Panel on Probabilistic Planning

Héctor Geffner, Universidad Simón Bolívar

$$\begin{array}{c}\textbf{Action}\\ \textbf{Model}\\ \textbf{Goals}\end{array} \longrightarrow \textbf{Planner} \longrightarrow \textbf{Controller} \;\begin{array}{c}\xrightarrow{actions}\\ \xleftarrow{observations}\end{array}\; \textbf{World}$$

Three key elements:

- **Models:** to understand general task
- **Languages:** to describe specific problems
- **Algorithms:** to solve them

Two important distinctions

- **dynamics:** deterministic vs. stochastic actions
- **feedback:** open vs. closed loop control

Classical Planning and SEARCH

- Classical Planning, i.e., open-loop planning, with deterministic actions and known initial state is a SEARCH problem:

  - state space S
  - initial state $s_0 \in S$
  - actions $A(s) \subseteq A$ applicable in each state $s \in S$
  - a transition function $f(s, a)$ for $s \in S$ and $a \in A(s)$
  - action costs $c(a, s) > 0$
  - a set $G \subseteq S$ of goal states

- A *solution* is a sequence of actions that leads to the goal
- A solution is *optimal* if it has minimum cost

Closed-Loop Planning and MDPs

- To understand *closed-loop* planning other mathematical models are needed.
- Closed-loop planning with *full feedback* is a **Markov Decision Problem:**

  - a state space S
  - actions $A(s) \subseteq A$ applicable in each state $s \in S$
  - transition probabilities $P_a(s'|s)$ for $s \in A$ and $a \in A(s)$
  - action costs $c(a, s) > 0$
  - a set $G \subseteq S$ of goal states

- Small change in model; significant change in **form** of the solution
- *Solutions* become functions (policies) mapping states into actions
- *Optimal* solutions have minimal *expected* costs
- Bellman, Putterman, Bertsekas, Dean, Russell, Boutilier, ...

Closed-Loop Planning and POMDPs

- Closed-loop planning with **partial feedback** is a **Partially Observable MDP:**

  - states $s \in S$
  - actions $A(s) \subseteq A$ s
  - costs $c(a, s) > 0$
  - transition probabilities $P_a(s'|s)$ for $s \in S$ and $a \in A(s)$
  - initial belief state b_0
  - final belief states b_F
  - observations o after action a with probabilities $P_a(o|s)$

- **Solutions** map belief states into actions (Sondik, Littman, Cassandra, Kaebling, ...)
- **Optimal** solutions minimize expected cost to final belief states

Representation Languages

- Action languages are well-understood today and can be used to defined SEARCH, MDP and POMDP models at a high-level ('Strips' like)

- e.g., the POMDP underlying Levesque's Omelette problem can be described as:

Action:	**grab-egg**$()$
Precond:	$\neg holding$
Effects:	$holding :=$ **true**
	$good? :=$ (**true** 0.5 ; **false** 0.5)
Action:	**break-egg**$(bowl : BOWL)$
Precond:	$holding \wedge (ngood(bowl) + nbad(bowl)) < 4$
Effects:	$holding :=$ **false**
	$good? \;\rightarrow\; ngood(bowl) := ngood(bowl) + 1$
	$\neg good? \;\rightarrow\; nbad(bowl) := nbad(bowl) + 1$
Action:	**clean**(bowl:BOWL)
Precond:	$\neg holding$
Effects:	$ngood(bowl) := 0 \;\; , nbad(bowl) := 0$
Action:	**inspect**$(bowl : BOWL)$
Effect:	**obs**$(nbad(bowl) > 0)$

Algorithms

- **Dynamic Programming** algorithms solve SEARCH, MDP and POMDPs using Bellman Equation to update cost estimates:

$$
\begin{array}{rl}
\text{SEARCH:} & V(s) := \min_{a \in A(s)}[c(a,s) + V(s_a)] \\
\text{MDP:} & V(s) := \min_{a \in A(s)}[c(a,s) + \sum_{s'} P_a(s'|s)V(s')] \\
\text{POMDP:} & V(b) := \min_{b \in A(b)}[c(a,b) + \sum_{o} P_a(o|b)V(b_a^o)]
\end{array}
$$

- **Problem:** most updates wasted on irrelevant states
- **Key idea:** use estimates to guide a greedy search, and only update states encountered (LRTA*, RTDP)
- **Result 1:** if initial estimates are admissible, optimality retained (Korf, Barto *et. al.*, Bertsekas)
- **Result 2:** if estimates are good enough, search is focused, and method can scale up to **large** problems

Summary

- General, well-founded and (we believe) practical approach to planning, given by
  1. SEARCH, MDP, and POMDP models
  2. high-level action languages
  3. real time dynamic programming algorithms
- Empirical results over a number of tasks
  - classical planning
  - contingent planning
  - sorting
  - decision tree induction
  - ⋮
- A number of open challenges
  - generalization
  - optimality bounds
  - ⋮