

Optimizing a Search

The Big Ideas:

- Basic Search can be augmented using costs and heuristics
- Costs reflect the expenditure associated with a certain transition
- Heuristics reflect an estimation of the total remaining cost to the goal

Introduction

Last week, we introduced the concept of search. We looked at the 6.01 implementation of a general search algorithm, addressed issues of common sense and dynamic programming, and localized a robot.

This week, we focus on improving search; we can systematically use information we have about the state space we're searching, in order to save us time and space.

Vocabulary

In order to engage the material, be able to communicate about the topic with others, and in particular ask questions, we encourage familiarity with the following terms:

Theory

- Cost
- Heuristic
- Consistency
- Admissibility

Practice

- `ucSearch`
- `A*`

Check Yourself

Theory: you should understand:

- The relationship between costs and heuristics
- Why not to use non-admissible heuristics
-

Practice: you should be able to:

- Identify the number of states visited and expanded as a consequence of running a particular search algorithm.
- Understand how the components we have built over the module combine in Design Lab.

Resources

Theory:

Chapter 8 of the 6.01 Course Notes is the official assigned reading for this week. In particular, the latter half concerning State Estimation is relevant to this week's assignments.

Practice:

The 6.01 Software Documentation will come in handy, in particular the modules `gridMap` and `util.make*Fill`.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.