# Stanford University

AA228/CS238: Decision Making Under Uncertainty

Fall 2023

Liam Kruse • Bishop Auditorium • email: *lkruse@stanford.edu*

---

**PROBLEM SESSION 4: EXACT SOLUTION METHODS** **October 18, 2023 4:00pm PT**

**Topic 1. MDP Overview**

a) Markov Decision Process (MDP): defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$

- $\mathcal{S}$ - State Space: the environment, the *minimum information set* required to make a decision

- $\mathcal{A}$ - Action Space: what the agent can do

- $T$ - Transition model: system dynamics (how the system evolves)

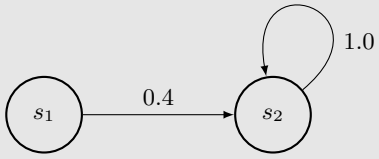



Figure 1: Simple MDP

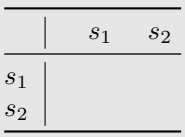



Figure 2: Transition Model

- $R$ - Reward Function: expected reward from taking action $a$ in state $s$ and transitioning to state $s'$

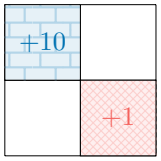
- $\gamma$ - Discount factor: used to weight future rewards

b) Utility: a discounted sequence of rewards

- Utility of a sequence of states **without** discounting: why is this problematic?

$$U([s_1, s_2, \ldots, s_n)]) = \sum_{t=1}^{n} r_t$$

- Thought exercise: would an agent want to collect rewards in the blue cell (bricks) or the red cell (crosshatch) forever?




  Is there a preference for $10 + 10 + 10 + \ldots$ or $1 + 1 + 1 + \ldots$ as $n \to \infty$ ("infinite horizon")?
- Solution: discount with $\gamma$!

c) Policy $\pi$: a function of the state that tells you *what to do* in every state

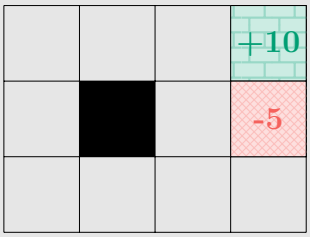



Figure 3: Optimal Path

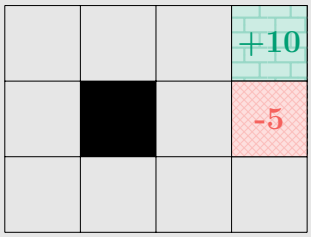


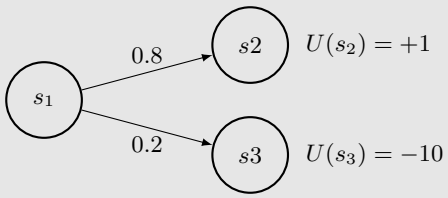

Figure 4: Optimal Policy

- $U^{\pi}(s) \rightarrow$ utility from executing policy $\pi$ from state $s$ (the *value function*)
- $\pi^*(s) = \arg\max_{\pi} U^{\pi}(s)$

d) Bellman Equation: "The expected utility of a state is the reward at that state plus the discounted sum of expected future rewards."

e) A Note On Expectation

An expected value is a *weighted average*

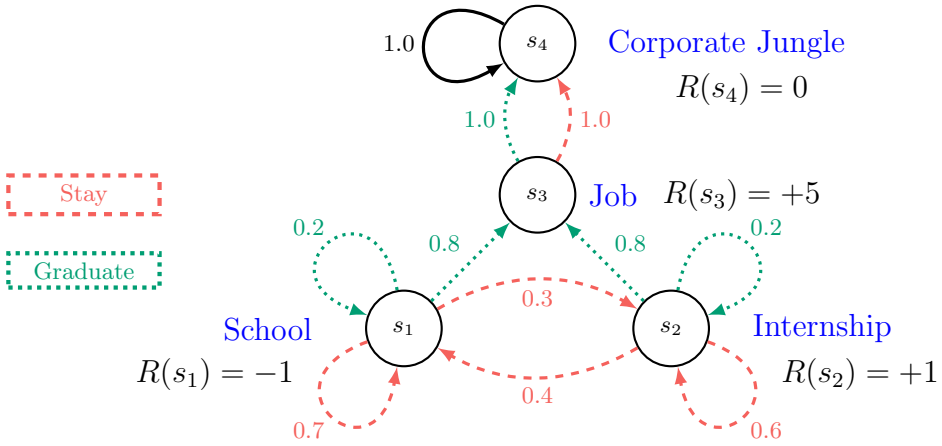



What is the expected utility when transitioning out of $s_1$?

**Topic 2. Value Iteration Example**

**Algorithm 1** The Value Iteration Algorithm

```
procedure VALUE ITERATION(P :: MDP, k_max)
    U(s) ← 0 for all s ∈ S
    for k ← 1, k_max do
        for all s ∈ P.S do
            U_{k+1}(s) = max_a (R(s, a) + γ Σ_{s'} T(s' | s, a) U_k(s'))
        end for
    end for
end procedure
```




a) Define the tuple for this MDP

b) Perform two iterations of value iteration:

c) What is our policy after two rounds of value iteration?

d) What is the time complexity of value iteration?