

Learning Sequences of Approximations for Hierarchical Motion Planning

Martim Brandão and Ioannis Havoutis

ICAPS2020



Motivation

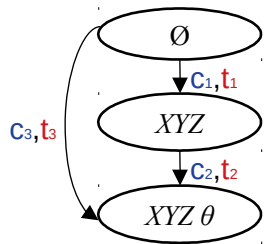
- Hierarchical motion planning: solve approximate problems to guide search on original problem
- Planner design is now a manual & ad-hoc process: which problem approximations? in which sequence?

Contributions

1. We automatically obtain sequences of problem approximations for hierarchical motion planning.
2. Users can control trade-off between computation speed and optimality.

Hierarchical planning as a graph

ST-DAG graph of problem approximations. State-space assumed factored: approxs. are combinations of factors. Edge costs introduce an order to approximations (shortest path on graph). Computation time budget for planning on each subspace (any-time planners)

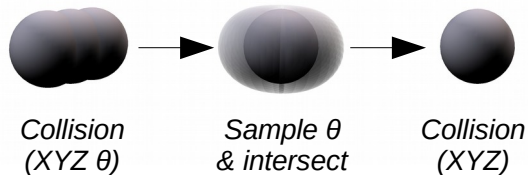


Learning the best sequences of approximations

- Evolutionary optimization to learn (c,t) on set of planning problems
- Estimate Pareto-front of total computation time vs final motion cost

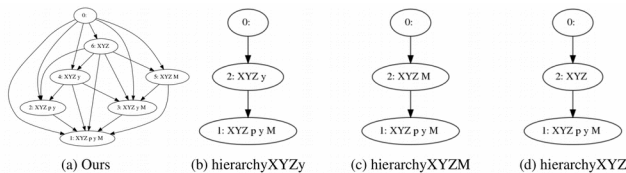
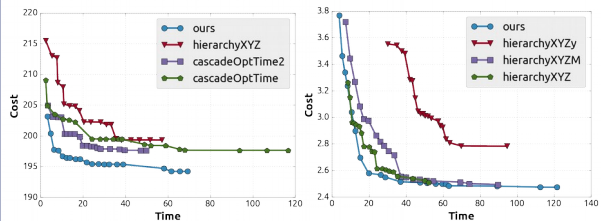
Obtaining collision geometry for sub-spaces

Sample states along removed components, intersect geometries.



Results

- Faster planning and lower-cost motion than best manual hierarchies
- Better performance on unseen problems



What is missing?

- More general hierarchies, e.g. general state-space projections
- Gradually refine discretization on same state-space
- Continuous parametrisation: now discontinuous, makes optimization difficult (inspiration from NN arch search?)