

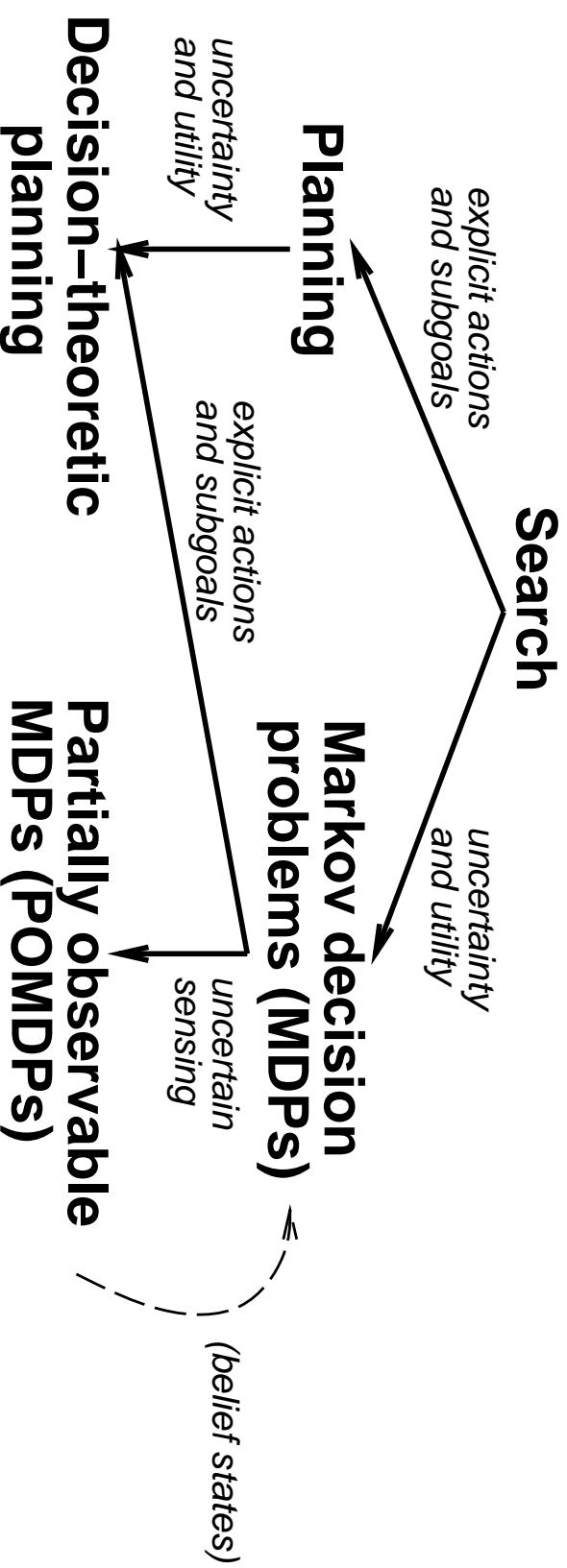
# Complex decisions

## CHAPTER 17, SECTIONS 1–3

# Outline

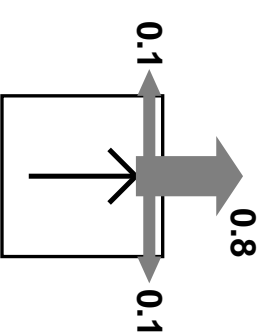
- ◇ Decision problems
- ◇ Value iteration
- ◇ Policy iteration

# Sequential decision problems



# Example MDP

3				<span style="border: 1px solid black; padding: 2px;">+1</span>
2				<span style="border: 1px solid black; padding: 2px;">-1</span>
1	START			
	1	2	3	4



Model  $M_{ij}^a \equiv P(j|i, a)$  = probability that doing  $a$  in  $i$  leads to  $j$

Each state has a *reward*  $R(i)$

= -0.04 (small penalty) for nonterminal states

=  $\pm 1$  for terminal states

# Solving MDPs

In search problems, aim is to find an optimal *sequence*

In MDPs, aim is to find an optimal *policy*  
i.e., best action for every possible state  
(because can't predict where one will end up)

Optimal policy and state values for the given  $R(i)$ :

3	→	→	→	<span style="border: 1px solid black; padding: 2px;">+1</span>
2	↑		↑	<span style="border: 1px solid black; padding: 2px;">-1</span>
1	↑	→	→	→
	1	2	3	4

3	0.812	0.868	0.912	<span style="border: 1px solid black; padding: 2px;">+1</span>
2	0.762		0.660	<span style="border: 1px solid black; padding: 2px;">-1</span>
1	0.705	0.655	0.611	0.388
	1	2	3	4

# Utility

In *sequential* decision problems, preferences are expressed between *sequences* of states

Usually use an *additive* utility function:

$U([s_1, s_2, s_3, \dots, s_n]) = R(s_1) + R(s_2) + R(s_3) + \dots + R(s_n)$   
(cf. path cost in search problems)

Utility of a *state* (a.k.a. its *value*) is defined to be

$U(s_i) =$  expected sum of rewards until termination  
assuming optimal actions

Given the utilities of the states, choosing the best action is just MEU: choose the action such that the expected utility of the immediate successors is highest.

# Bellman equation

Definition of utility of states leads to a simple relationship among utilities of neighboring states:

$$\begin{aligned} & \text{expected sum of rewards} \\ &= \text{current reward} \\ &+ \text{expected sum of rewards after taking best action} \end{aligned}$$

Bellman equation (1957):

$$U(i) = R(i) + \max_a \sum_j U(j) M_{ij}^a$$

$$\begin{aligned} U(1,1) = & -0.04 \\ & + \max \{ 0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1), \\ & \quad 0.9U(1,1) + 0.1U(1,2) \\ & \quad 0.9U(1,1) + 0.1U(2,1) \\ & \quad 0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1) \} \end{aligned}$$

*up*
*left*
*down*
*right*

One equation per state =  $n$  nonlinear equations in  $n$  unknowns

# Value iteration algorithm

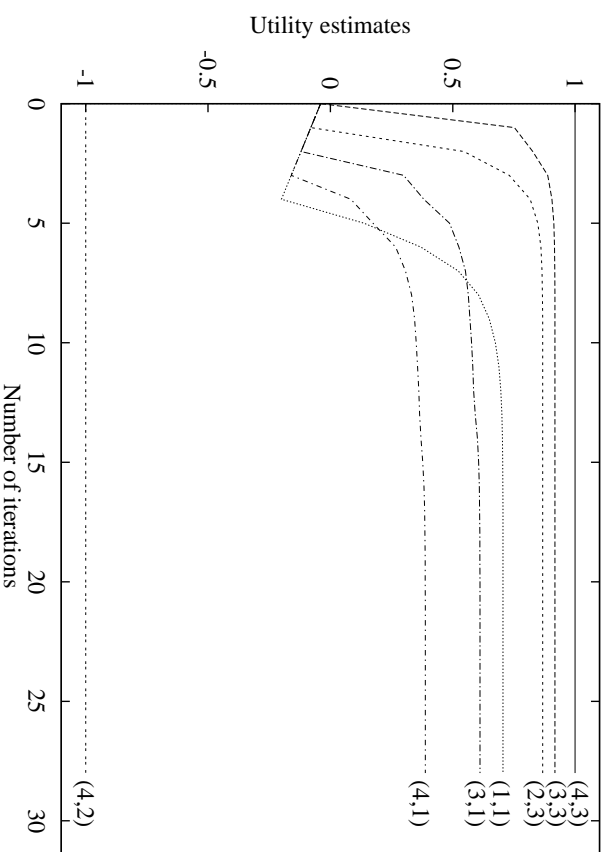
Idea: Start with arbitrary utility values

Update to make them locally consistent with Bellman eqn.

Everywhere locally consistent  $\Rightarrow$  global optimality

repeat until “no change”

$$U(i) \leftarrow R(i) + \max_a \sum_j U(j) M_{ij}^a \quad \text{for all } i$$





## Policy iteration (Howard, 1960)

Idea: search for optimal policy and utility values simultaneously

Algorithm:

- $\pi \leftarrow$  an arbitrary initial policy
- repeat until no change in  $\pi$ 
  - compute utilities given  $\pi$
  - update  $\pi$  as if utilities were correct (i.e., local MEU)

To compute utilities given a fixed  $\pi$ :

$$U(i) = R(i) + \sum_j U(j) M_{ij}^{\pi(i)} \quad \text{for all } i$$

i.e.,  $n$  simultaneous linear equations in  $n$  unknowns, solve in  $O(n^3)$

## What if I live forever? (digression)

Using the additive definition of utilities,  $U(i)$ s are infinite!

Moreover, value iteration fails to terminate

How should we compare two infinite lifetimes?

1) Discounting: future rewards are discounted at rate  $\gamma \leq 1$

$$U([s_0, \dots, s_\infty]) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

Maximum utility bounded above by  $R_{\max}/(1 - \gamma)$

Smaller  $\gamma \Rightarrow$  shorter horizon

2) Maximize system gain = average reward per time step

Theorem: optimal policy has constant gain after initial transient

E.g., taxi driver's daily scheme cruising for passengers