

Planning and Optimization

F2. Bellman Equation & Linear Programming

Malte Helmert and Gabriele Röger

Universität Basel

November 30, 2020

Planning and Optimization

November 30, 2020 — F2. Bellman Equation & Linear Programming

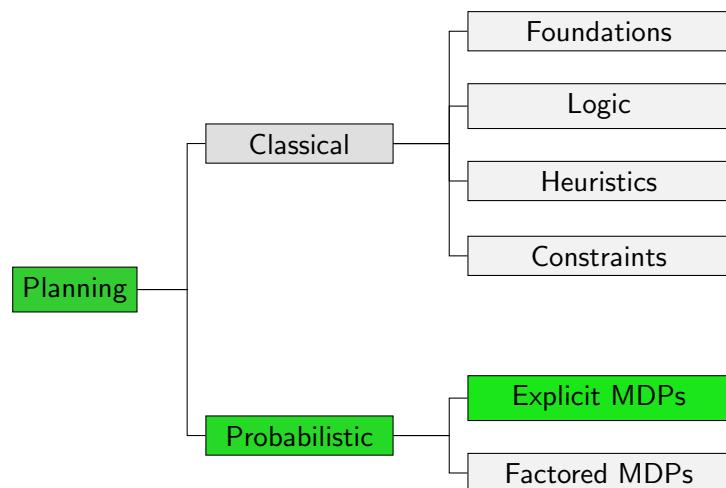
F2.1 Introduction

F2.2 Bellman Equation

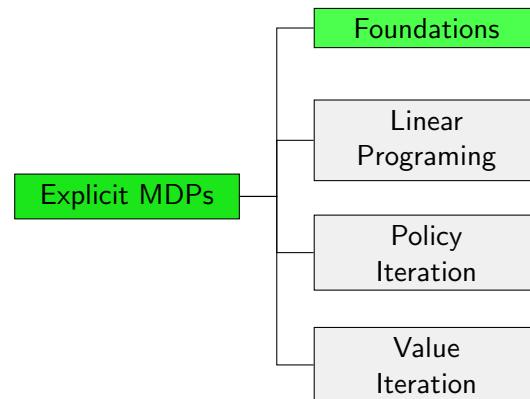
F2.3 Linear Programming

F2.4 Summary

Content of this Course



Content of this Course: Explicit MDPs



F2.1 Introduction

Quality of Solutions

- ▶ Solution in classical planning: plan
- ▶ Optimality criterion of a solution in classical planning: minimize plan cost
- ▶ Solution in probabilistic planning: policy
- ▶ What is the optimality criterion of a solution in probabilistic planning?

Example: Swiss Lotto

Example (Swiss Lotto)

What is the **expected payoff** of placing one bet in Swiss Lotto for a cost of CHF 2.50 with (simplified) payouts and probabilities:

CHF 30.000.000 with prob. 1/31474716	(6 + 1)
CHF 1.000.000 with prob. 1/5245786	(6)
CHF 5.000 with prob. 1/850668	(5)
CHF 50 with prob. 1/111930	(4)
CHF 10 with prob. 1/11480	(3)

Solution:

$$\frac{30000000}{31474716} + \frac{1000000}{5245786} + \frac{5000}{850668} + \frac{50}{111930} + \frac{10}{11480} - 2.5 \approx -1.35.$$

Expected Values under Uncertainty

Definition (Expected Value of a Random Variable)

Let X be a random variable with a finite number of **outcomes** $d_1, \dots, d_n \in \mathbb{R}$, and let d_i happen with probability $p_i \in [0, 1]$ (for $i = 1, \dots, n$) s.t. $\sum_{i=1}^n p_i = 1$.

The **expected value** of X is $\mathbb{E}[X] = \sum_{i=1}^n (p_i \cdot d_i)$.

F2.2 Bellman Equation

Value Functions for MDPs

Definition (Value Functions for MDPs)

Let π be a policy for MDP $\mathcal{T} = \langle S, A, R, T, s_0, \gamma \rangle$.

The **state-value** $V_\pi(s)$ of $s \in S_\pi(s_0)$ under π is defined as

$$V_\pi(s) := Q_\pi(s, \pi(s))$$

where the **action-value** $Q_\pi(s, a)$ of s and a under π is defined as

$$Q_\pi(s, a) := R(s, a) + \gamma \cdot \sum_{s' \in \text{succ}(s, a)} T(s, a, s') \cdot V_\pi(s').$$

The state-value $V_\pi(s)$ describes the **expected reward** of applying π in MDP \mathcal{T} , starting from s .

Bellman Equation in MDPs

Definition (Bellman Equation in MDPs)

Let $\mathcal{T} = \langle S, A, R, T, s_0, \gamma \rangle$ be an MDP.

The **Bellman equation** for a state s of \mathcal{T} is the set of equations that describes $V_*(s)$, where

$$V_*(s) := \max_{a \in A(s)} Q_*(s, a)$$

$$Q_*(s, a) := R(s, a) + \gamma \cdot \sum_{s' \in \text{succ}(s, a)} T(s, a, s') \cdot V_*(s').$$

The solution $V_*(s)$ of the Bellman equation describes the **maximal expected reward** that can be achieved from state s in MDP \mathcal{T} .

Optimal Policy in MDPs

What is the policy that achieves the maximal expected reward?

Definition (Optimal Policy in MDPs)

Let $\mathcal{T} = \langle S, A, R, T, s_0, \gamma \rangle$ be an MDP.

A policy π is an **optimal policy** if $\pi(s) \in \arg \max_{a \in A(s)} Q_*(s, a)$ for all $s \in S_\pi(s_0)$ and the **expected reward** of π in \mathcal{T} is $V_*(s_0)$.

Value Functions for SSPs

Definition (Value Functions for SSPs)

Let $\mathcal{T} = \langle S, A, c, T, s_0, S_* \rangle$ be an SSP and π be a policy for \mathcal{T} .

The **state-value** $V_\pi(s)$ of s under π is defined as

$$V_\pi(s) := \begin{cases} 0 & \text{if } s \in S_* \\ Q_\pi(s, \pi(s)) & \text{otherwise,} \end{cases}$$

where the **action-value** $Q_\pi(s, a)$ of s and a under π is defined as

$$Q_\pi(s, a) := c(a) + \sum_{s' \in \text{succ}(s, a)} T(s, a, s') \cdot V_\pi(s').$$

The state-value $V_\pi(s)$ describes the **expected cost** of applying π in SSP \mathcal{T} , starting from s .

Optimal Policy in SSPs

What is the policy that achieves the minimal expected cost?

Definition (Optimal Policy in SSPs)

Let $\mathcal{T} = \langle S, A, c, T, s_0, S_* \rangle$ be an SSP.

A policy π is an **optimal policy** if $\pi(s) \in \arg \min_{a \in A(s)} Q_*(s, a)$ for all $s \in S_\pi(s_0) \setminus S_*$ and the **expected cost** of π in \mathcal{T} is $V_*(s_0)$.

Bellman Equation in SSPs

Definition (Bellman Equation in SSPs)

Let $\mathcal{T} = \langle S, A, c, T, s_0, S_* \rangle$ be an SSP.

The **Bellman equation** for a state s of \mathcal{T} is the set of equations that describes $V_*(s)$, where

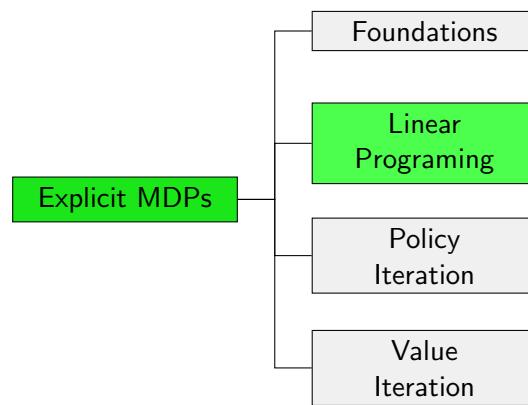
$$V_*(s) := \begin{cases} 0 & \text{if } s \in S_* \\ \min_{a \in A(s)} Q_*(s, a) & \text{otherwise,} \end{cases}$$

$$Q_*(s, a) := c(a) + \sum_{s' \in \text{succ}(s, a)} T(s, a, s') \cdot V_*(s').$$

The solution $V_*(s)$ of the Bellman equation describes the **minimal expected cost** that can be achieved from state s in SSP \mathcal{T} .

F2.3 Linear Programming

Content of this Course: Explicit MDPs



Linear Programming for SSPs

- ▶ Bellman equation: set of equations that describes the **expected cost** for each state.
- ▶ there are $|S|$ variables and $|S|$ equations (replacing Q_* in V_* with the corresponding equation)
- ▶ If we solve these equations, we can determine an optimal policy for the SSP from the state-values.
- ▶ **Problem:** how can we deal with the **minimization**?

⇒ We have solved the “same” problem before with the help of an LP solver

Reminder: LP for Shortest Path in State Space

Variables

Non-negative variable $Distance_s$ for each state s

Objective

Maximize $Distance_{s_0}$

Subject to

$Distance_{s_*} = 0$ for all goal states s_*

$Distance_s \leq Distance_{s'} + c(\ell)$ for all transitions $s \xrightarrow{\ell} s'$

LP for Expected Cost in SSP

Variables

Non-negative variable $ExpCost_s$ for each state s

Objective

Maximize $ExpCost_{s_0}$

Subject to

$ExpCost_{s_*} = 0$ for all goal states s_*

$ExpCost_s \leq \left(\sum_{s' \in S} T(s, a, s') \cdot ExpCost_{s'} \right) + c(a)$

for all $s \in S$ and $a \in A(s)$

LP for Expected Reward in MDP

Variables

Non-negative variable ExpReward_s for each state s

Objective

Minimize ExpReward_{s_0}

Subject to

$$\text{ExpReward}_s \geq (\gamma \cdot \sum_{s' \in S} T(s, a, s') \text{ExpReward}_{s'}) + R(s, a)$$

for all $s \in S$ and $A \in A(s)$

Complexity of Probabilistic Planning

- ▶ an **optimal solution** for MDPs or SSPs can be computed with an **LP solver**
- ▶ requires $|S|$ variables and $|S| \cdot |A|$ constraints
- ▶ we know that LPs can be solved in **polynomial time**
- ▶ \Rightarrow solving MDPs or SSPs is a **polynomial time** problem

How does this relate to the complexity result for classical planning?

Solving MDPs or SSPs is polynomial in $|S| \cdot |A|$.

F2.4 Summary

Summary

- ▶ The state-values of a policy specify the **expected reward (cost)** of following that policy.
- ▶ The **Bellman equation** describes the state-values of an optimal policy.
- ▶ **Linear Programming** can be used to solve MDPs and SSPs in time **polynomial** in the size of the MDP/SSP.