Planning for Autonomous Robots

Professor Nick Hawes, Oxford Robotics Institute, University of Oxford.

This course will cover Artificial Intelligence (AI) techniques that can be used to plan behaviour for autonomous robots, particularly mobile robots.

Part 1: Problem Solving as Search

This part will introduce the basics ideas behind problem solving as search, one of the fundamental techniques in AI. It will discuss how to abstract a problem into a state space, and then introduce a number of algorithms for searching for solutions in that state space. This part will draw heavily on shortest path route planning problems, and in this context introduce uniformed and informed methods to solve them. We will also briefly cover some other types of route planning problems (e.g travelling salesperson and its variants).

Part 2: Deterministic Planning

The second part of this course will introduce a wider variety of behaviour planning problems, and show how they can be abstracted into first order predicate logic. We will then look at a range of algorithms for efficiently solving problems encoded in this way. This will cover the PDDL/STRIPS formalism, plus algorithms that operate it using forward and backward search, plus partial order planning. We will also briefly cover some other types of planning approaches, such as hierarchical methods and reactive planning.

Part 3: Planning Under Uncertainty

The previous two parts of the course assume that actions performed by the planning agent always achieve the desired effect. However, in robotics this is often not the case. Therefore this part of the course moves to planning under uncertainty. We now formulate planning problems as Markov decision processes (MDPs), which capture some of the uncertainty an autonomous mobile robot may experience when executing plans. We look at how to use MDPs to formulate planning problems as Stochastic Shortest Path problems (SSPs), and look at algorithms for solving SSPs to produce policies.

Part 4: Planning in Continuous Space

Up to now we have discretised the space within which the robot is operating, leading to a discrete state space. Many robotics problems require operation in continuous space. In this final part of the course we look at sampling-based methods for dynamically discretising and planning routes in continuous space, including probabilistic road maps and rapidly exploring random trees.

Suggested Research Projects

1. Choose a collection of search algorithms and implement them in a language of your choosing. Compare their theoretical properties (e.g. time and space complexity, completeness, optimality etc.) to the performance observed in practice on a set of benchmark problems of your choosing.
2. Develop a PDDL domain for a household robot, research a PDDL planner which can solve it, download the planner (if available) and apply it to instances of problems from your domain.
3. Implement a grid world SSP problem, represent it as an MDP, and build an MDP solver using value iteration, policy iteration, or another approach to solve it.
4. Sampling-based planners struggle to solve problems with narrow gaps. Research solutions to this problem. Implement a sampling-based planner to solve 2D robot navigation problems and extend it with a solution to the narrow gap problem.

More Advanced

5. In a multi-agent travelling salesperson problem (TSP) you have to choose which agent visits which city, and the order in which each agent makes their visit. Research then implement a solution to this problem, assuming real-valued positive edge weights. How does your solution scale with numbers of agents and cities? How can you solve large instances by trading off solution quality for solution time?
6. Use data from a real world source to create planning problems (either deterministic or non-deterministic) and use one of the algorithms you've learned about to solve it. Examples could be driving or walking route planning from online maps, playing card/board/video games, or multi-mode travel planning (i.e. using car, train and bus for a single route).
7. Develop a PDDL domain for a voice assistant that is able to solve multi-step problems by asking questions, and querying online databases. Here you should represent facts that need to be known by the system in order to answer the question. This is a type of modal logic and requires careful handling when planning.
8. Build an MDP of a problem featuring uncertain durations of actions. For example an underwater robot moving through an ocean with different currents/weather patterns, or a mobile robot in a shopping mall. Create and solve MDPs that require an agent to reach a destination by a given time. This will require handling of deadlines.