Introduction to Artificial Intelligence

Complex decisions

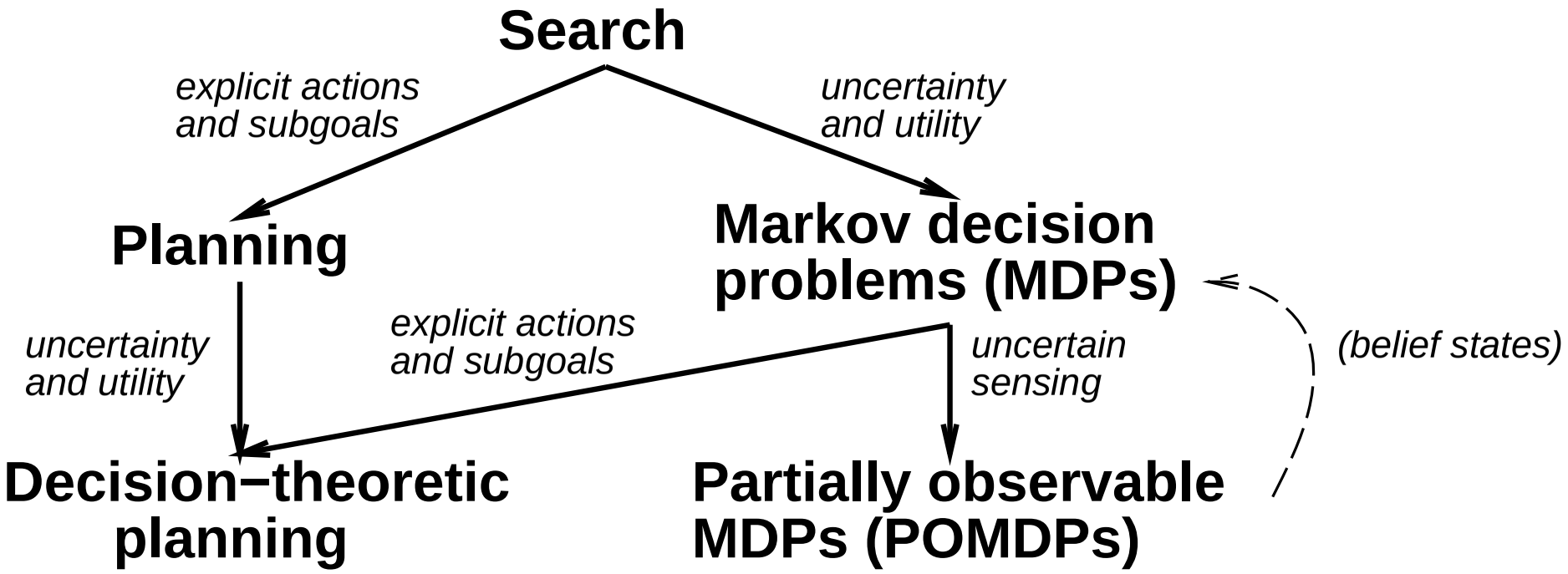
Chapter 17, Sections 1–3

Dieter Fox

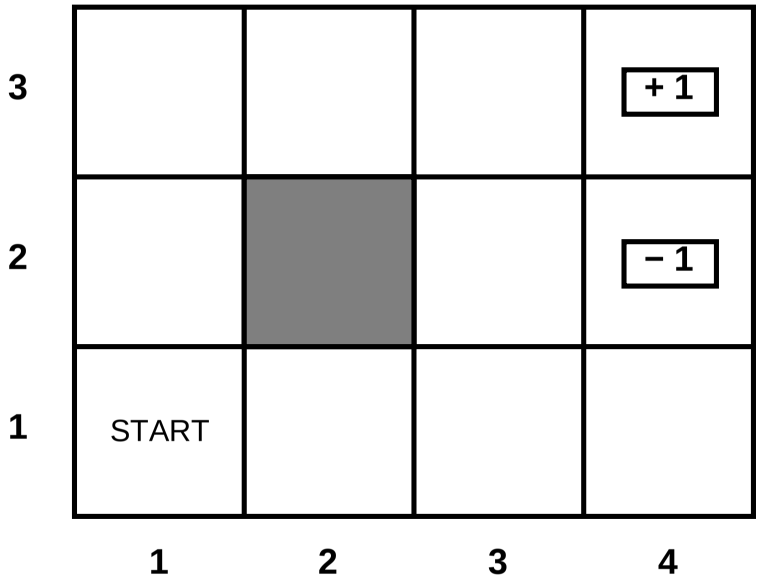
Outline

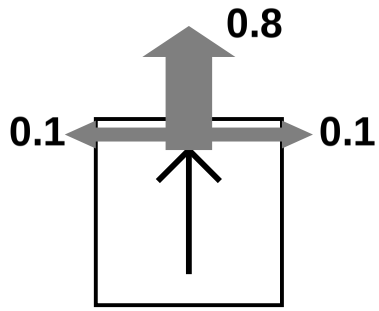
- ♢ Decision problems
- ♢ Value iteration
- ♢ Policy iteration

Sequential decision problems



Example MDP





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

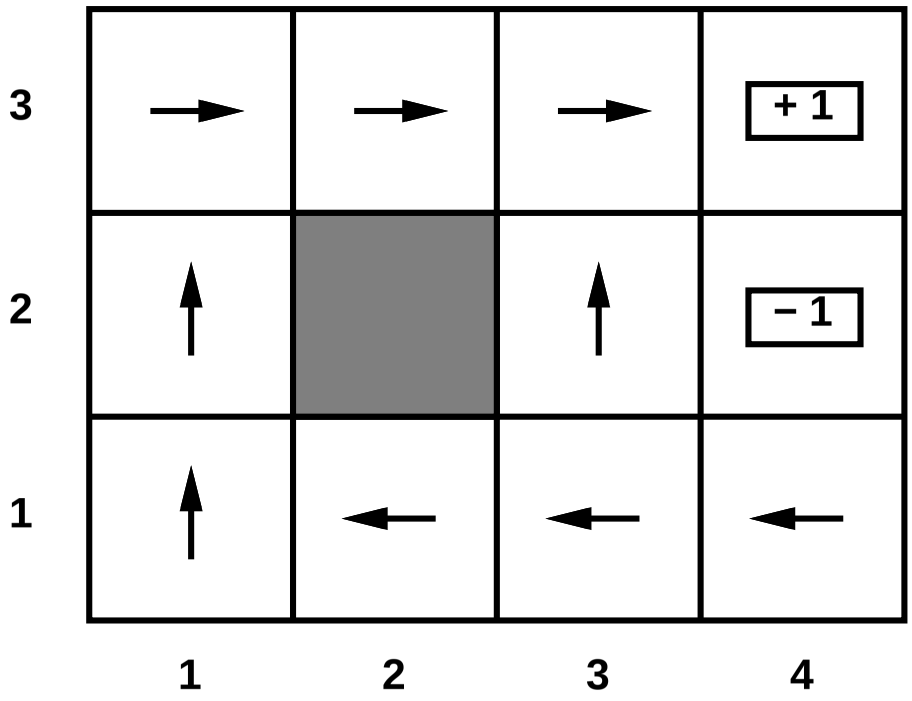
= ± 1 for terminal states

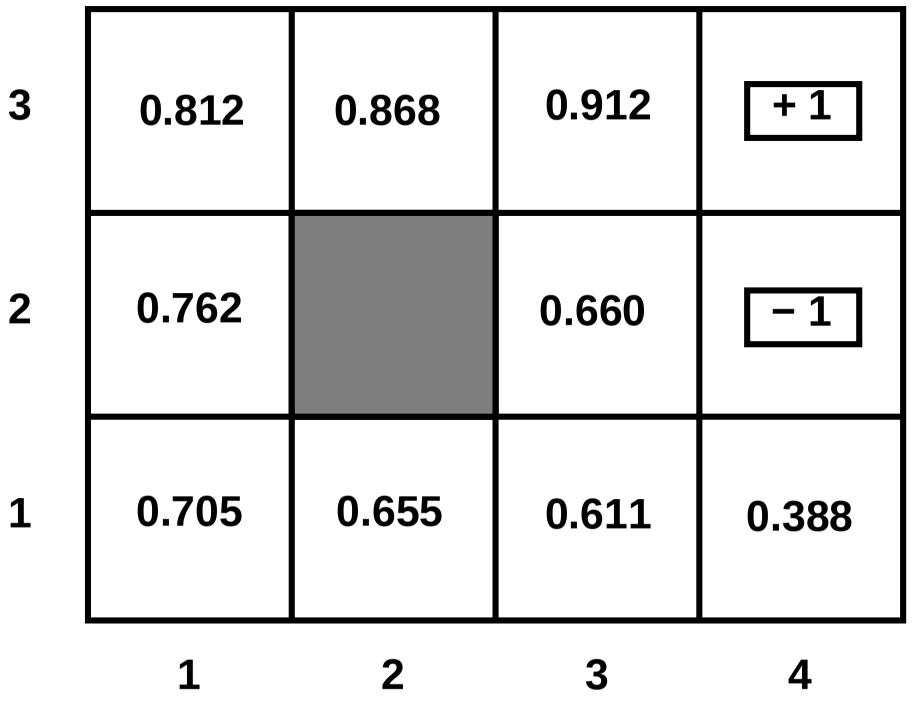
Solving MDPs

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 $R(i)$:





Utility

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

Usually use an *additive* utility function:

$$U([s_1, s_2, s_3, \ldots, s_n]) = R(s_1) + R(s_2) + R(s_3) + \cdots + R(s_n)$$

(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**

$U(s_i) =$ **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:

expected sum of rewards
= **current reward**
+ **expected sum of rewards after taking best action**

Bellman equation (1957):

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

$$\begin{aligned} U(1,1) = & -0.04 \\ & + \max\{0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1), && \textit{up} \\ & + \max\{0.9U(1,1) + 0.1U(1,2) && \textit{left} \\ & + \max\{0.9U(1,1) + 0.1U(2,1) && \textit{down} \\ & + \max\{0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1)\} && \textit{right} \end{aligned}$$

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

Value iteration algorithm

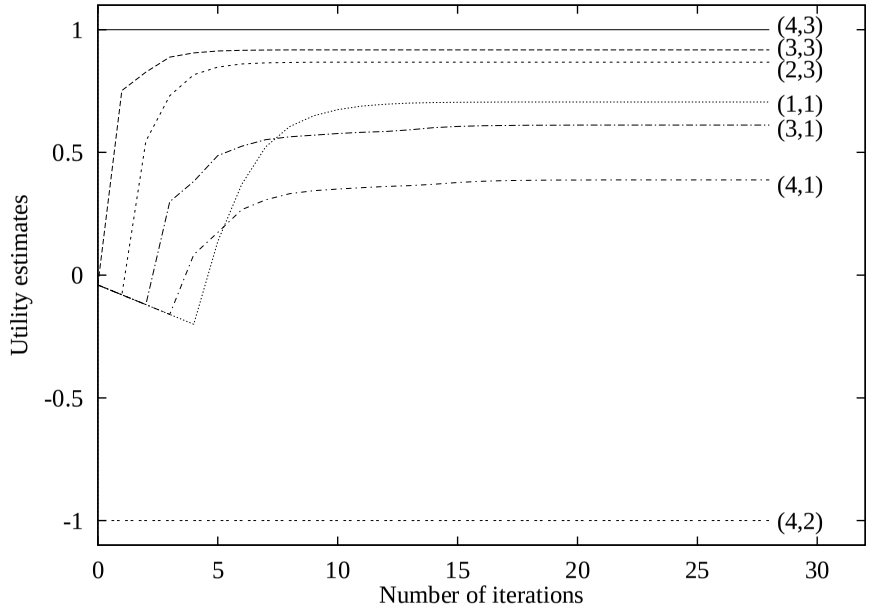
Idea: Start with arbitrary utility values

Update to make them **locally consistent** with Bellman eqn.

Everywhere locally consistent $\Rightarrow$ global optimality

repeat until "no change"

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



Summary

- We can design rational agents based on **probability theory** and **utility theory**
- Sequential decision making in stochastic environments (MDPs) can be solved by computing a **policy**
- **Value iteration** is an algorithm for computing optimal policies.