Algorithmic Human-Robot Interaction

Task Planning II

CSCI 7000

Prof. Brad Hayes

Computer Science Department

University of Colorado Boulder

Papers for next Thursday:
Need 1 PRO (10m) and 1 CON (5m) speaker each

1. Designing Robot Learners that Ask Good Questions

   Maya Cakmak and Andrea Thomaz

2. Anticipating human actions for collaboration in the presence of task and sensor uncertainty

   Kelsey Hawkins et al.

Last Time…

Fully Observable, Deterministic: Classical AI Planning

- finite and discrete state space S
- a **known initial state** $s_0 \in S$
- a set $S_G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each $s \in S$
- a **deterministic transition function** $s' = f(a, s)$ for $a \in A(s)$
- positive **action costs** $c(a, s)$

A **solution** or **plan** is a sequence of applicable actions $\pi = a_0, \ldots, a_n$ that maps s_0 into S_G; i.e., there are states $s_0, \ldots, s_{n+1}$ s.t. $s_{i+1} = f(a_i, s_i)$ and $a_i \in A(s_i)$ for $i = 0, \ldots, n$, and $s_{n+1} \in S_G$.

The plan is **optimal** if it minimizes the **sum of action costs** $\sum_{i=0,n} c(a_i, s_i)$. If costs are all 1, plan cost is plan **length**

Different **models** obtained by relaxing assumptions in **bold** . . .

Uncertainty but No Feedback: Conformant Planning

- finite and discrete state space S
- a **set of possible initial state** $S_0 \in S$
- a set $S_G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each $s \in S$
- a **non-deterministic** transition function $F(a, s) \subseteq S$ for $a \in A(s)$
- uniform action costs $c(a, s)$

A **solution** is still an **action sequence** but must achieve the goal for **any possible initial state and transition**

More complex than **classical planning**, verifying that a plan is **conformant** intractable in the worst case; but special case of **planning with partial observability**

Planning with Markov Decision Processes

MDPs are **fully observable, probabilistic** state models:

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

– **Solutions** are **functions (policies)** mapping states into actions

– **Optimal** solutions minimize **expected cost** to goal

What about uncertainty?

Adding Uncertainty to MDPs

- Traditional MDPs are defined with:
  - States – {(0,0), (0,1), …}
  - Actions – {move_north, …}
  - Rewards – R(S,A,S’) -> Reward
  - Transition Probabilities – P(State | State,Action)

- But what if I don’t know what state I’m in?

  - Sometimes state variables can be ***latent***
    (e.g., “is the campus wi-fi working?”,
    “is my teammate in a bad mood today?”)

  - Rather than directly measuring them, we receive ***observations***
    (e.g., “nobody’s staring into their laptops”,
    “my teammate just punched Sawyer in its tablet-face”)

Adding Uncertainty to MDPs

- Traditional MDPs are defined with:
  - States – {(0,0), (0,1), …}
  - Actions – {move_north, …}
  - Rewards – R(S,A,S') -> Reward
  - Transition Probabilities – P(State | State,Action)

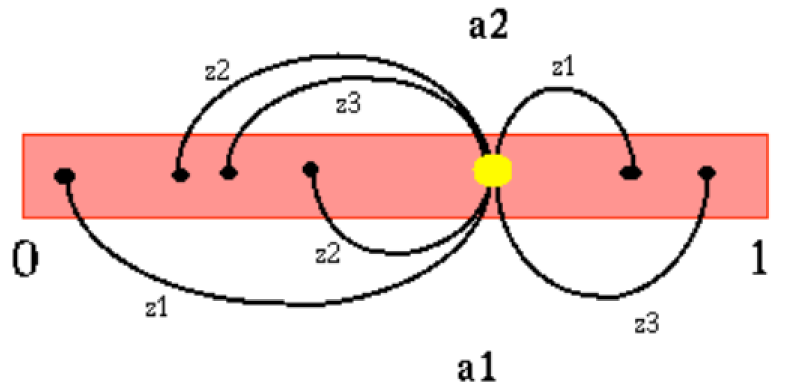
- But what if I don't know what state I'm in?

  - Rather than maintaining a "current state", maintain a **belief distribution**
    - A distribution over states indicating the probability that I think I'm in each one
  - By taking an **action** in a **state**, I receive **observations** that tell me about the state I just entered

Partially Observable MDPs (POMDPs)

- Traditional MDPs are defined with:
  - States – S = {(0,0), (0,1), …}
  - Actions – A = {move_north, …}
  - Rewards – R(s,a,s') -> Reward
  - Transition Probabilities – T(s,a,s') -> P(s'| s, a)
- Now we have to add:
  - Observation set – $\mathrm{O} = \{o_1, o_2, \dots\}$
  - Observation prob. – $\Omega = P(o_1, \dots, o_i \mid S)$
- Also have to augment:
  - Current State (now *belief*) – B = [0.1, 0.6, 0.2, 0.1, …]
  - Policy *(no longer $S \to A$)* – $\pi: B \to A$

POMDP: Trivial Example



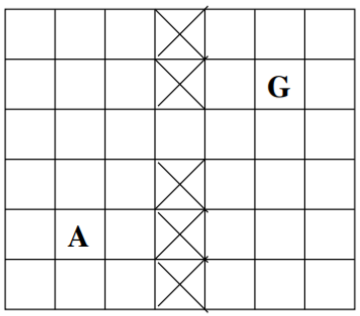
Two states: {0,1} Two Actions: {a1, a2}

Three Observations: {z1, z2, z3}

- A dot's position in the red bar indicates our belief over these states. (Yellow is current belief)
- B = [p, 1-p] indicates $p\%$ chance of being in State 0, and $1 - p\%$ chance of being in State 1.
- Executing a1 and observing z3 tells us that we're very likely to be in State 1
- Executing a1 and observing z1 tells us that we're very likely to be in State 0

Identifying Types of Planning Problems

Agent **A** must reach **G**, moving one cell at a time in **known** map



- If actions deterministic and initial location known, planning problem is **Classical**
- If actions non-deterministic and location observable, it's an **MDP** or **FOND**
- If actions non-deterministic and location partially obs, **POMDP** or **Contingent**

Different combinations of uncertainty and feedback: diff problems, diff models

Planner is generic solver for instances of a particular model

Classical planners, MDP Planners, POMDP planners, . . .

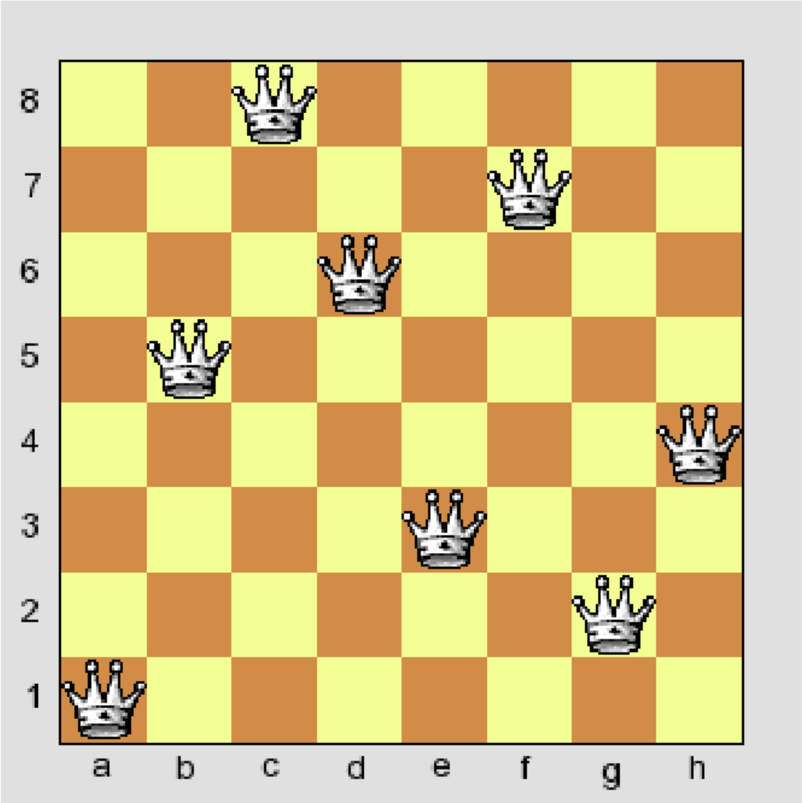
STRIPS: Language for Classical Planning

- A **problem** in Strips is a tuple $P = \langle F, O, I, G \rangle$:
  - F stands for set of all **atoms** (boolean vars)
  - O stands for set of all **operators** (actions)
  - $I \subseteq F$ stands for **initial situation**
  - $G \subseteq F$ stands for **goal situation**

- Operators $o \in O$ **represented** by
  - the **Add** list $Add(o) \subseteq F$
  - the **Delete** list $Del(o) \subseteq F$
  - the **Precondition** list $Pre(o) \subseteq F$

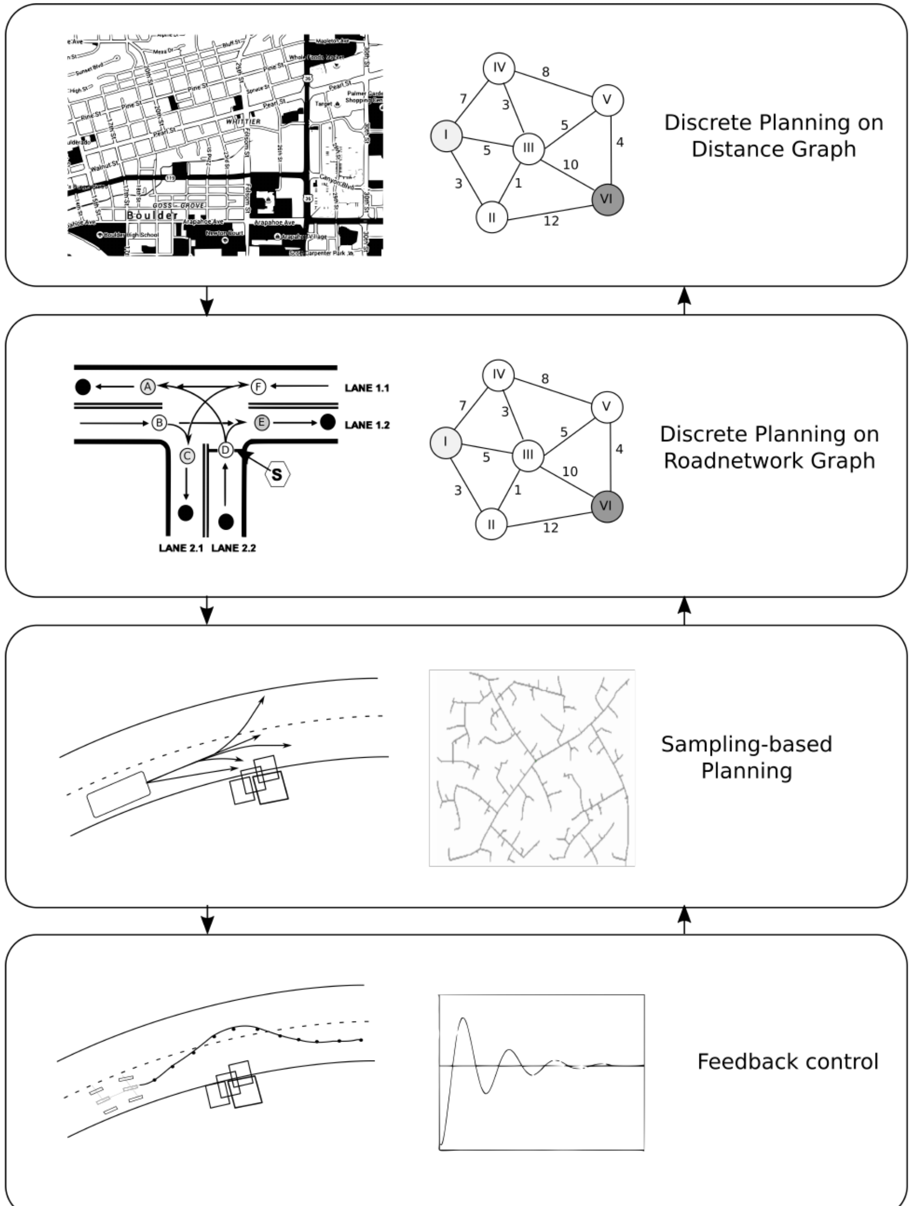
- Pickup(X)
  - P: grip(∅) ∧ clear(X) ∧ ontable(X)
  - A: grip(X)
  - D: onTable(X) ∧ grip(∅)

Problem Formulation Matters!
The 8 Queens Problem



- Formulation #1:
  - Place a queen on any open square
  - Repeat until all queens are placed
  - State space of $\frac{64!}{56!} = 1.78 * 10^{14}$
- Formulation #2:
  - Place a queen on any row 1 square
  - Place a queen on any row 2 square
  - State space of $8^8 = 1.68 * 10^7$
- Formulation #3:
  - Place a queen on any row 1 square
  - Place a queen on any row 2 square not sharing a column…
  - State space of $8! = 40{,}320$

Planning across length scales



Short Quiz: 5 minutes

Paper Talks!

<u>Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective</u>

<u>Akgun et al.</u>

PRO: Ryan Leonard
CON: Shivendra Agrawal

Planning human-aware motions using a sampling-based costmap planner

Jim Mainprice et al.

Paper Talks!

Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective

Akgun et al.

<u>Planning human-aware motions using a sampling-based costmap planner</u>

<u>Jim Mainprice et al.</u>

PRO: Jack Kawell
CON: Shruthi Sukumar

Project Ideas List is Online Now

- Fill out by 10pm tonight!
  - You aren’t bound to projects you pick/add, it is meant to be a starting point for project planning.
- Everyone will pitch a project idea on Tuesday!
  - Submit the idea you intend to pitch on Moodle by Monday
  - Groups will be formed during class after pitches
- Project meetings available tomorrow and Monday
  - Happy to meet with anyone looking for additional project topic guidance, please set up an appointment by e-mail. Bradley.Hayes@Colorado.edu

Short Presentation: 5-8 minutes

- Motivate the problem that your solution aims to solve
- Introduce the platforms and sensors you require
- Describe the proposed solution (high level terms ok!)
- Describe how you can evaluate success or failure

Aim high! We can work together to figure out graceful failure modes.

(I also highly encourage picking a compelling team name, and having fun with the proposal presentation)