Optimality & Constraints in the Open Motion Planning Library

Mark Moll
mark@picknik.ai | http://moll.ai
Director of Research, PickNik Robotics
Sr. Research Scientist, Rice University
Outline

● OMPL overview
● Planning with costs
● Planning with constraints
● Planning with multiple goals
● Summary
Preface

OMPL Overview
OMPL: a generic motion planning library

- Focused on sampling-based algorithms
- Does not contain representations of robots / environments
- Higher-level robotics software framework is needed for OMPL to be practically useful:
OMPL metrics

- > 2,600 registered users (many more get OMPL from package managers or do not register)
- > 70,000 downloads
- > 800 citations
- OMPL web site since January 2011:
  - ~500,000 sessions
  - ~200,000 unique visitors
  - ~2,000,000 page views
Benchmarking

http://plannerarena.org
Part 1

Planning with costs
What is *optimal* motion?

- Shortest? (Using what distance metric?)
- Smoothest?
- Safest?

![Path Length](image1.png)
![Clearance](image2.png)
![Combination](image3.png)
General cost infrastructure

- Costs can be defined per state (waypoint) along a path (e.g., using a cost map)
- Additionally, the cost of the motion between waypoints can be defined (e.g., path length)
- The way costs are aggregated can be defined (sum, min, max, etc.)
- Terminal costs, admissible heuristics
- Many optimizing planners (with different optimality guarantees)
  - PRM*, RRT*, BIT*, FMT, SST, ...
- Path post-processing tools are (mostly) cost-aware
A counterintuitive way to optimize paths

1. Use a bag of planners to compute feasible (but suboptimal) solution paths
2. Simplify and hybridize paths (graft good partial path segments together)
3. Repeat until convergence / timeout (keep all solution paths from all previous iterations)

→ Converges faster than most planners that explicitly optimize!

More testing needed to select good portfolio of planners to get good trade-offs in (1) path quality, (2) compute time, (3) repeatability.

OMPL & other path optimization techniques

OMPL:
typically fast at finding feasible solutions, convergence to optimal is slow

Trajectory optimization (CHOMP, TrajOpt, GPMP2):
fast, but current implementations still fail often to produce feasible trajectories

OMPL + trajectory optimization: win-win!

Part 2

Planning with Constraints
Constrained motion planning

168 degrees of freedom
69 constraints

motion planned in ~14.5s.
NASA's Robonaut2 climbing inside the ISS
Part 3

Planning with Multiple Goals
Motivation

High-level specifications often have multiple valid interpretations.

Can translate interpretations to (sub)goals for motion planning.

Problem:
how to choose “good” goals:
feasible IK solutions exist and is reachable from current pose

**Approach**  
(for pick-and-place tasks):

Create implicit goal regions corresponding to all possible interpretations (e.g., end effector constraints, placement constraints).

Grow search tree, bias towards “best” goal state.

Initial goal cost can be based on heuristic (cost to come)

During planning, cost is adjusted through penalty-reward scheme based on success in expanding towards goal state.
Experiments

~50% faster in finding feasible solution compared to considering all goals equally likely.

“Pick up any of the blocks”

“Park in any of the green spaces”
<table>
<thead>
<tr>
<th>OMPL feature</th>
<th>Status in MoveIt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning with costs</td>
<td>Available now</td>
</tr>
<tr>
<td>Planning with constraints</td>
<td>Research code, significant work to make it a general purpose feature</td>
</tr>
<tr>
<td>Planning with multiple goals</td>
<td>Research code, no timeline yet for getting into MoveIt</td>
</tr>
</tbody>
</table>
Acknowledgements

Ioan Șucan  Lydia Kavraki  Julia Badger  Zak Kingston

Luis Torres  Bryce Willey  Ryan Luna  Juan David Hernández