CSE-481
Mobile Robotics Capstone

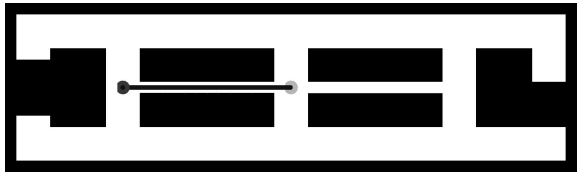
Planning and Control:

Markov Decision Processes

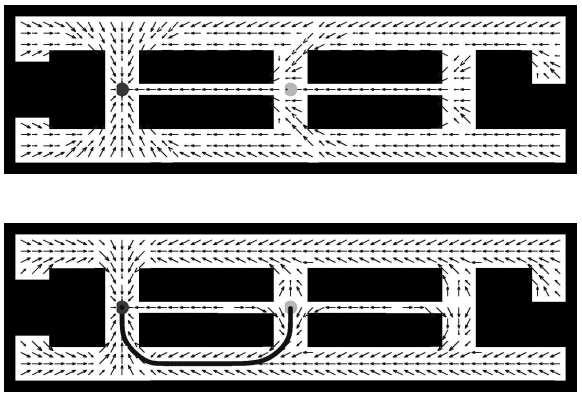
Problem Classes

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

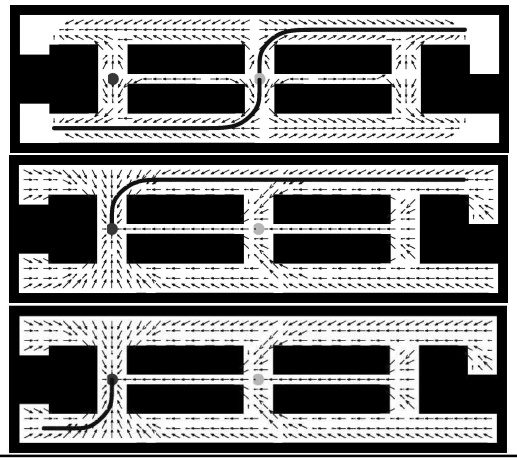
Deterministic, fully observable



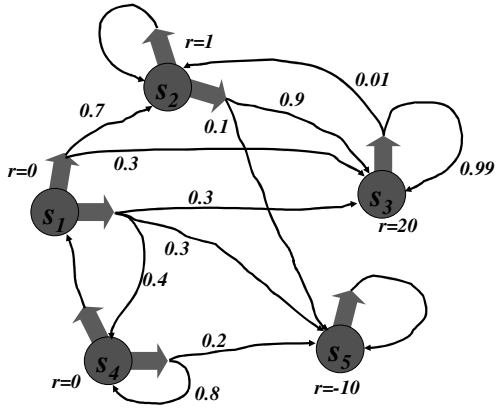
Stochastic, Fully Observable



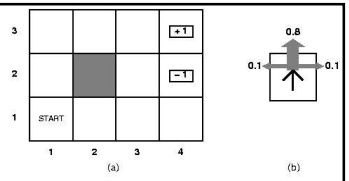
Stochastic, Partially Observable



Markov Decision Process (MDP)



Markov Decision Process Example



- Robot must reach state (4, 3) (reward of +1) and avoid state (4, 1) (punishment of –1). Actions are North, South, West, East.
- **Deterministic version:** All actions always lead to the next square in the selected direction, except that moving into a wall results in no change in position.
- **Stochastic version:** Each action achieves the intended effect with probability 0.8, but the rest of the time, the agent moves at right angles to the intended direction.

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):

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

- Policy (fully observable case):

$$\pi: \; x_t \rightarrow u_t$$

- 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:

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

- Optimal policy:

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

- 1-step optimal policy:

$$\pi_1(x) = \operatorname*{argmax}_{u} \; r(x,u)$$

- Value function of 1-step optimal policy:

$$V_1(x) = \gamma \max_{u} r(x,u)$$

2-step Policies

- Optimal policy:

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

- Value function:

$$V_2(x) = \gamma \max_{u} \left[r(x,u) + \int V_1(x') p(x' \,|\, u, x) dx'\right]$$

T-step Policies

- Optimal policy:

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

- Value function:

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

Infinite Horizon

- Optimal policy:

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

- 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

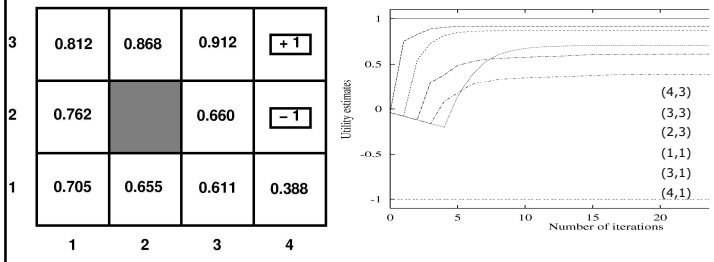
$$\hat{V}(x) \leftarrow \gamma \max_u \left[r(x,u) + \int \hat{V}(x') p(x'|u,x) dx' \right]$$

  - endfor
- endrepeat

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

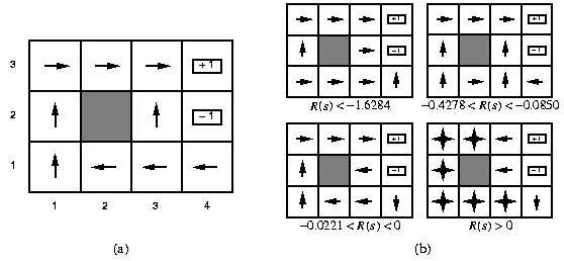
MDP Example

- Value function for R(s) = -0.04



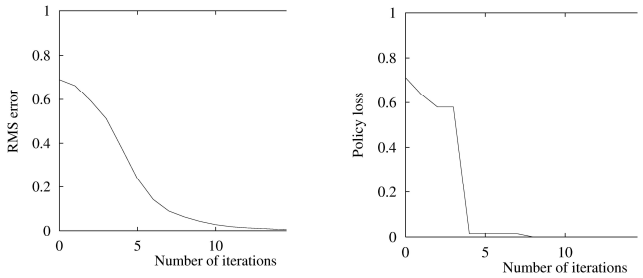
MDP Example

- Optimal policies



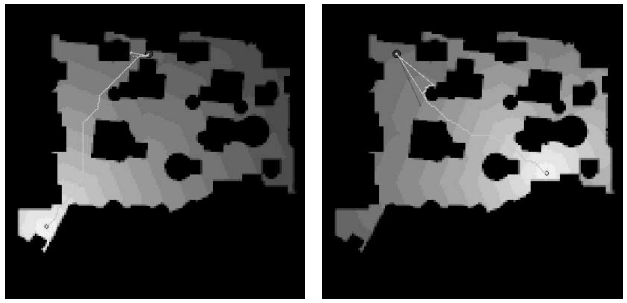
Value Function and Policy

- Value function error and policy error



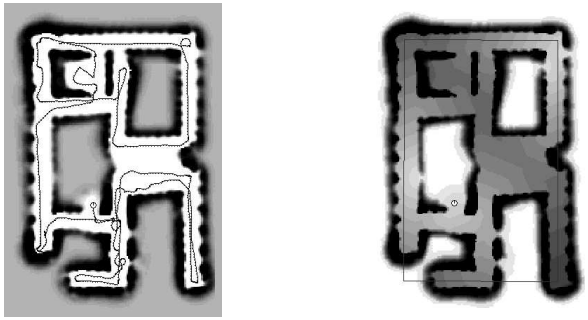
Value Iteration for Motion Planning

(assumes knowledge of robot's location)

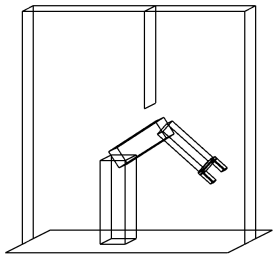


Frontier-based Exploration

- Every unknown location is a target point.



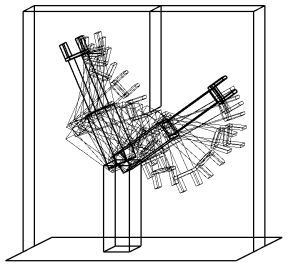
Manipulator Control



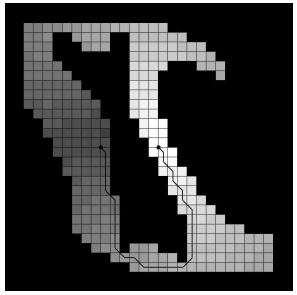
Arm with two joints

Configuration space

Manipulator Control Path

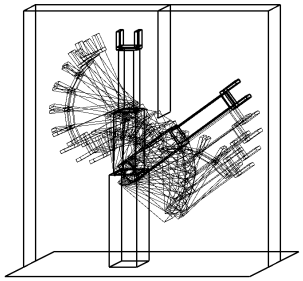


State space

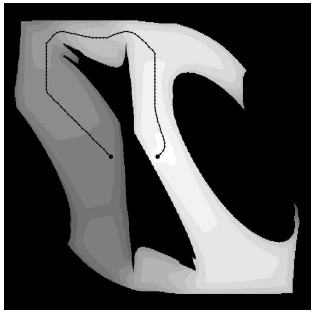


Configuration space

Manipulator Control Path



State space



Configuration space