Planning and Optimization

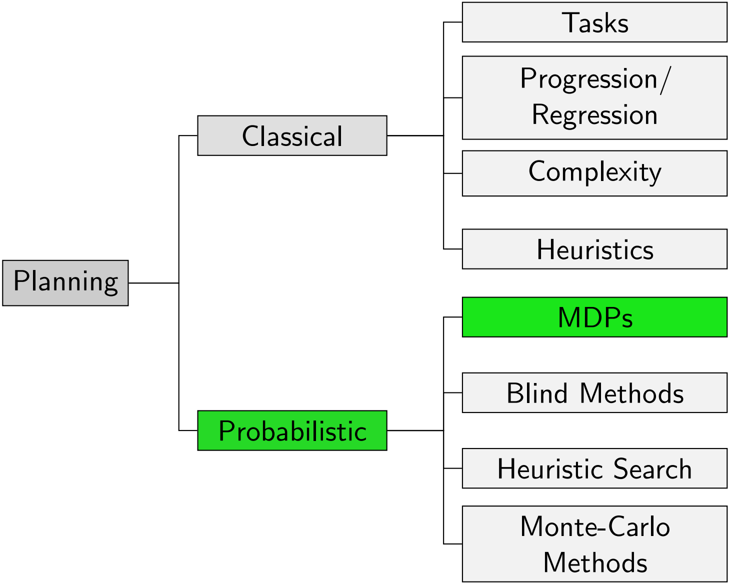
F2. Policies & Compact Description

Gabriele Röger and Thomas Keller

Universität Basel

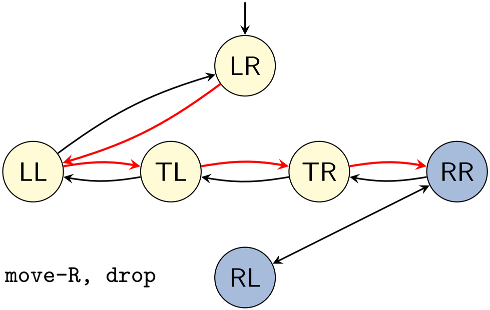
November 21, 2018

Content of this Course



Policies & Value Functions

Solutions in SSPs

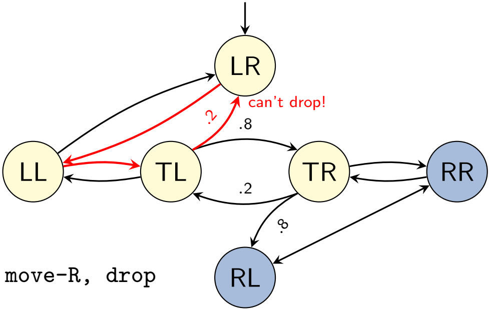


`move-L, pickup, move-R, drop`

- solution in deterministic transition systems is plan, i.e., a goal path from s_0 to some $s_\star \in S_\star$
- cheapest plan is optimal solution
- deterministic agent that executes plan will reach goal

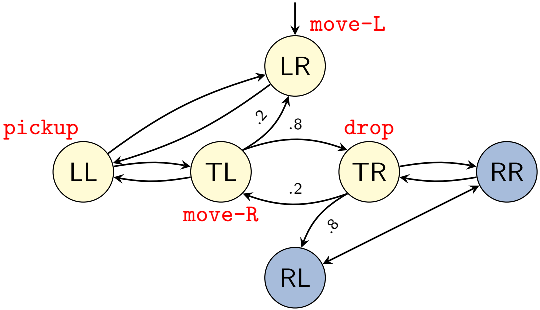
Solutions in SSPs

`move-L, pickup, move-R, drop`



- probabilistic agent will not reach goal or cannot execute plan
- non-determinism can lead to different outcome than anticipated in plan
- require a more general solution: a policy

Solutions in SSPs



- policy must be allowed to be cyclic
- policy must be able to branch over outcomes
- policy assigns applicable labels to states

Policy for SSPs

Definition (Policy for SSPs)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ be an SSP. A policy for $\mathcal{T}$ is a mapping $\pi : S \to L \cup \{\bot\}$ such that $\pi(s) \in L(s) \cup \{\bot\}$ for all s.

The set of reachable states $S_\pi(s)$ from s under π is defined recursively as the smallest set satisfying the rules

- $s \in S_\pi(s)$ and
- $\text{succ}(s', \pi(s')) \subseteq S_\pi(s)$ for all $s' \in S_\pi(s) \setminus S_\star$ where $\pi(s') \neq \bot$.

If $\pi(s') \neq \bot$ for all $s' \in S_\pi(s)$, then π is executable in s.

Policy Representation

- size of explicit representation of executable policy π is $|S_\pi(s_0)|$
- often, $|S_\pi(s_0)|$ similar to $|S|$
- compact policy representation, e.g. via value function approximation or neural networks, is active research area
  $\Rightarrow$ not covered in this course
- instead, we consider small state spaces for basic algorithms
- or online planning where planning for the current state s_0 is interleaved with execution of $\pi(s_0)$

Value Functions of SSPs

Definition (Value Functions of SSPs)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ be an SSP and π be an executable policy for $\mathcal{T}$. The state-value $V_\pi(s)$ of s under π is defined as

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

where the action-value $Q_\pi(s, \ell)$ under π is defined as

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

Example: Value Functions of SSPs

Example

Consider example task and π with $\pi(\mathsf{LR}) = \texttt{move-L}$, $\pi(\mathsf{LL}) = \texttt{pickup}$, $\pi(\mathsf{TL}) = \texttt{move-R}$ and $\pi(\mathsf{TR}) = \texttt{drop}$.

$$V_\star(\mathsf{LR}) = 1 + V_\star(\mathsf{LL})$$
$$V_\star(\mathsf{LL}) = 1 + V_\star(\mathsf{TL})$$
$$V_\star(\mathsf{TL}) = 1 + (0.8 \cdot V_\star(\mathsf{RR})) + (0.2 \cdot V_\star(\mathsf{LR}))$$
$$V_\star(\mathsf{TR}) = 1 + V_\star(\mathsf{RR})$$
$$V_\star(\mathsf{RL}) = 0$$
$$V_\star(\mathsf{RR}) = 0$$

What is the solution of this? $\Rightarrow$ next week!

Bellman Optimality Equation

Definition (Optimal Policy in SSPs)

Let the Bellman optimality equation for a state s of an SSP be the set of equations that describes $V_\star(s)$, where

$$V_\star(s) := \begin{cases} 0 & \text{if } s \in S_\star \\ \min_{\ell \in L(s)} Q_\star(s,\ell) & \text{otherwise,} \end{cases}$$

$$Q_\star(s,\ell) := c(\ell) + \sum_{s' \in \mathrm{succ}(s,\ell)} \left(T(s,\ell,s') \cdot V_\star(s') \right).$$

A policy $\pi^\star$ is an optimal policy if $\pi^\star(s) \in \arg\min_{\ell \in L(s)} Q_\star(s,\ell)$ for all $s \in S$, and the expected cost of $\pi^\star$ in $\mathcal{T}$ is $V_\star(s_0)$.

Dead-end States

- dead-end states are a problem with our formalization
- each policy with non-zero probability of reaching a dead-end has infinite state-value
- one solution is to search for policy with highest probability to reach the goal
- unfortunately, this ignores costs
- there is also research on dead-end detection
- in this course, we only consider SSPs, FH-MDPs and DR-MDPs that are dead-end free

Policies for FH-MDPs

- What is the optimal policy for the SSP at the blackboard?

Policies for FH-MDPs

- What is the optimal policy for the SSP at the blackboard?
- Can we do better if we regard this as an FH-MDP?

Policies for FH-MDPs

- What is the optimal policy for the SSP at the blackboard?
- Can we do better if we regard this as an FH-MDP?
- Yes, by acting differently close to the horizon.

Policy for FH-MDPs

Definition (Policy for FH-MDPs)

Let $\mathcal{T} = \langle S, L, R, T, s_0, H \rangle$ be an FH-MDP. A policy for $\mathcal{T}$ is a mapping $\pi : S \times \{1, \dots, H\} \rightarrow L \cup \{\perp\}$ such that $\pi(s, d) \in L(s) \cup \{\perp\}$ for all s.

The set of reachable states $S_\pi(s, d)$ from s with d steps-to-go under π is defined recursively as the smallest set satisfying the rules

- $\langle s, d \rangle \in S_\pi(s, d)$ and
- $\langle s'', d' - 1 \rangle \in S_\pi(s, d)$ for all $s'' \in \text{succ}(s', \pi(s'))$ and $\langle s', d' \rangle \in S_\pi(s)$ with $d' > 0$ and $\pi(s', d') \neq \perp$.

If $\pi(s', d') \neq \perp$ for all $\langle s', d' \rangle \in S_\pi(s, d)$ with $d' > 0$, then π is executable in s.

Value Functions for FH-MDPs

Definition (Value Functions for FH-MDPs)

Let $\mathcal{T} = \langle S, L, c, T, s_0, H \rangle$ be an FH-MDP and π be an executable policy for $\mathcal{T}$. The state-value $V_\pi(s, d)$ of s and d under π is defined as

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

where the action-value $Q_\pi(s, d, \ell)$ under π is defined as

$$Q_\pi(s, d, \ell) := R(s, \ell) + \sum_{s' \in \text{succ}(s, \ell)} \big(T(s, \ell, s') \cdot V_\pi(s', d - 1) \big).$$

Bellman Optimality Equation

Definition (Optimal Policy in FH-MDPs)

Let the Bellman optimality equation for a state s of an FH-MDP be the set of equations that describes $V_\star(s, d)$, where

$$V_\star(s, d) := \begin{cases} 0 & \text{if } d = 0 \\ \max_{\ell \in L(s)} Q_\star(s, d, \ell) & \text{otherwise,} \end{cases}$$

$$Q_\star(s, d, \ell) := R(s, \ell) + \sum_{s' \in \mathrm{succ}(s, \ell)} \left(T(s, \ell, s') \cdot V_\star(s', d - 1) \right).$$

A policy $\pi^\star$ is an optimal policy if $\pi^\star(s, d) \in \arg\max_{\ell \in L(s)} Q_\star(s, d, \ell)$ for all $s \in S$ and $d \in \{1, \ldots, H\}$, and the expected reward of $\pi^\star$ in $\mathcal{T}$ is $V_\star(s_0, H)$.

(Optimal) Policy and Value Functions for DR-MDPs

- policy does not distinguish states based on steps-to-go (or rather the reverse “distance-from-init”)
- value functions have no “terminal case”
- value functions discount reward with γ
- Bellman optimality equation derived from value functions as for FH-MDP

Factored MDPs

Factored SSPs

We would like to specify huge SSPs without enumerating states. In classical planning, we achieved this via propositional planning tasks:

- represent different aspects of the world in terms of different Boolean state variables
- treat state variables as atomic propositions
  $\rightsquigarrow$ a state is a valuation of state variables
- n state variables induce 2^n states
  $\rightsquigarrow$ exponentially more compact than "flat" representations

$\Rightarrow$ can also be used for SSPs

Reminder: Syntax of Operators

Definition (Operator)

An operator o over state variables V is an object with three properties:

- a precondition $pre(o)$, a logical formula over V
- an effect $eff(o)$ over V, defined on the following slides
- a cost $cost(o) \in \mathbb{R}_0^+$

$\Rightarrow$ can also be used for SSPs

Reminder: Syntax of Effects

Definition (Effect)

Effects over state variables V are inductively defined as follows:

- If $v \in V$ is a state variable, then v and $\neg v$ are effects (atomic effect).
- If $e_1, \dots, e_n$ are effects, then $(e_1 \land \dots \land e_n)$ is an effect (conjunctive effect).
  The special case with $n = 0$ is the empty effect $\top$.
- If χ is a logical formula and e is an effect, then $(\chi \rhd e)$ is an effect (conditional effect).

Parentheses can be omitted when this does not cause ambiguity.

Syntax of Probabilistic Effects

Definition (Effect)

Effects over state variables V are inductively defined as follows:

- If $v \in V$ is a state variable, then v and $\neg v$ are effects (atomic effect).
- If $e_1, \dots, e_n$ are effects, then $(e_1 \wedge \dots \wedge e_n)$ is an effect (conjunctive effect).
  The special case with $n = 0$ is the empty effect $\top$.
- If χ is a logical formula and e is an effect, then $(\chi \rhd e)$ is an effect (conditional effect).
- If $e_1, \dots, e_n$ are effects and $p_1, \dots, p_n \in [0, 1]$ such that $\sum_{i=1}^{n} p_i = 1$, then $(p_1 : e_1 | \dots | p_n : e_n)$ is an effect (probabilistic effect).

Parentheses can be omitted when this does not cause ambiguity.

- FDR tasks can be generalized to SSPs in the same way
- both propositional and FDR tasks can be generalized to FH-MDPs and DR-MDPs

Summary

Summary

- Policies consider branching and cycles
- State-value of a policy describes expected reward of following that policy
- Related Bellman optimality equation describes optimal policy
- Compact descriptions that induce SSPs and MDPs analogous to classical planning