16-782 Fall’23
Planning & Decision-making in Robotics

Introduction;
What is Planning, Role of Planning in Robots

Maxim Likhachev
Robotics Institute
Carnegie Mellon University
Class Logistics

• Instructor:
  Maxim Likhachev – maxim@cs.cmu.edu

• TA:
  Abigail Breitfeld - abreitfe@andrew.cmu.edu
  Tejus Gupta – tejusg@andrew.cmu.edu

• Website:
  http://www.cs.cmu.edu/~maxim/classes/robotplanning_grad

• Announcements, Questions, Recorded Lectures:
  - on Piazza
  - should have received an email with access info
About Me

- **My Research Interests:**
  - Planning, Decision-making, Learning
  - Applications: planning for complex robotic systems including aerial and ground robots, manipulation platforms, small teams of heterogeneous robots

- More info: [http://www.cs.cmu.edu/~maxim](http://www.cs.cmu.edu/~maxim)
- Search-based Planning Lab: [http://www.sbpl.net](http://www.sbpl.net)
About Me

• Also, currently split between CMU and Waymo, where I’m heavily involved in planning for self-driving vehicles
Class Objectives at High-level

- Understand and learn how to implement most popular planning and decision-making approaches in robotics
- Understand the challenges and basic approaches to interleaving planning and execution in robotic systems
- Learn common uses of planning/decision-making in robotics
- Get a sense for doing research in the area of planning/decision-making in robotics
What is Planning?

• According to Wikipedia: “Planning is the process of thinking about an organizing the activities required to achieve a desired goal.”
What is Planning for Robotics?

• According to Wikipedia: “Planning is the process of thinking about an organizing the activities required to achieve a desired goal.”

• Given
  – model (states and actions) of the robot(s) $M^R = \langle S^R, A^R \rangle$
  – a model of the world $M^W$
  – current state of the robot $s^R_{current}$
  – current state of the world $s^W_{current}$
  – cost function $C$ of robot actions
  – desired set of states for robot and world $G$

• Compute a plan $\pi$ that
  – prescribes a set of actions $a_1, \ldots a_K$ in $A^R$ the robot should execute
  – reaches one of the desired states in $G$
  – (preferably) minimizes the cumulative cost of executing actions $a_1, \ldots a_K$
Few Examples

• Given
  – model (states and actions) of the robot(s) \( M^R = \langle S^R, A^R \rangle \)
  – a model of the world \( M^W \)
  – current state of the robot \( s^R_{\text{current}} \)
  – current state of the world \( s^W_{\text{current}} \)
  – cost function \( C \) of robot actions
  – desired set of states for robot and world \( G \)

• Compute a plan \( \pi \) that
  – prescribes a set of actions \( a_1, \ldots, a_K \) in \( A^R \) the robot should execute
  – reaches one of the desired states in \( G \)
  – (preferably) minimizes the cumulative cost of executing actions \( a_1, \ldots, a_K \)

Planning for omnidirectional robot:

What is \( M^R \)?
What is \( M^W \)?
What is \( s^R_{\text{current}} \)?
What is \( s^W_{\text{current}} \)?
What is \( C \)?
What is \( G \)?
Few Examples

• **Given**
  – model (states and actions) of the robot(s) \( M^R = \langle S^R, A^R \rangle \)
  – a model of the world \( M^W \)
  – current state of the robot \( s^R_{\text{current}} \)
  – current state of the world \( s^W_{\text{current}} \)
  – cost function \( C \) of robot actions
  – desired set of states for robot and world \( G \)

• **Compute a plan \( \pi \) that**
  – prescribes a set of actions \( a_1, \ldots, a_K \) in \( A^R \) the robot should execute
  – reaches one of the desired states in \( G \)
  – (preferably) minimizes the cumulative cost of executing actions \( a_1, \ldots, a_K \)

Planning for omnidirectional drone:

*What is \( M^R \)?*
*What is \( M^W \)?*
*What is \( s^R_{\text{current}} \)?*
*What is \( s^W_{\text{current}} \)?*
*What is \( C \)?*
*What is \( G \)?*

*MacAllister et al., 2013*
Few Examples

• **Given**
  – model (states and actions) of the robot(s) $M^R = <S^R, A^R>$
  – a model of the world $M^W$
  – current state of the robot $s^R_{current}$
  – current state of the world $s^W_{current}$
  – cost function $C$ of robot actions
  – desired set of states for robot and world $G$

• **Compute a plan $\pi$ that**
  – prescribes a set of actions $a_1, \ldots, a_K$ in $A^R$ the robot should execute
  – reaches one of the desired states in $G$
  – (preferably) minimizes the cumulative cost of executing actions $a_1, \ldots, a_K$

Planning for autonomous navigation:

What is $M^R$?
What is $M^W$?
What is $s^R_{current}$?
What is $s^W_{current}$?
What is $C$?
What is $G$?

Likhachev & Ferguson, ‘09; part of Tartanracing team from CMU for the Urban Challenge 2007 race
Few Examples

• **Given**
  - model (states and actions) of the robot(s) \( M^R = <S^R, A^R> \)
  - a model of the world \( M^W \)
  - current state of the robot \( s^R_{current} \)
  - current state of the world \( s^W_{current} \)
  - cost function \( C \) of robot actions
  - desired set of states for robot and world \( G \)

• **Compute a plan \( \pi \) that**
  - prescribes a set of actions \( a_1, ..., a_K \) in \( A^R \) the robot should execute
  - reaches one of the desired states in \( G \)
  - (preferably) minimizes the cumulative cost of executing actions \( a_1, ..., a_K \)

Planning for autonomous flight among people : \( \text{Narayanan et al., 2012} \)

**What is \( M^R \)?**
**What is \( M^W \)?**
**What is \( s^R_{current} \)?**
**What is \( s^W_{current} \)?**
**What is \( C \)?**
**What is \( G \)?**
Few Examples

• Given
  – model (states and actions) of the robot(s) $M^R = <S^R, A^R>$
  – a model of the world $M^W$
  – current state of the robot $s^R_{current}$
  – current state of the world $s^W_{current}$
  – cost function $C$ of robot actions
  – desired set of states for robot and world $G$

• Compute a plan $\pi$ that
  – prescribes a set of actions $a_1, ..., a_K$ in $A^R$ the robot should execute
  – reaches one of the desired states in $G$
  – (preferably) minimizes the cumulative cost of executing actions $a_1, ..., a_K$

Planning for a mobile manipulator robot opening a door:  

What is $M^R$?
What is $M^W$?
What is $s^R_{current}$?
What is $s^W_{current}$?
What is $C$?
What is $G$?

Gray et al., 2013

SBPL (Search-based Planning)

autonomous opening of a spring-loaded door with full body
Few Examples

- **Given**
  - model (states and actions) of the robot(s) $M^R = \langle S^R, A^R \rangle$
  - a model of the world $M^W$
  - current state of the robot $s^R_{current}$
  - current state of the world $s^W_{current}$
  - cost function $C$ of robot actions
  - desired set of states for robot and world $G$

- **Compute a plan $\pi$ that**
  - prescribes a set of actions $a_1, \ldots, a_K$ in $A^R$ the robot should execute
  - reaches one of the desired states in $G$
  - (preferably) minimizes the cumulative cost of executing actions $a_1, \ldots, a_K$

Planning for a mobile manipulator robot assembling a birdcage:  Cohen et al., 2015

*What is $M^R$?*
*What is $M^W$?*
*What is $s^R_{current}$?*
*What is $s^W_{current}$?*
*What is $C$?*
*What is $G$?*
• **Given**
  – model (states and actions) of the robot(s) $M^R = <S^R, A^R>$
  – a model of the world $M^W$
  – current state of the robot $s^R_{\text{current}}$
  – current state of the world $s^W_{\text{current}}$
  – cost function $C$ of robot actions
  – desired set of states for robot and world $G$

• **Compute a plan $\pi$ that**
  – prescribes a set of actions $a_1, \ldots a_K$ in $A^R$ the robot should execute
  – reaches one of the desired states in $G$
  – (preferably) minimizes the cumulative cost of executing actions $a_1, \ldots a_K$

Planning/decision-making for a mobile manipulator unloading a truck:

*What is $M^R$?*
*What is $M^W$?*
*What is $s^R_{\text{current}}$?*
*What is $s^W_{\text{current}}$?*
*What is $C$?*
*What is $G$?*
Assuming Infinite Computational Resources…

*Where does Planning break?*
Assuming Infinite Computational Resources…

Where does Planning break?

Reliance on the knowledge/accuracy of the model!

Role of Learning in Planning?
Planning vs. Learning

Model-based approach

Learning models $M^R$, $M^W$ and cost function $C$

Planning using models $M^R$, $M^W$ and cost function $C$

Model-free approach

Learning the mapping from “what robot sees” onto “what to do next” using rewards received by the robot (Reinforcement Learning) or demonstrations (Behavior Cloning)
Planning within a Typical Autonomy Architecture

- **Planning**: What do I do next?
- **Plan Execution/Controller**: How do I do the next action?
- **Perception**: What do I see?
- **Localization**: Where am I?

Connections:
- **Planning** to **Plan Execution/Controller**
- **Plan Execution/Controller** to **Perception**
- **Plan Execution/Controller** to **Localization**
- **Perception** to **feedback from sensors**
- **Localization** to **feedback from actuators**

Commands flow from Plan Execution/Controller to Perception and Localization.
Planning vs. Trajectory Following vs. Control

local planning (trajectory following)

global planning

controller

Images from wikipedia
Class Logistics

• Books (optional):

- Planning Algorithms by Steven M. LaValle

- Heuristic Search, Theory and Applications by Stefan Edelkamp and Stefan Schroedl

- Principles of Robot Motion, Theory, Algorithms, and Implementations by Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki and Sebastian Thrun

- Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
Class Prerequisites

• Knowledge of programming (e.g., C, C++)

• Working knowledge of data structures & basic Computer Science algorithms (e.g., graphs, linked lists, priority queues, BFS/DFS, etc.)

• Prior exposure to robotics
Class Objectives

- Understand and learn how to implement most popular planning algorithms in robotics including heuristic search-based planning algorithms, sampling-based planning algorithms, task planning, planning under uncertainty and multi-robot planning
- Learn basic principles behind the design of planning representations
- Understand core theoretical principles that many planning algorithms rely on and learn how to analyze theoretical properties of the algorithms
- Understand the challenges and basic approaches to interleaving planning and execution in robotic systems
- Learn common uses of planning in robotics
- Get a sense for doing research in the area of planning/decision-making in robotics
## Tentative Schedule for Planning and Decision-making in Robotics Class

### Fall 2023

<table>
<thead>
<tr>
<th>Date</th>
<th>Day</th>
<th>Topic</th>
<th>HW out</th>
<th>HW due</th>
</tr>
</thead>
<tbody>
<tr>
<td>28-Aug</td>
<td>Mon</td>
<td>Introduction; What is Planning?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-Aug</td>
<td>Wed</td>
<td>planning representations: explicit vs. implicit graphs, skeletonization, cell decomposition &amp; lattice-based graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-Sep</td>
<td>Mon</td>
<td>LABOR DAY - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-Sep</td>
<td>Wed</td>
<td>search algorithms: A*, Multi-goal A*, Weighted A*, Backward A*</td>
<td></td>
<td>HW1</td>
</tr>
<tr>
<td>11-Sep</td>
<td>Mon</td>
<td>search algorithms: Heuristic functions, Multi-Heuristic A*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13-Sep</td>
<td>Wed</td>
<td>interleaving planning and execution: Anytime heuristic search, Incremental heuristic search</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-Sep</td>
<td>Mon</td>
<td>interleaving planning and execution: Real-time heuristic Search</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-Sep</td>
<td>Wed</td>
<td>case study: planning for autonomous driving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-Sep</td>
<td>Mon</td>
<td>planning representations: PRM for continuous spaces</td>
<td></td>
<td>HW1</td>
</tr>
<tr>
<td>27-Sep</td>
<td>Wed</td>
<td>planning representations/search algorithms: RRT, RRT-Connect, RRT*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-Oct</td>
<td>Mon</td>
<td>case study: planning for mobile manipulators and legged robots</td>
<td></td>
<td>HW2</td>
</tr>
<tr>
<td>4-Oct</td>
<td>Wed</td>
<td>search algorithms: Markov Property, dependent vs. independent variables, Dominance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-Oct</td>
<td>Mon</td>
<td>case study: planning for coverage, mapping and surveillance tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-Oct</td>
<td>Wed</td>
<td>planning representations: state-space vs. symbolic representation for task planning</td>
<td></td>
<td>HW2</td>
</tr>
<tr>
<td>16-Oct</td>
<td>Mon</td>
<td>FALL BREAK - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-Oct</td>
<td>Wed</td>
<td>FALL BREAK - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-Oct</td>
<td>Mon</td>
<td>search algorithms: planning on symbolic representations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-Oct</td>
<td>Wed</td>
<td>planning under uncertainty: Minimax formulation, Minimax Backward A*</td>
<td></td>
<td>HW3</td>
</tr>
<tr>
<td>30-Oct</td>
<td>Mon</td>
<td>planning under uncertainty: Markov Decision Processes, Value Iteration, RTDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-Nov</td>
<td>Wed</td>
<td>final project proposal presentations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-Nov</td>
<td>Mon</td>
<td>planning under uncertainty: Markov Decision Processes, Value Iteration, RTDP (cont'd)</td>
<td></td>
<td>HW3</td>
</tr>
<tr>
<td>8-Nov</td>
<td>Wed</td>
<td>planning under uncertainty: Partially-Observable Markov Decision Processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13-Nov</td>
<td>Mon</td>
<td>planning under uncertainty: Partially-Observable Markov Decision Processes (cont'd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-Nov</td>
<td>Wed</td>
<td>exam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-Nov</td>
<td>Mon</td>
<td>TBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-Nov</td>
<td>Wed</td>
<td>THANKSGIVING - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-Nov</td>
<td>Mon</td>
<td>multi-robot planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29-Nov</td>
<td>Wed</td>
<td>multi-robot planning (cont'd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-Dec</td>
<td>Mon</td>
<td>learning in planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-Dec</td>
<td>Wed</td>
<td>final project presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Three Homworks + Final Project

• **All homeworks are individual** (no groups)

• Final projects is a group project (3-5 people per group)

• Homeworks are programming assignments based on the material

• Final project is a research-like project
  - For example: to develop and implement a planner for a robot planning problem of your choice
  - Or: to extend a particular planning algorithm to improve its running time or to handle additional conditions

  - Two presentations (proposal and final) and meetings with groups
Class Structure

• Grading

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three homeworks</td>
<td>33%</td>
</tr>
<tr>
<td>Exam</td>
<td>20%</td>
</tr>
<tr>
<td>In-class pop quizzes</td>
<td>10%</td>
</tr>
<tr>
<td>Final project</td>
<td>32%</td>
</tr>
<tr>
<td>Participation</td>
<td>5%</td>
</tr>
</tbody>
</table>

• Exam is tentatively scheduled for Nov. 15

• Late Policy
  - 3 free late days
  - No late days may be used for the final project!
  - Each additional late day will incur a 10% penalty
Questions about the class?