

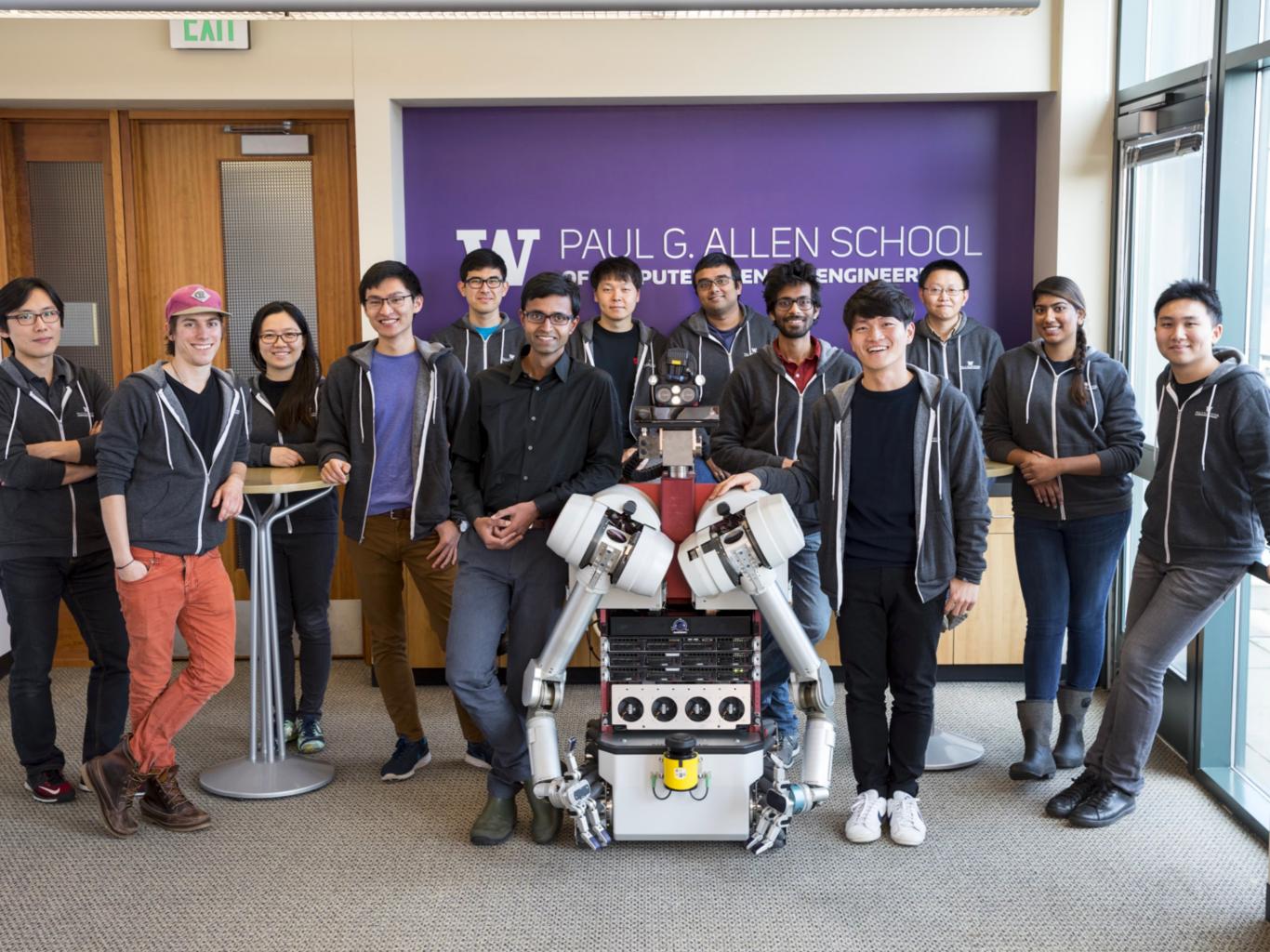
HERB

Carnegie Mellon University

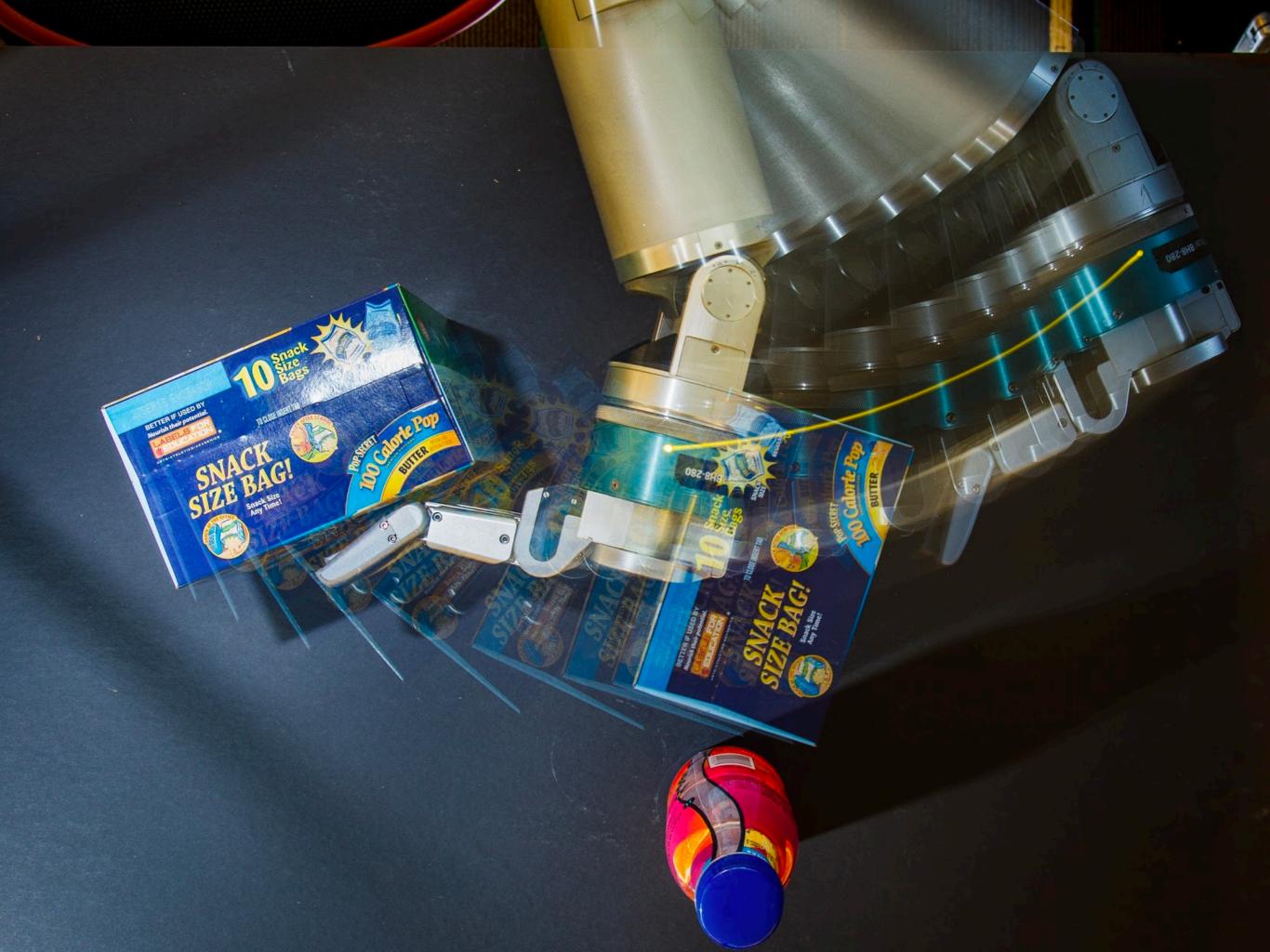
New Perspectives on the Piano Movers' Problem



Siddhartha Srinivasa Boeing Endowed Professor Personal Robotics Lab University of Washington



Manipulation



Personal Robotics Lab

Motion Planning is a technology

 $\leftarrow \rightarrow$ C (i) Not Secure | ompl.kavrakilab.org

OMPL Download Documentation - Gallery Code - Issues Community - About - Blog

🖈 🜔 🦞 🎙 🔁 🗌 🌚 🗄

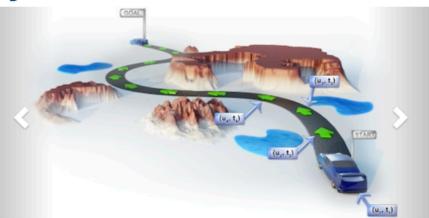
Q- Search

The Open Motion Planning Library

OMPL, the Open Motion Planning Library, consists of many state-of-the-art samplingbased motion planning algorithms. OMPL itself does not contain any code related to, e.g., collision checking or visualization. This is a deliberate design choice, so that OMPL is not tied to a particular collision checker or visualization front end. The library is designed so it can be easily integrated into systems that provide the additional needed components.

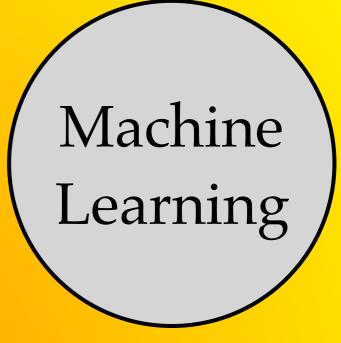
OMPL.app, the front-end for OMPL, contains a lightweight wrapper for the FCL and PQP collision checkers and a simple GUI based on PyQt / PySide. The graphical frontend can be used for planning motions for rigid bodies and a few vehicle types (firstorder and second-order cars, a blimp, and a quadrotor). It relies on the Assimp library to import a large variety of mesh formats that can be used to represent the robot and its environment.

Current version: 1.4.0Click for citation,Released: Jun 25, 2018if you use OMPL in your work



Formalizing the Core

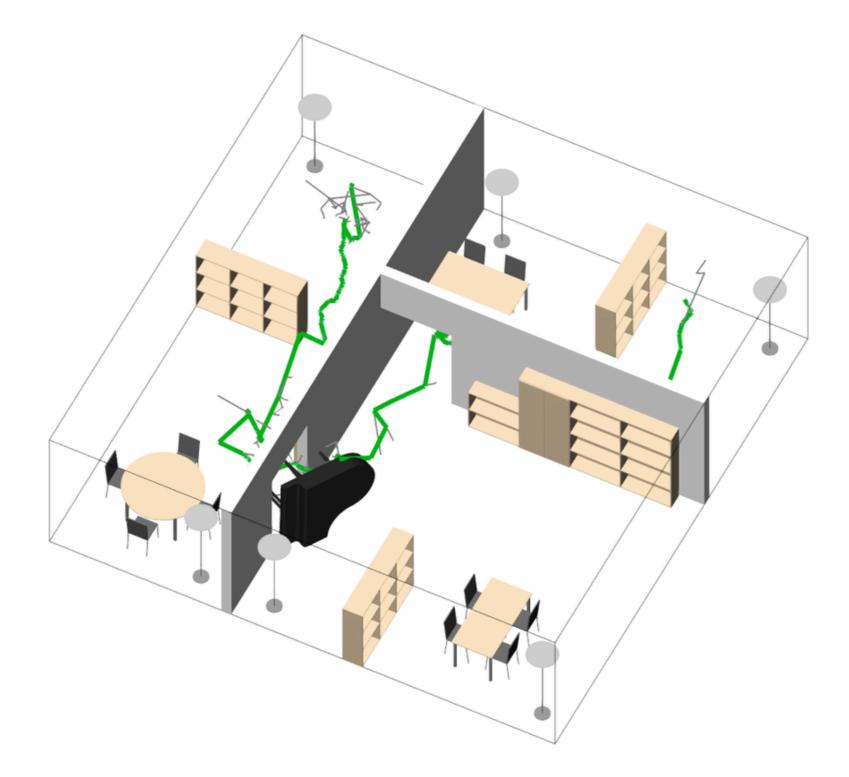




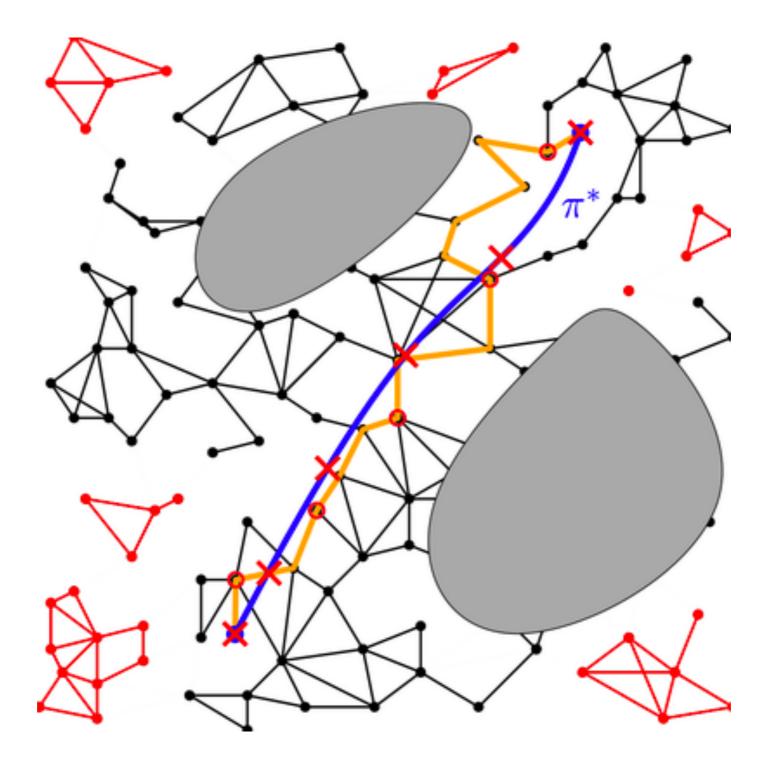
Formalizing the Core



The Piano Movers' Problem

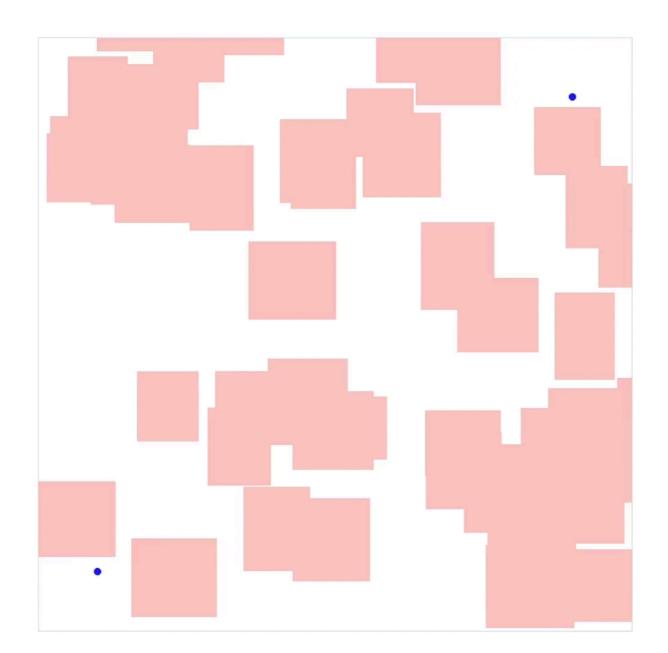






A* Search is Optimal ...

Exploits Optimal Substructure Best-first Search over Vertices

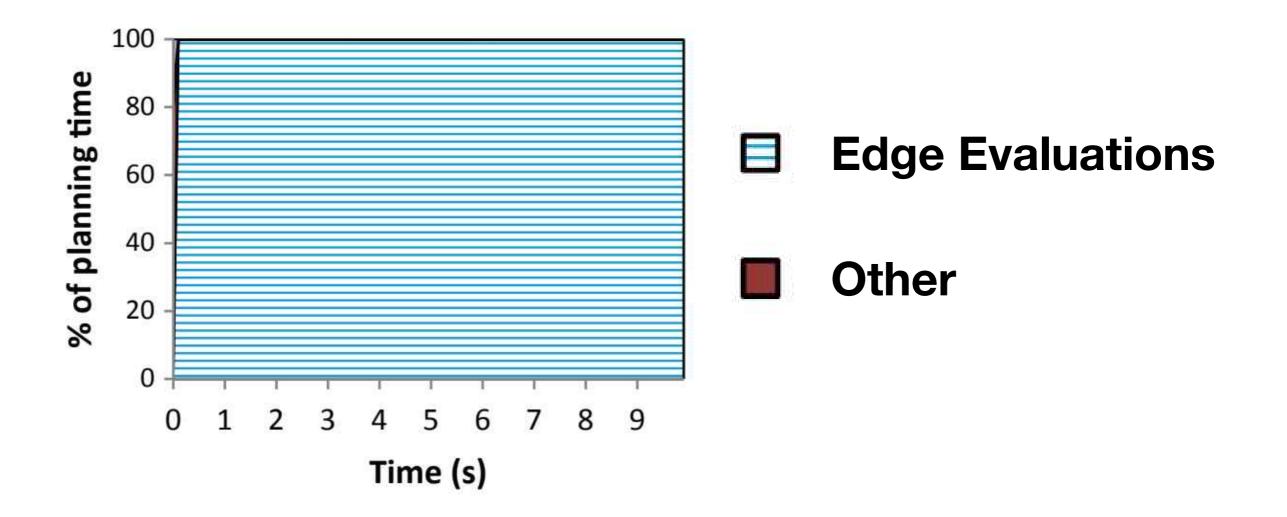


A* Search is Optimal ...

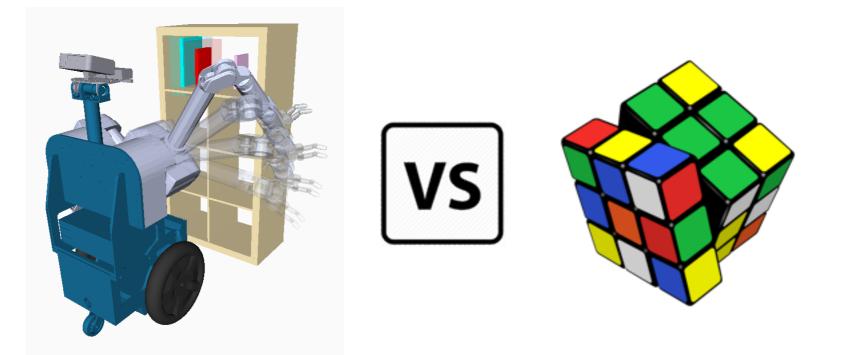
Expands the Fewest Number of Vertices

But is this what we *really* want in Motion Planning?

Edge Evaluation Dominates Planning Time



Hauser, Kris., Lazy collision checking in asymptotically-optimal motion planning. ICRA 2015



Edge Evaluation vs. Search Explicit vs. Implicit Graphs

Is there a Search Algorithm that Minimizes the Number of Edge Evaluations?

LazySP

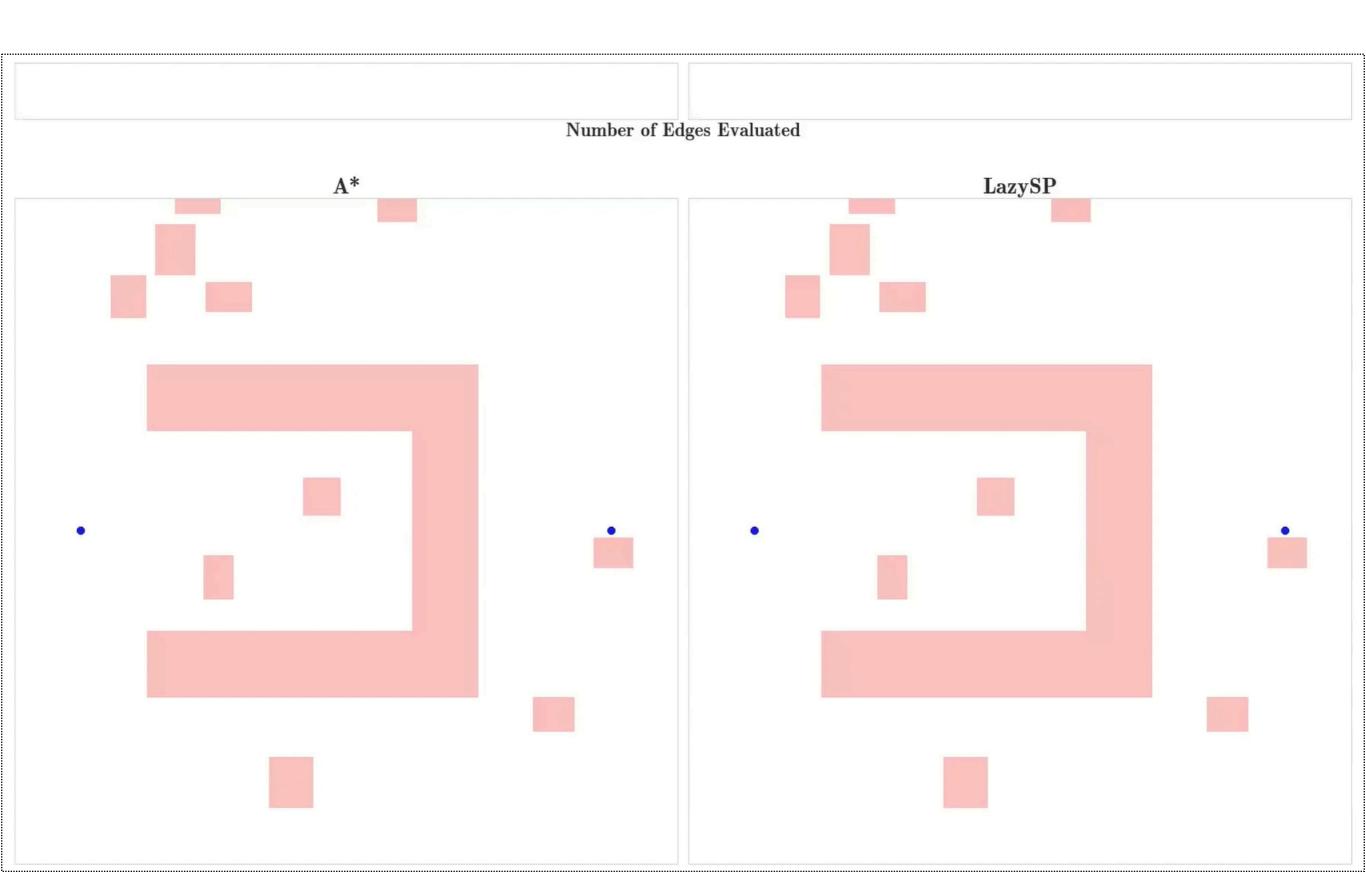
ICAPS 2018 [Best Conference Paper Award Winner]

First Provably Edge-Optimal A*-like Search Algorithm







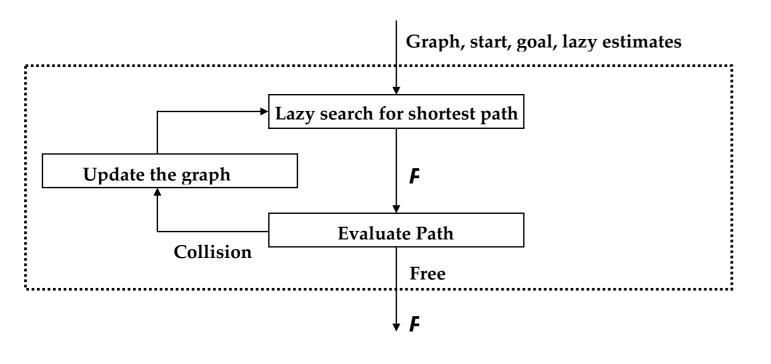


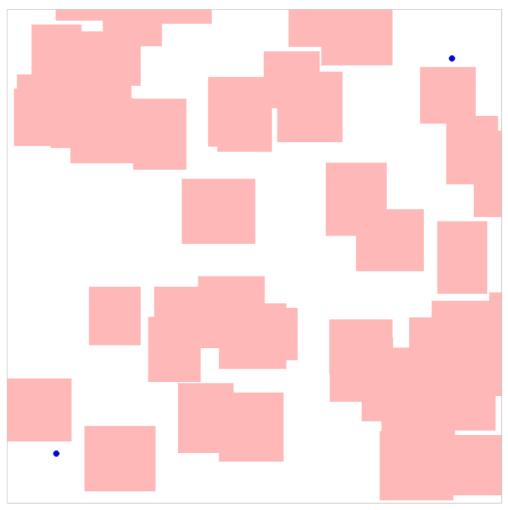
LazySP

Greedy Best-first Search over Paths

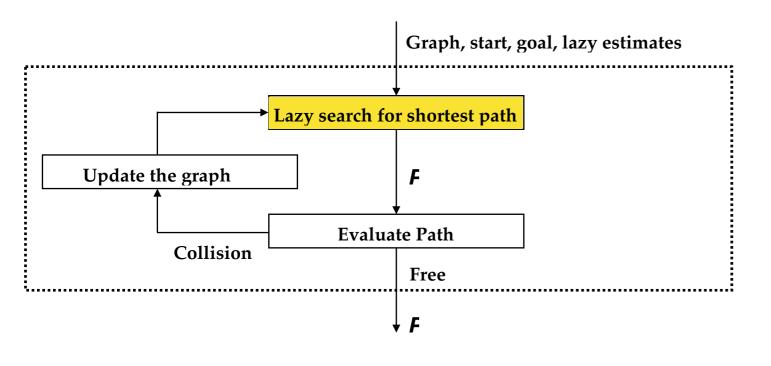
To find the shortest path, eliminate all shorter paths!

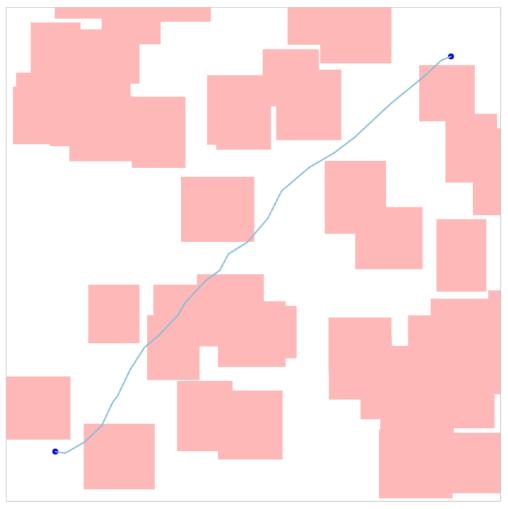




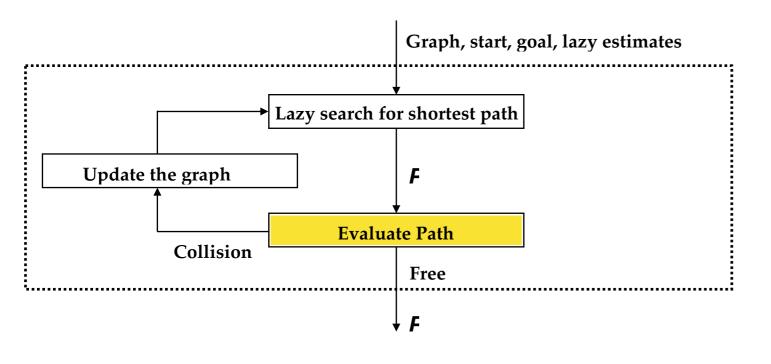


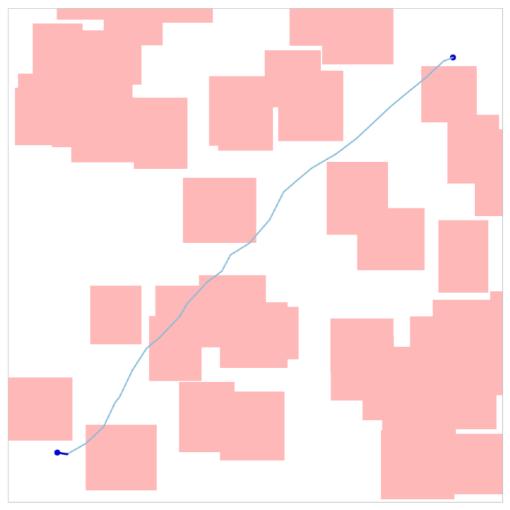




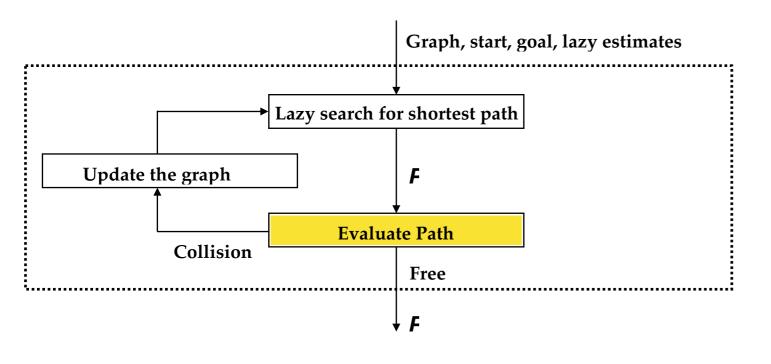


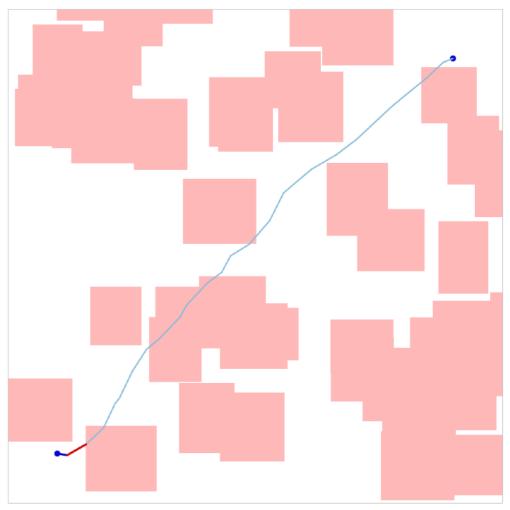




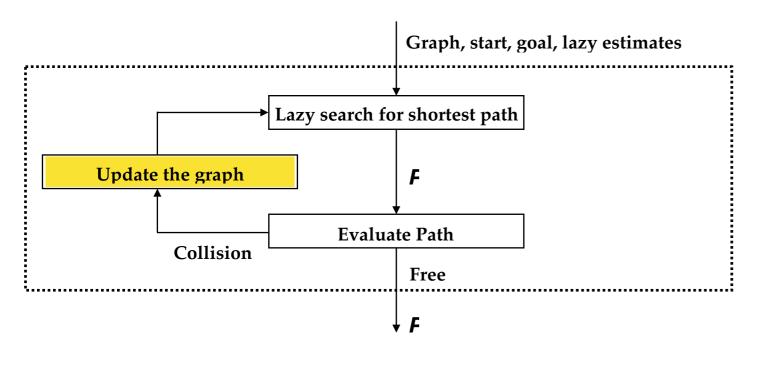


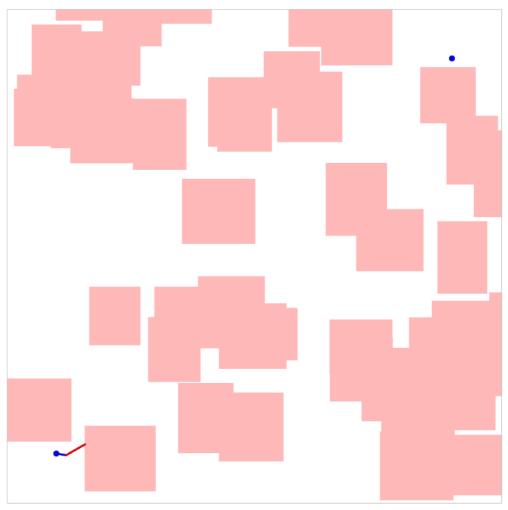




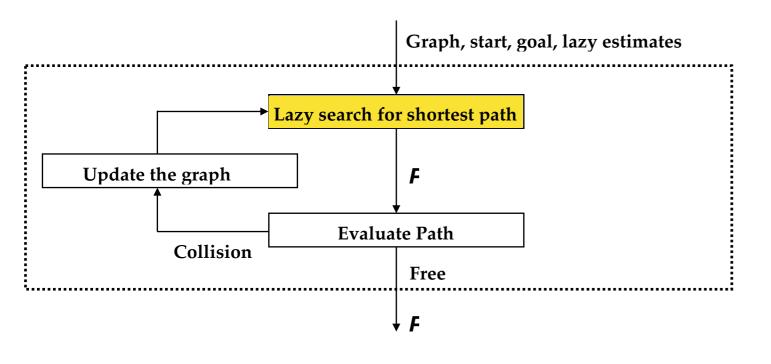


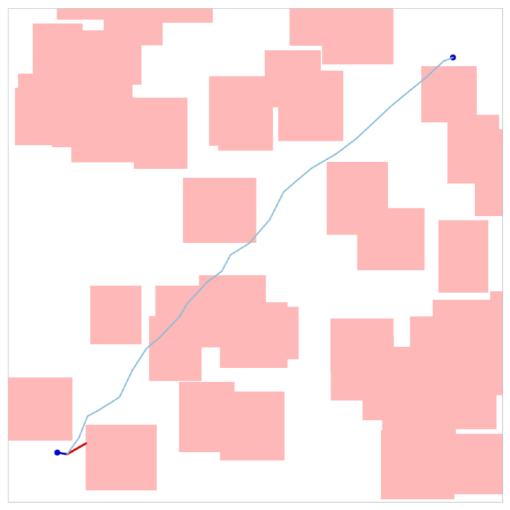




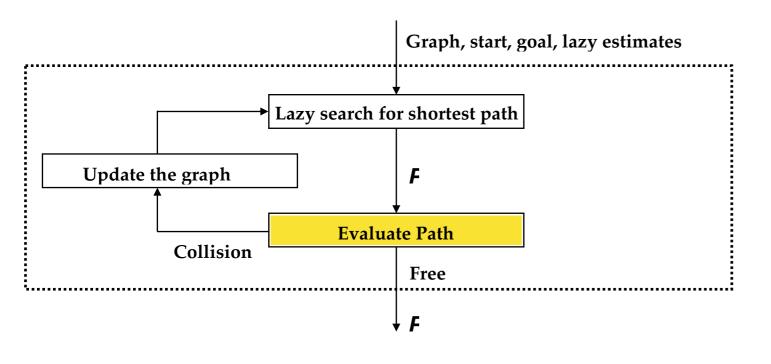


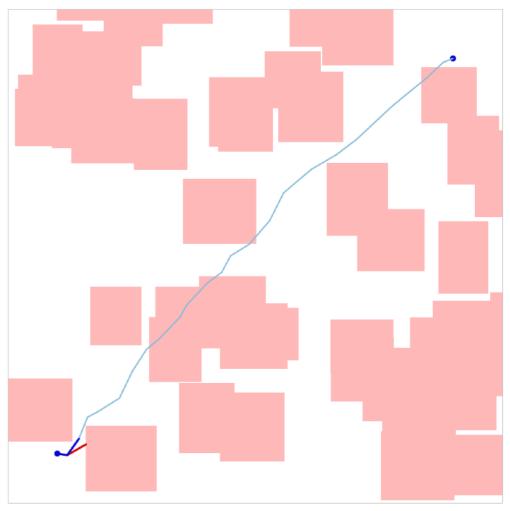




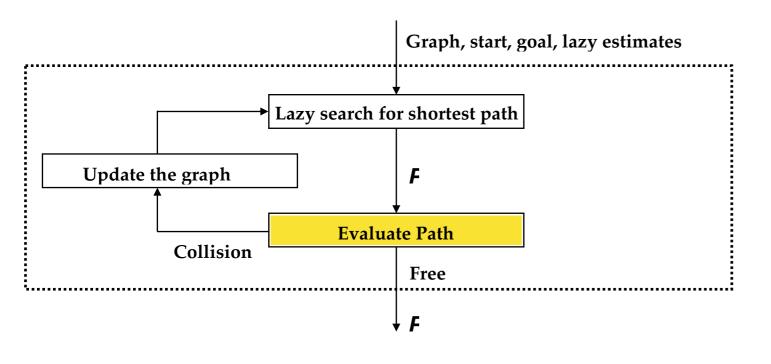


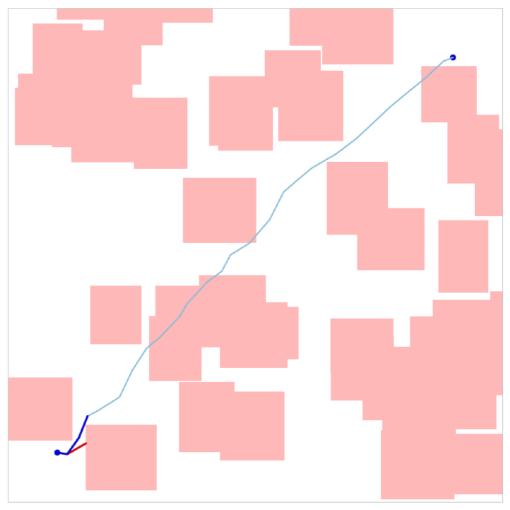




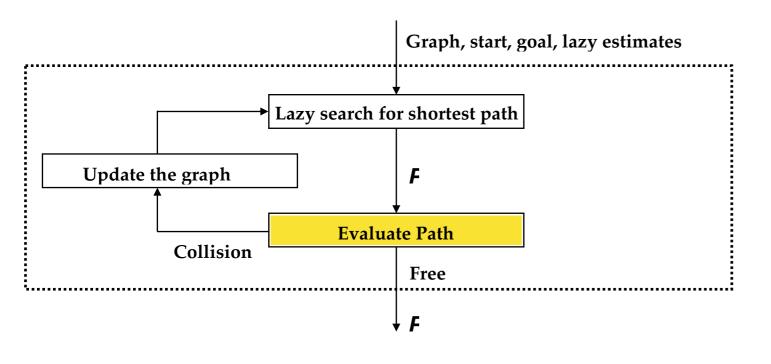


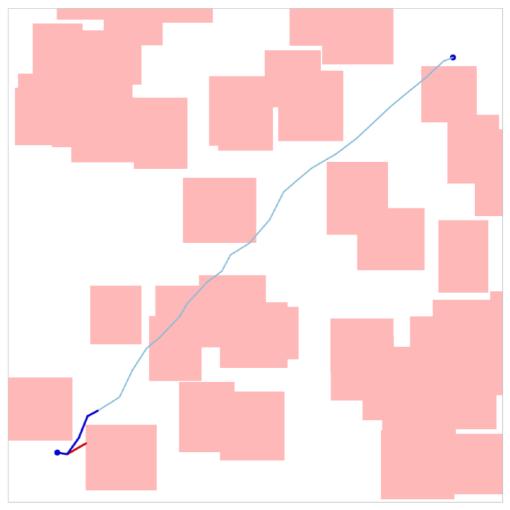




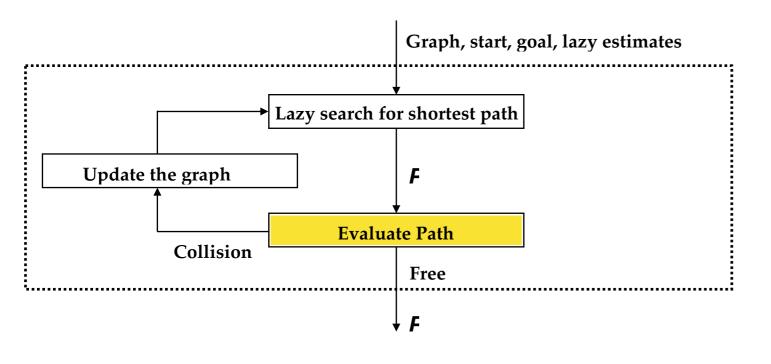


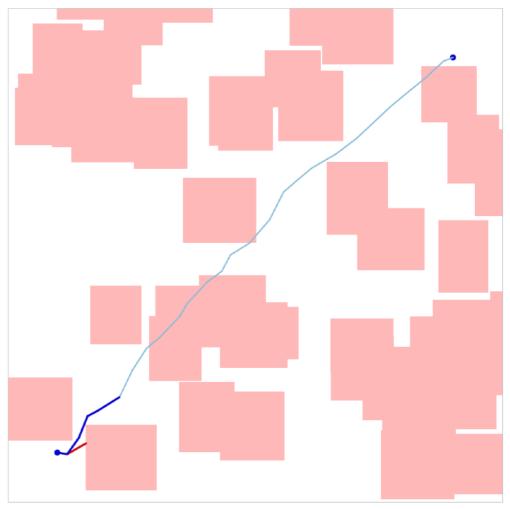




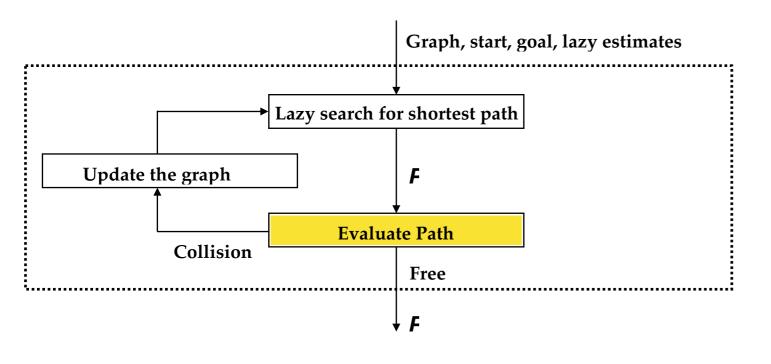


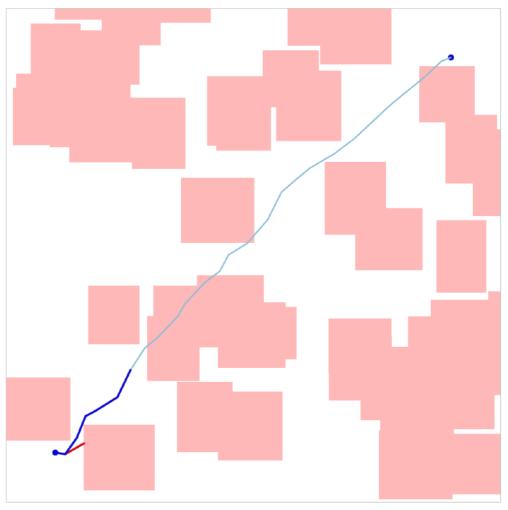




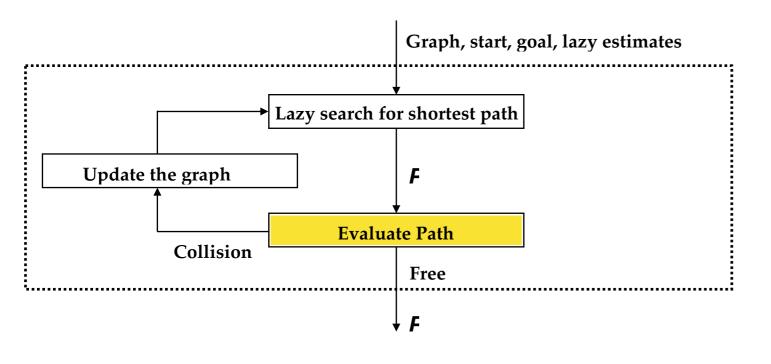


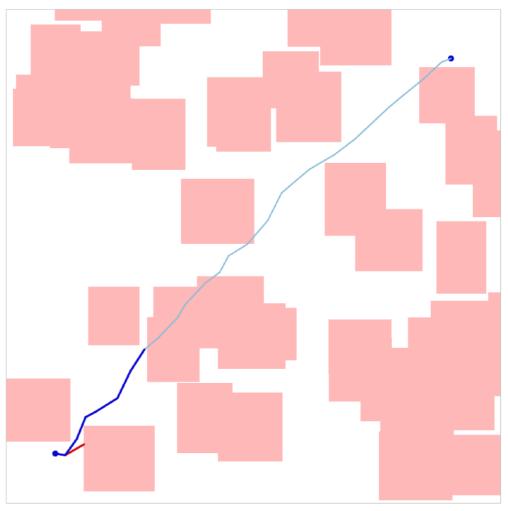




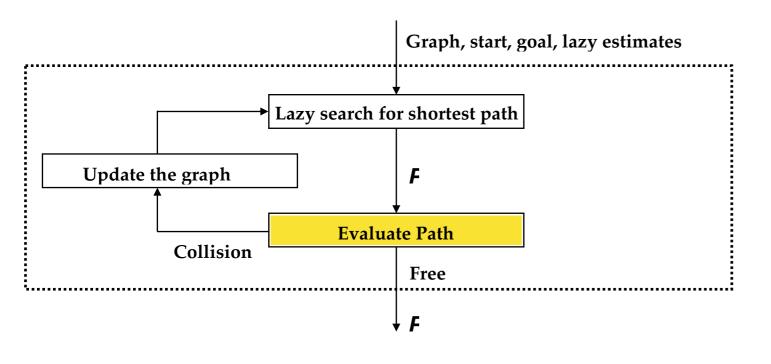


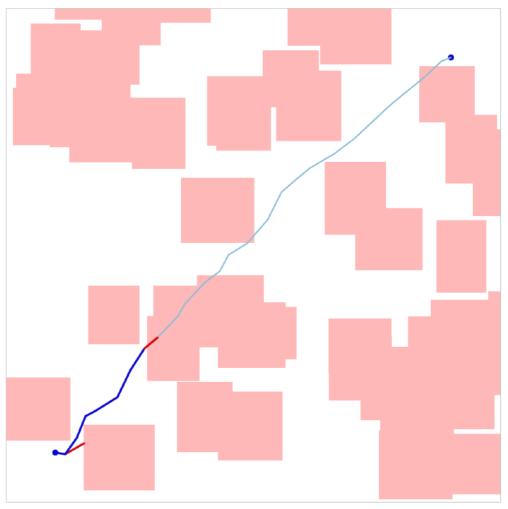




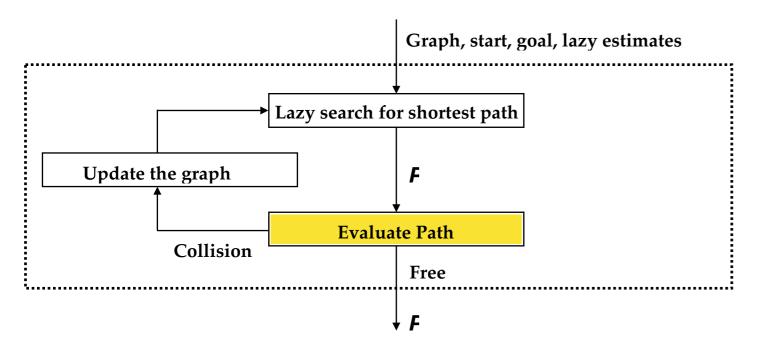


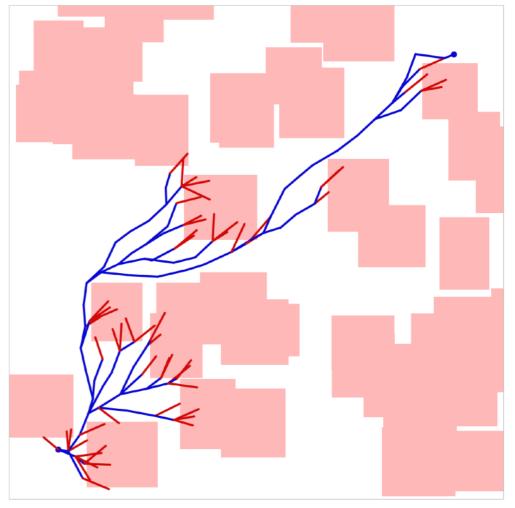




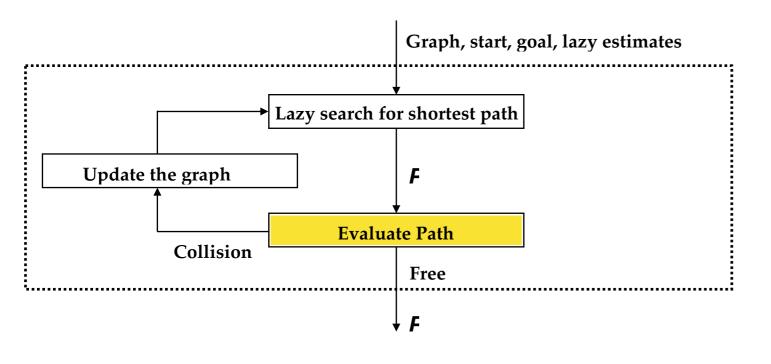


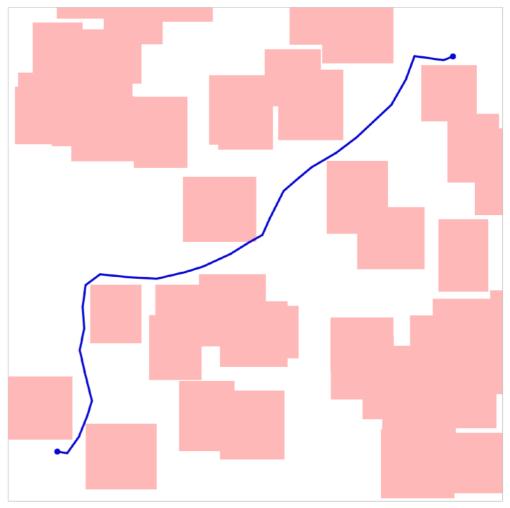






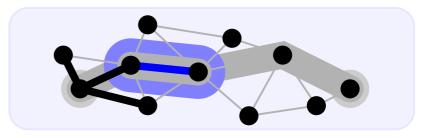




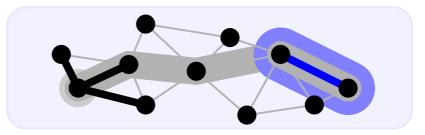


Edge Selectors

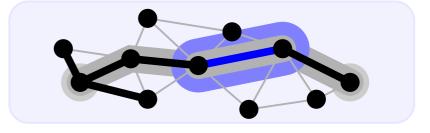
Forward (first unevaluated edge)



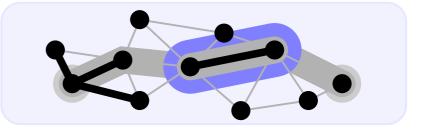
Reverse (last unevaluated edge)



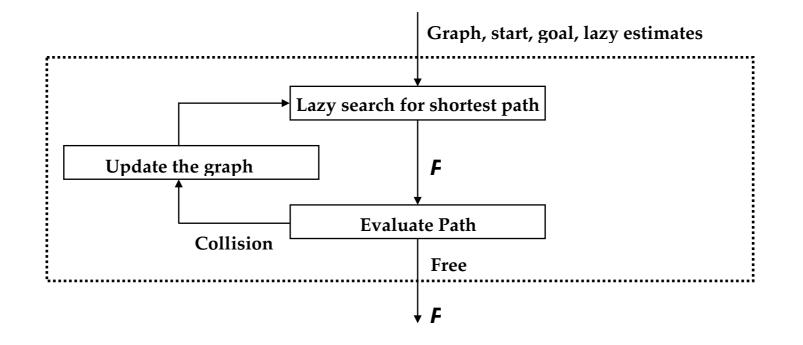
Alternate (alternate Forward and Reverse)



Bisect (furthest from an unevaluated edge)

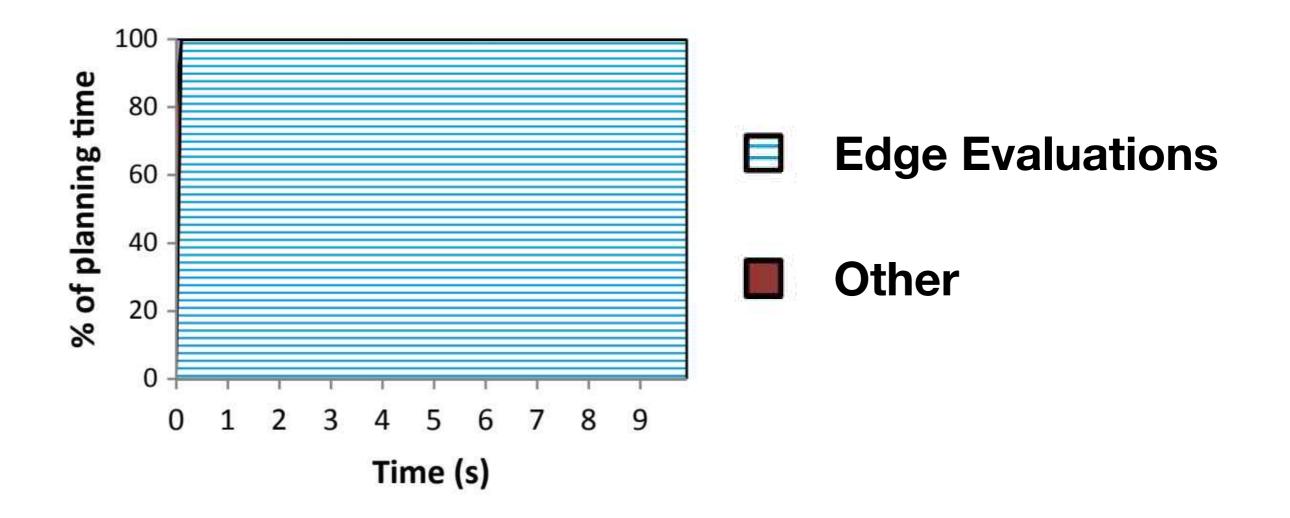


The Oracle is a LazySP Selector!



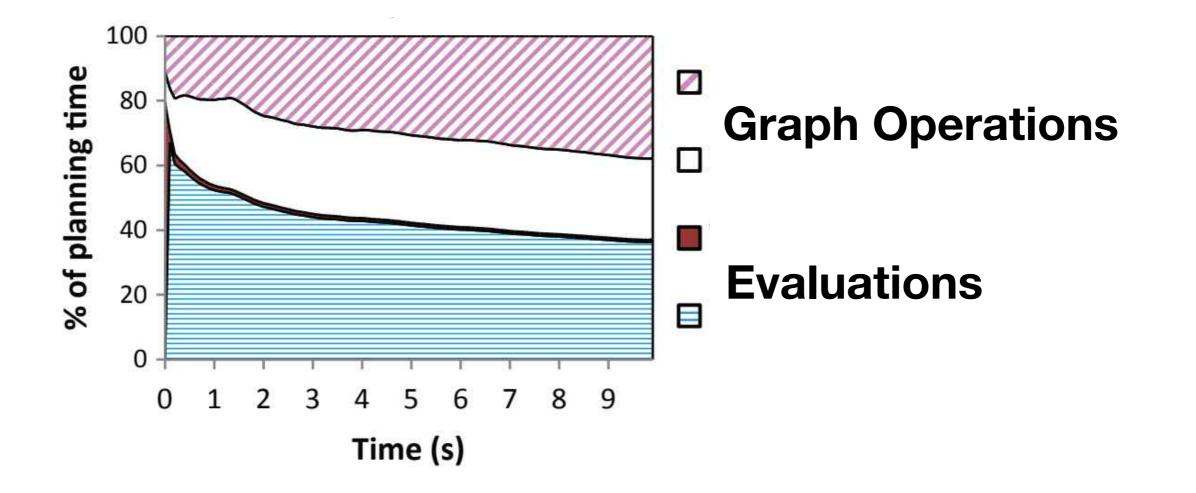


The Story Continues ...



Hauser, Kris., Lazy collision checking in asymptotically-optimal motion planning. ICRA 2015

The Story Continues ...



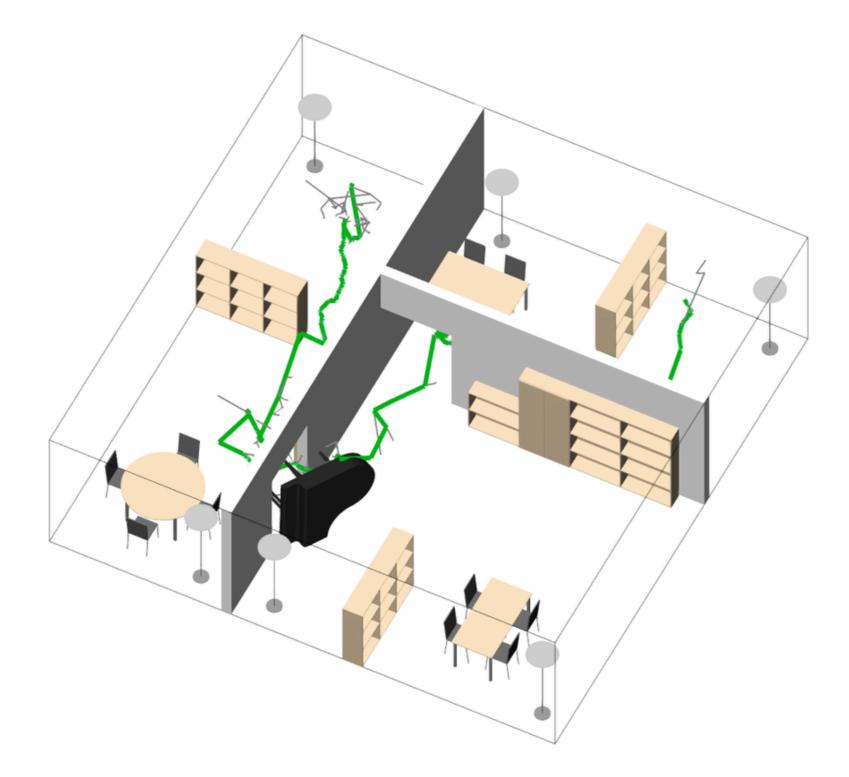
Is there a Search Algorithm that Minimizes the Number of Edge Evaluations?

LazySP

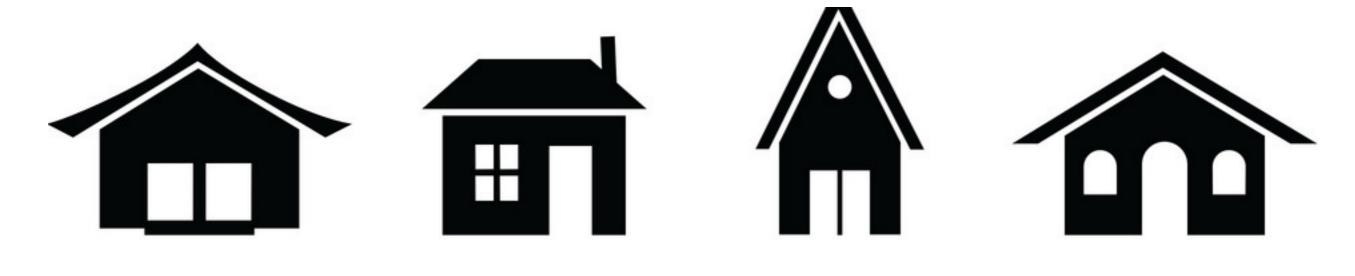
ICAPS 2018 [Best Conference Paper Award Winner]

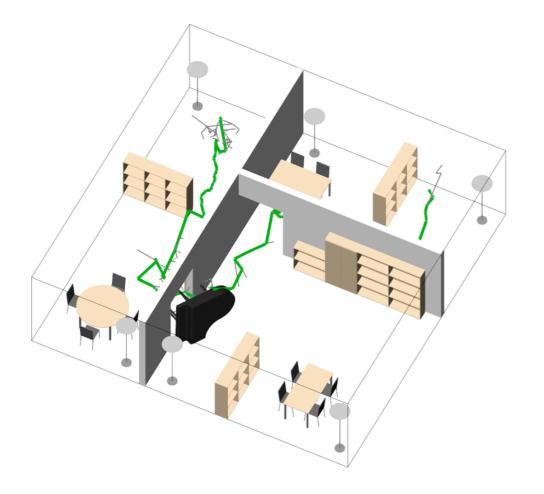
First Provably Edge-Optimal A*-like Search Algorithm

The Piano Movers' Problem



The Experienced Piano Movers' Problem

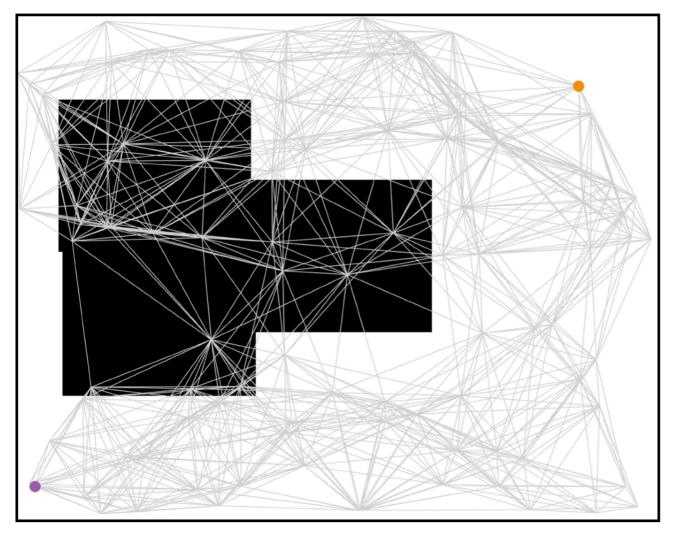




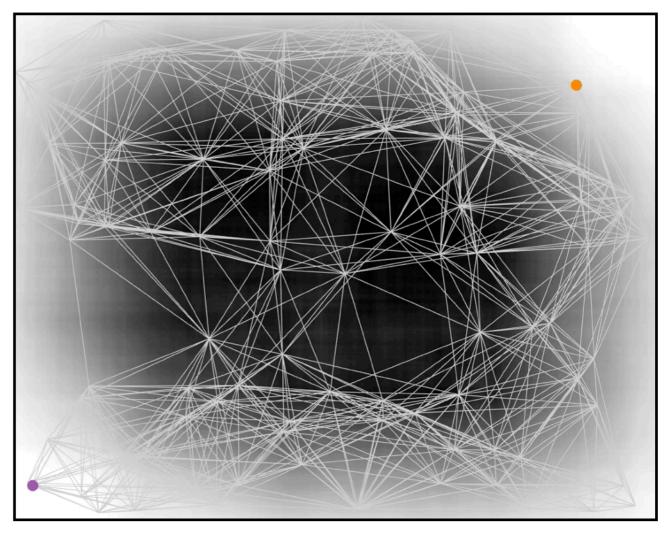
New Piano. New House. Same Mover.

A Bayesian Approach to Edge Evaluation

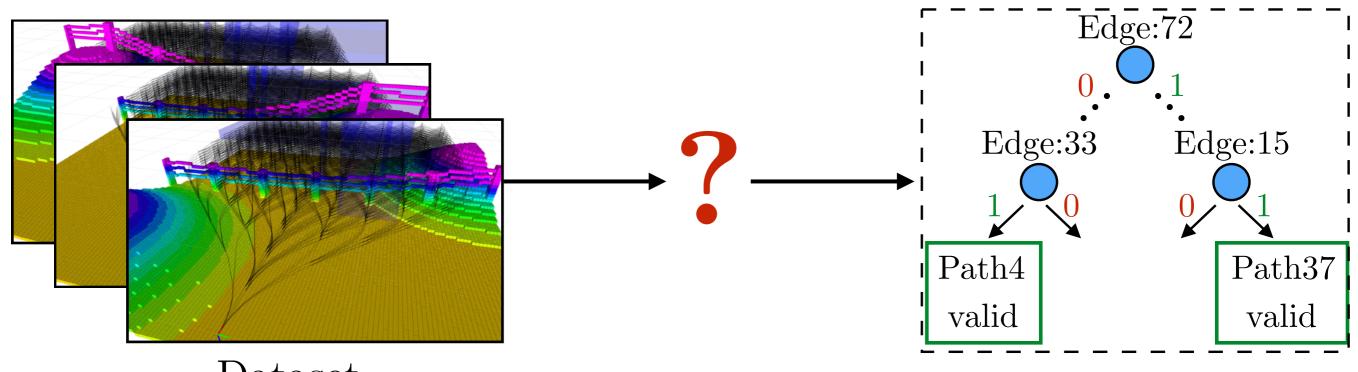
Ground truth



Agent's belief



Approach: Offline decision tree via DIRECT

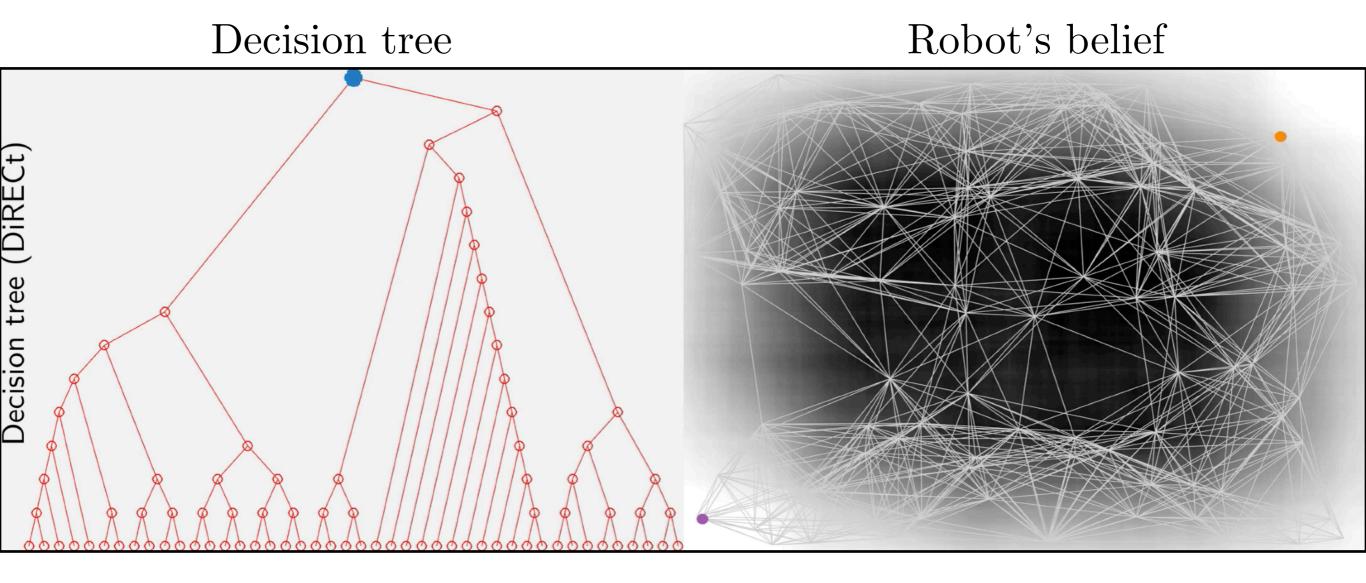


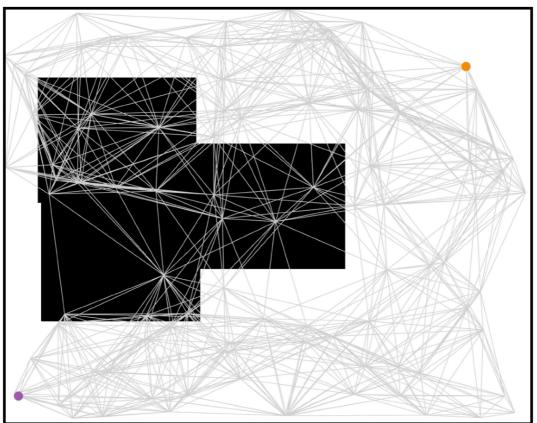
Dataset

Problem is NP-Hard (Javdani et al. '14)

DIRECT (Chen et al '15) frames it as a graph cut problem Greedy policy ($\mathcal{O}(|E| |\Xi| |\Phi|)$) is near-optimal

Executing DIRECT online is expensive, cached as a decision tree





Ground truth

Problem: What if test world not in dataset?

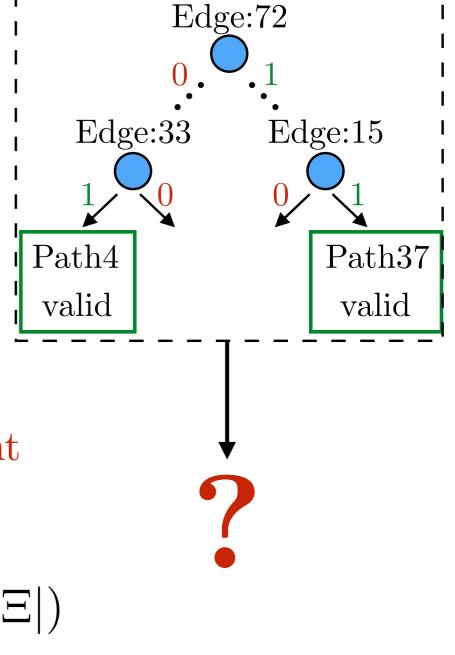
This can happen because we cannot enumerate all $\mathcal{O}(2^{|E|})$ worlds.

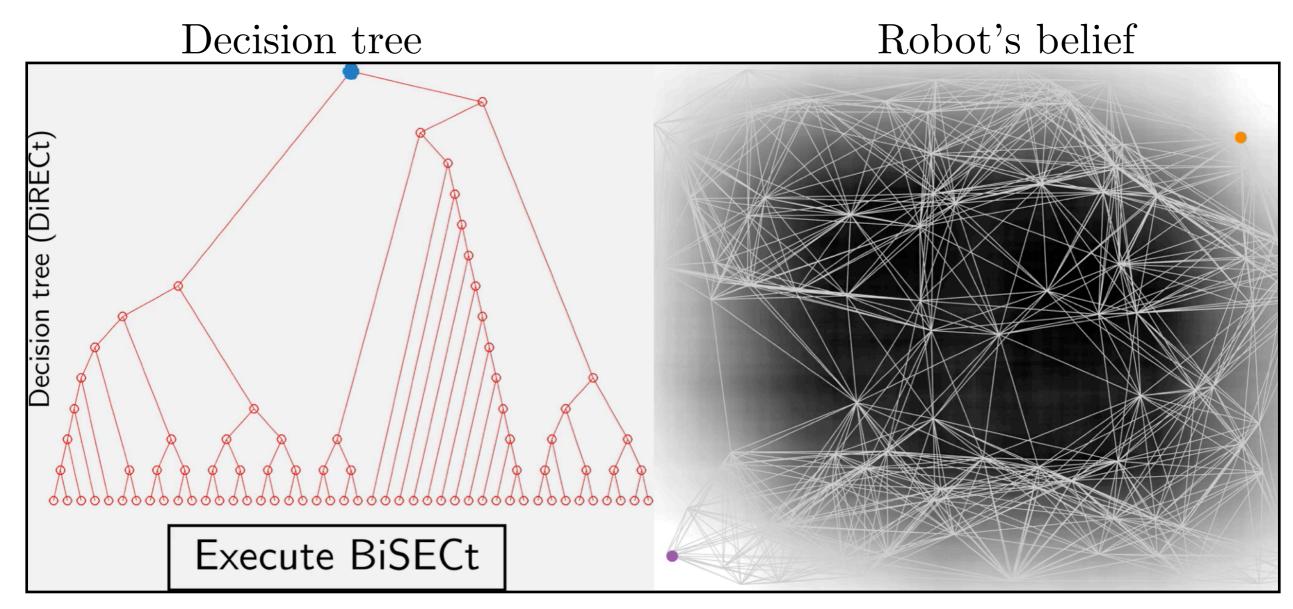
Solution: We developed BISECT [NIPS'17] that can reason about all worlds implicitly.

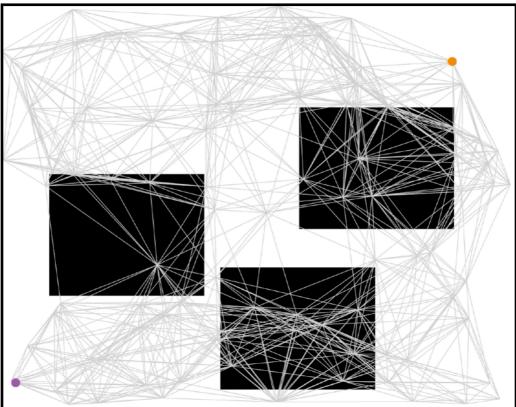
1. BISECT assumes edges are independent Bernoulli r.v.

2. BISECT has linear complexity: $\mathcal{O}(|E||\Xi|)$

3. We execute BISECT from the leaf of the tree

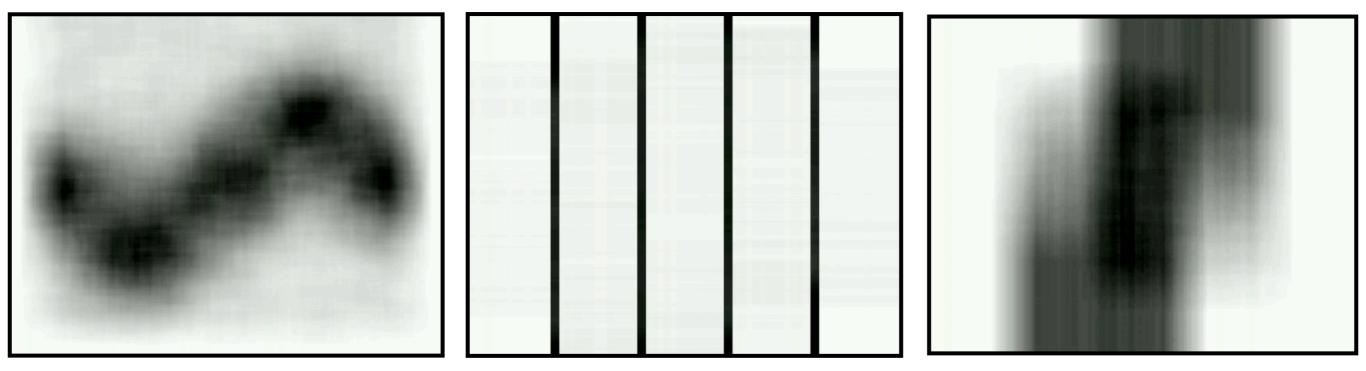






Ground truth

DIRECT + BISECT exploits structure



DIRECT + BISECT (14)

DIRECT + BISECT (24)

DIRECT + BISECT (23)

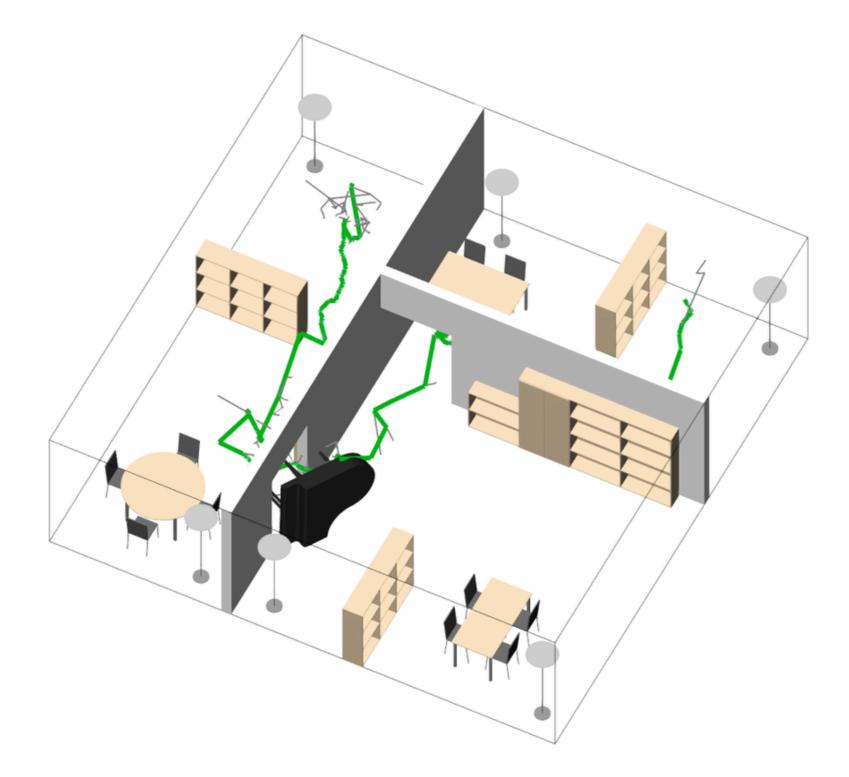


LAZYSP (155)

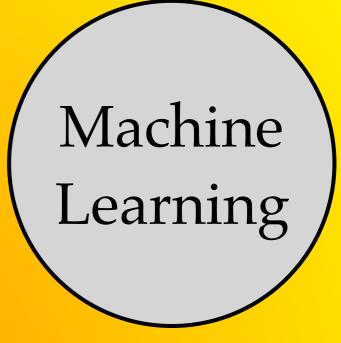
LAZYSP (838)

LAZYSP (251)

The Piano Movers' Problem







Formalizing the Core





HERB

Carnegie Mellon University

New Perspectives on the Piano Movers' Problem



Siddhartha Srinivasa Boeing Endowed Professor Personal Robotics Lab University of Washington