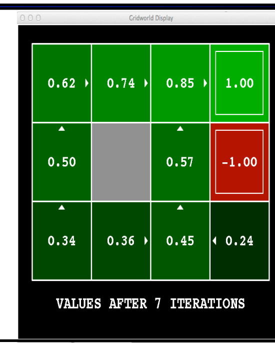


k=4

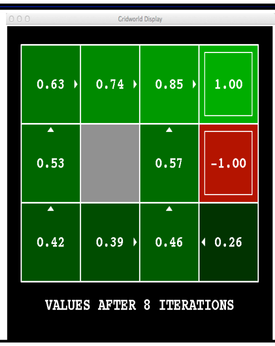
k=5

k=6

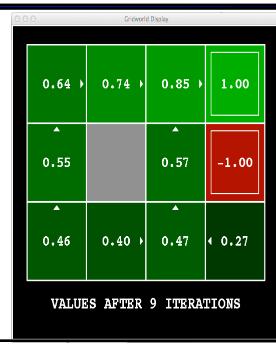
k=7



k=8

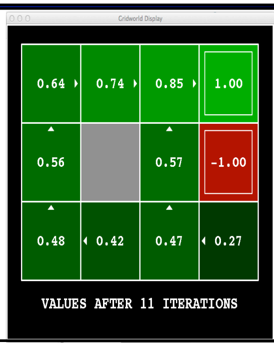


k=9

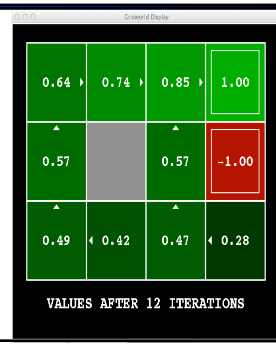


k=10

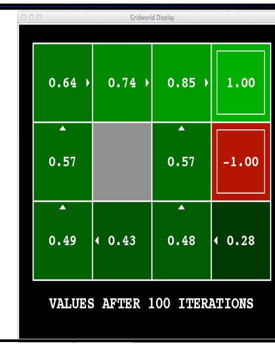
k=11



k=12

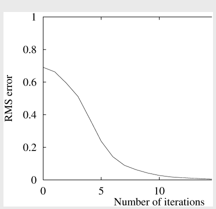


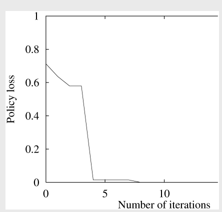
k=100



Value Function and Policy

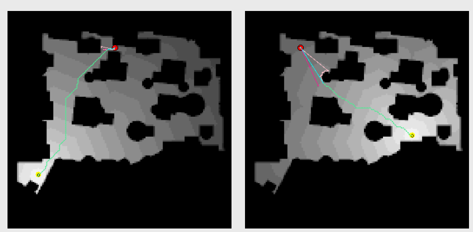
- Each step takes O(|A| |S| |S|) time.
- Number of iterations required is polynomial in |S|, |A|, 1/(1-gamma)





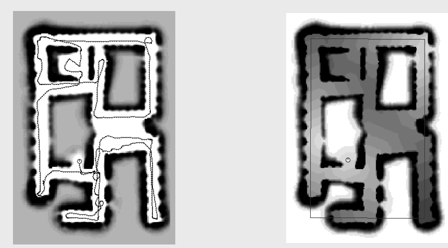
Value Iteration for Motion Planning

(assumes knowledge of robot's location)

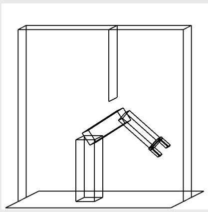


Frontier-based Exploration

- Every unknown location is a target point.



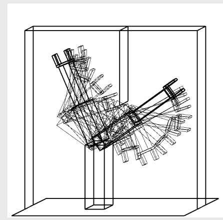
Manipulator Control



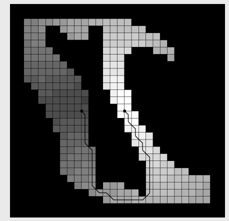
Arm with two joints

Configuration space

Manipulator Control Path

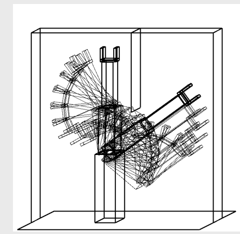


State space

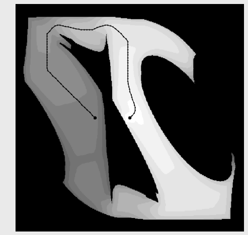


Configuration space

Manipulator Control Path



State space



Configuration space

POMDPs

- In POMDPs we apply the very same idea as in MDPs.
- 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.
- For finite horizon problems, the resulting value functions are piecewise linear and convex.
- In each iteration the **number of linear constraints grows exponentially**.
- Full fledged POMDPs have only been applied to very small state spaces with small numbers of possible observations and actions.
- **Approximate solutions are becoming more and more capable**.