16-350 Spring’20
Planning Techniques for Robotics

Introduction;
What is Planning for Robotics?

Maxim Likhachev
Robotics Institute
Carnegie Mellon University
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)
Class Logistics

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

• TA:
  Navyata Sanghvi – nsanghvi@andrew.cmu.edu

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

• Mailing List for Announcements and Questions:
  Will be set it up shortly
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++)

• Knowledge of data structures

• Some prior exposure to robotics (e.g., Intro to Robotics class) is preferred
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
## Tentative Class Schedule

<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>13-Jan</td>
<td>Mon</td>
<td>Introduction: What is Planning?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-Jan</td>
<td>Wed</td>
<td>planning representations: grid-based graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-Jan</td>
<td>Mon</td>
<td>MLK DAY - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-Jan</td>
<td>Wed</td>
<td>NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-Jan</td>
<td>Mon</td>
<td>search algorithms: Uninformed A*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29-Jan</td>
<td>Wed</td>
<td>search algorithms: A*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-Feb</td>
<td>Mon</td>
<td>heuristics, weighted A*, Backward A*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-Feb</td>
<td>Wed</td>
<td>interleaving planning and execution: Anytime heuristic search</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-Feb</td>
<td>Mon</td>
<td>interleaving planning and execution: Freespace assumption, Incremental heuristic search</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-Feb</td>
<td>Wed</td>
<td>interleaving planning and execution: Limited Horizon search, LRTA*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-Feb</td>
<td>Mon</td>
<td>planning representations: lattice-based graphs, explicit vs. implicit graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19-Feb</td>
<td>Wed</td>
<td>case study: planning for autonomous driving</td>
<td></td>
<td>HW1</td>
</tr>
<tr>
<td>24-Feb</td>
<td>Mon</td>
<td>planning representations: PRM for continuous spaces</td>
<td></td>
<td>HW1</td>
</tr>
<tr>
<td>26-Feb</td>
<td>Wed</td>
<td>planning representations/search algorithms: RRT, RRT-Connect, RRT*</td>
<td></td>
<td>HW2</td>
</tr>
<tr>
<td>2-Mar</td>
<td>Mon</td>
<td>planning representations/search algorithms: RRT, RRT-Connect, RRT* (cont'd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-Mar</td>
<td>Wed</td>
<td>case study: planning for mobile manipulation and articulated robots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-Mar</td>
<td>Mon</td>
<td>SPRING BREAK - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-Mar</td>
<td>Wed</td>
<td>SPRING BREAK - NO CLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16-Mar</td>
<td>Mon</td>
<td>search algorithms: Multi-goal A*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-Mar</td>
<td>Wed</td>
<td>case study: planning for exploration and surveillance tasks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23-Mar</td>
<td>Mon</td>
<td>search algorithms: Markov Property, dependent vs. independent variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-Mar</td>
<td>Wed</td>
<td>planning representations: state-space vs. symbolic representation for task planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-Mar</td>
<td>Mon</td>
<td>search algorithms: symbolic task planning algorithms</td>
<td></td>
<td>HW2</td>
</tr>
<tr>
<td>1-Apr</td>
<td>Wed</td>
<td>final project proposal presentations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-Apr</td>
<td>Mon</td>
<td>planning under uncertainty: Minimax formulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-Apr</td>
<td>Wed</td>
<td>planning under uncertainty: Expected Cost Minimization formulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13-Apr</td>
<td>Mon</td>
<td>planning under uncertainty: Solving Markov Decision Processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-Apr</td>
<td>Wed</td>
<td>planning under uncertainty: Solving Markov Decision Processes (cont'd)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20-Apr</td>
<td>Mon</td>
<td>exam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-Apr</td>
<td>Wed</td>
<td>multi-robot planning: centralized planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27-Apr</td>
<td>Mon</td>
<td>multi-robot planning: decentralized planning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29-Apr</td>
<td>Wed</td>
<td>final project presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Class Structure

• Grading

<table>
<thead>
<tr>
<th>Assignment</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 April 20 (no final exam)

• 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 with 50% being the upper limit on the penalty
Three Homeworks + Final Project

• All homeworks are individual (no groups)

• Final project are in groups of 2-3 students

• Homeworks are programming assignments

• Final project is a research-like project. For example:
  - to develop a planner for a robot planning problem of your choice
  - to extend an existing or develop a new planning algorithm
  - to prove novel properties of a planning algorithm

  - Get a feel for doing research: Individual meetings with groups, Two class presentations (initial idea and final)
Three Homeworks + Final Project

• Homework assignments for Masters students will have additional scope

• Undergraduate students will have an option to tackle this additional scope and receive bonus points
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 $s^R_{\text{current}}$
  – 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 $a_1, \ldots a_K$
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 \( s^{R\text{current}} \)
  – 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 \( G \)?
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?

*feedback from sensors*

*feedback from actuators*

*plan*
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 \( s^R_{\text{current}} \)
  – 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 $a_1, \ldots a_K$ 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 flight among people:  

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 $G$?*
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 a mobile manipulator robot assembling a birdcage:  
*Cohen et al., 2015*

*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 \( s^R_{current} \)
  – 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 unloading a truck:

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

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)
Planning vs. Trajectory Following vs. Control

local planning (trajectory following)

global planning

controller

Images from wikipedia
Questions about the class?