# Robot path planning in high-dimensional space

Jing Yang -- jyang@cse.yorku.ca
Supervisors: Michael Jenkin and Patrick Dymond
Department of Computer Science and Engineering, York University, Toronto, Canada

## Introduction

- Three key components in an intelligent robot:

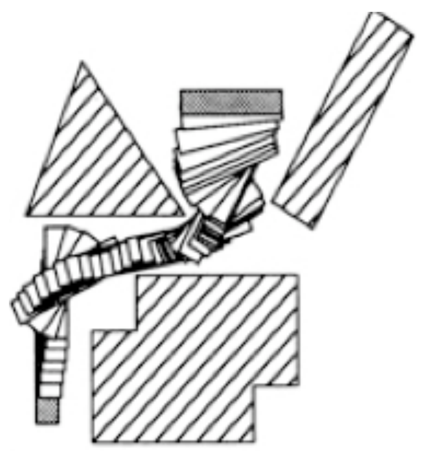
**Perception**:
vision, hearing, touch, smell, ...

**Cognition**:
think, reason, **plan**, learn, ...

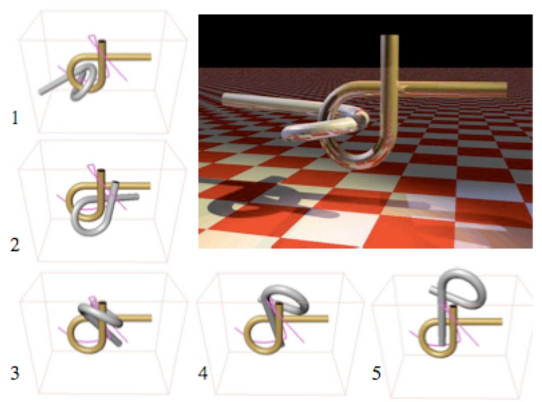
**Action**:
**motion**, **manipulation**, speaking, ...

- Here we look at path planning specifically
  - Computing a collision-free path for a robot (rigid or articulated object) among static obstacles

Geometry of robot and obstacles | Kinematics of robot | Start and goal configurations

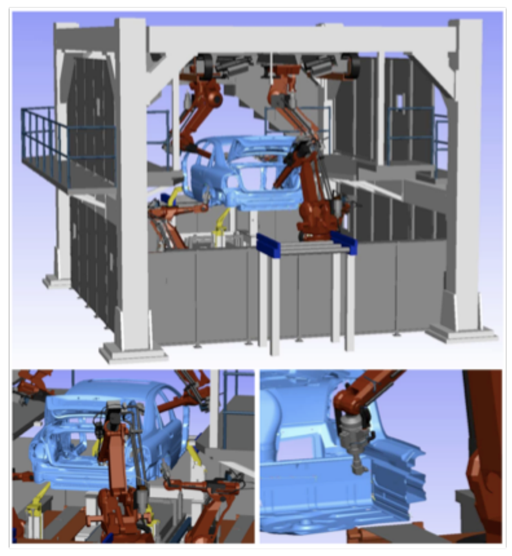


Input

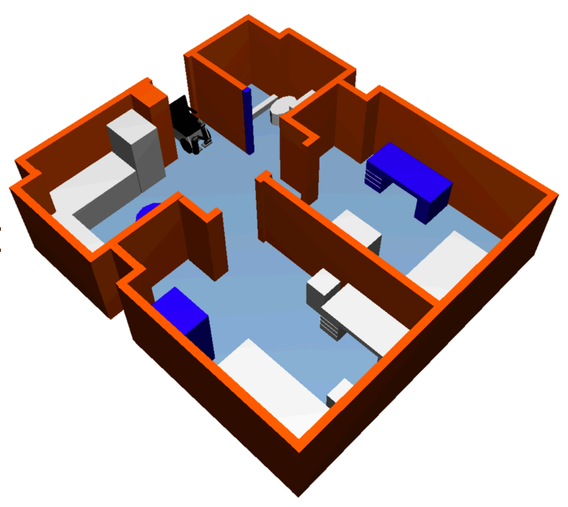




Path planner



Output

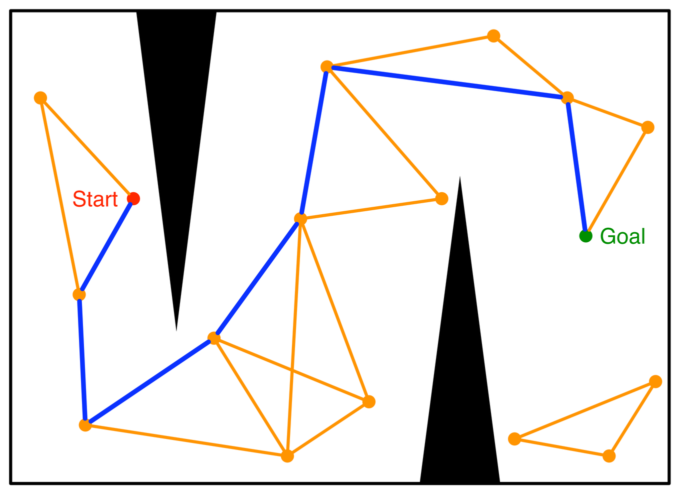


Continuous sequence of collision-free robot configurations connecting the start and goal configurations

  - Computational complexity: **NP-hard** in the degrees of freedom (DOFs) of the robot
  - Leads to the development of probabilistic solutions for high DOF problems

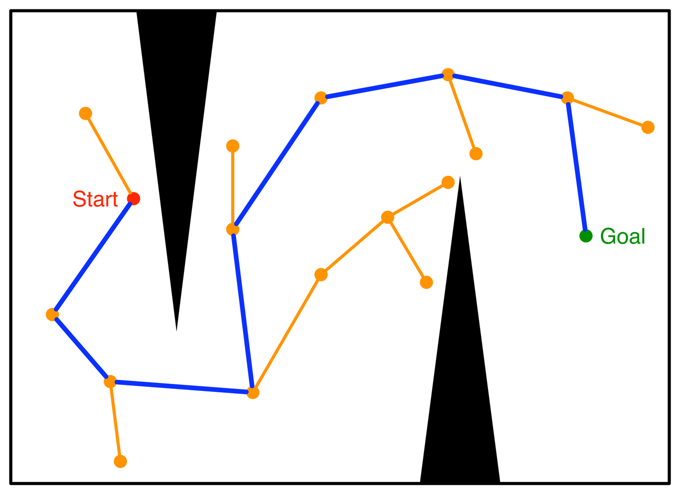
## Path planning in high-dimensional space

- **Heuristic/Sampling-based Path Planner**
  - Build a graph (roadmap or tree) to represent the geometric structure of the search space
  - Search the graph for a path connecting start and goal





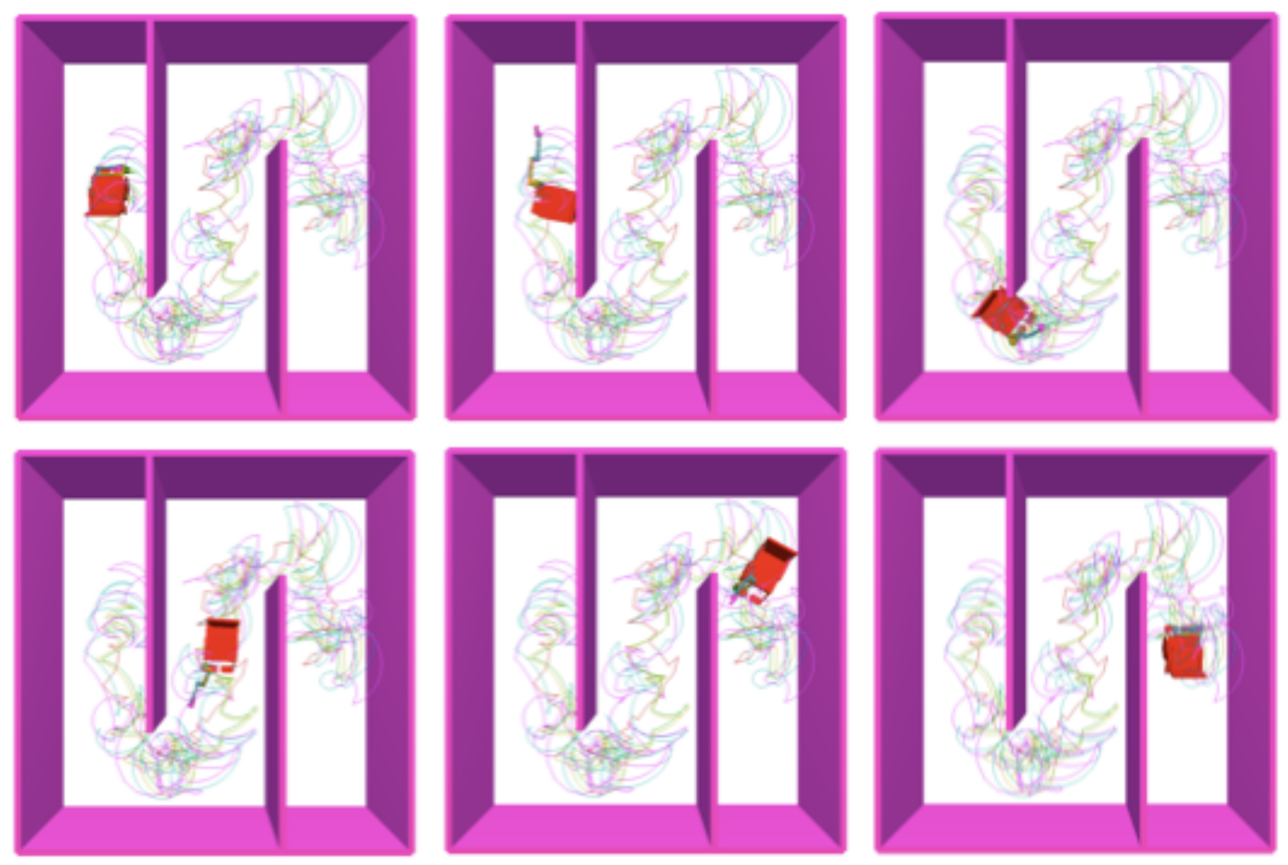
Roadmap-based Planner (PRM)



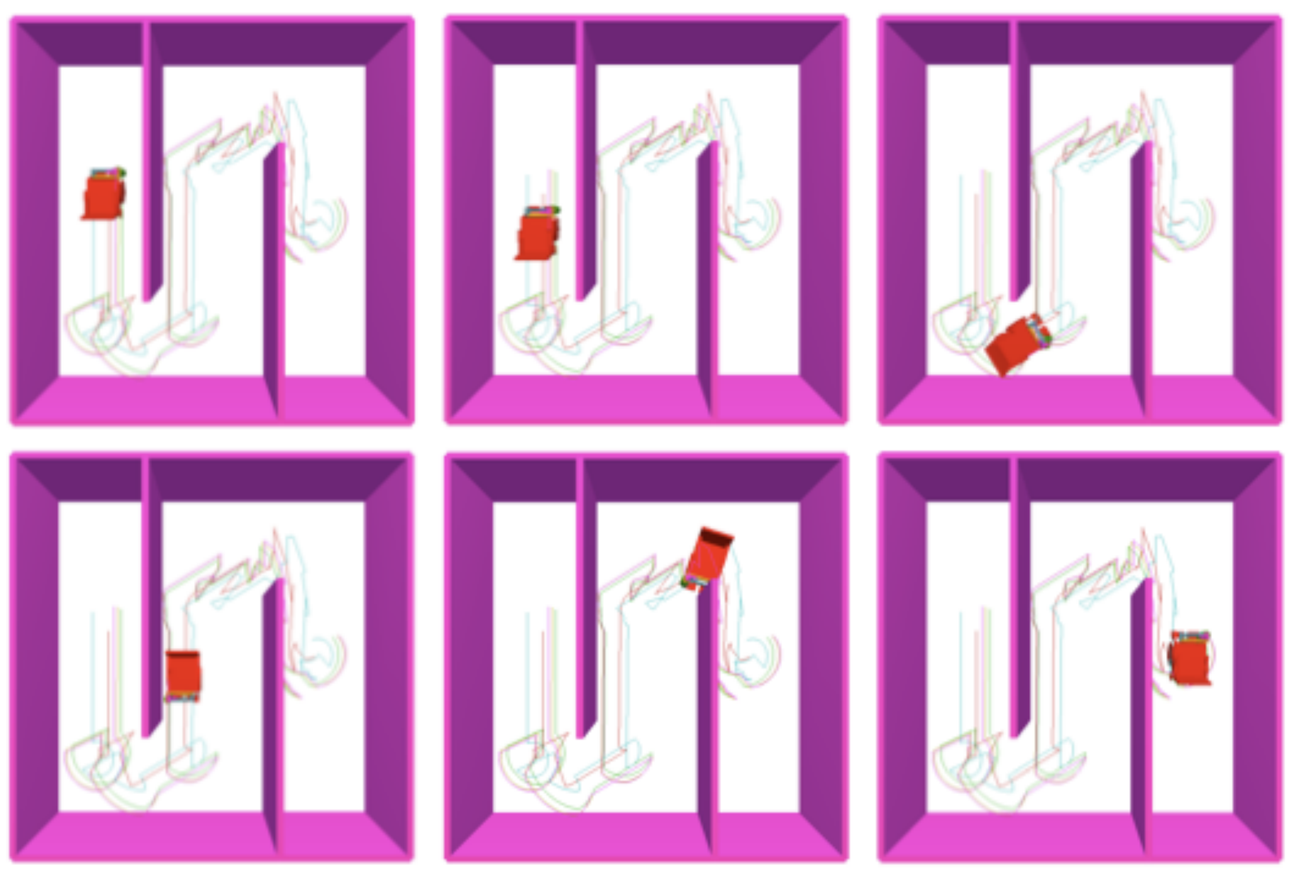


Tree-based Planner (RRT)

- **Advantages**
  - Efficient for high-dimensional and complex problems
  - Probabilistic completeness
- **Limits**
  - **Non-optimal/non-practical solutions**



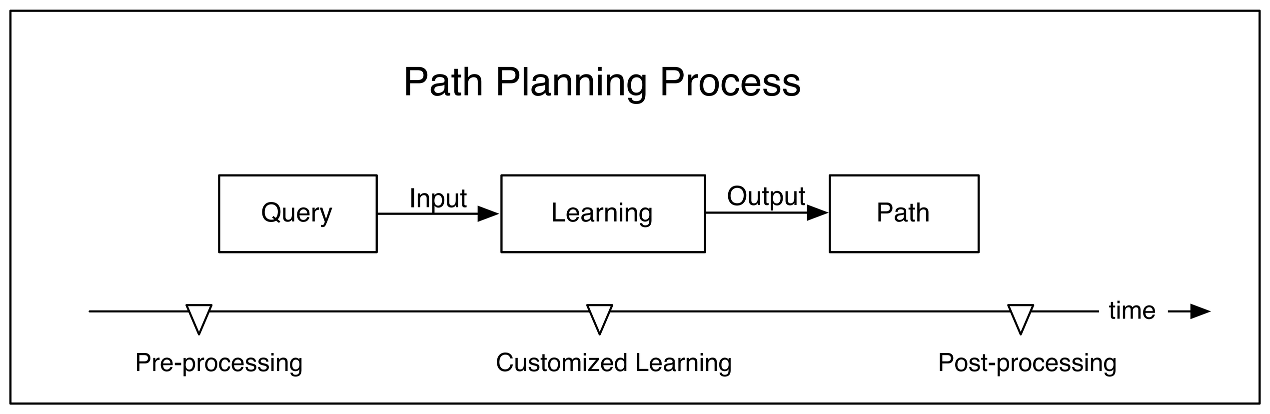
Sample path computed by PRM



Sample path computed by RRT

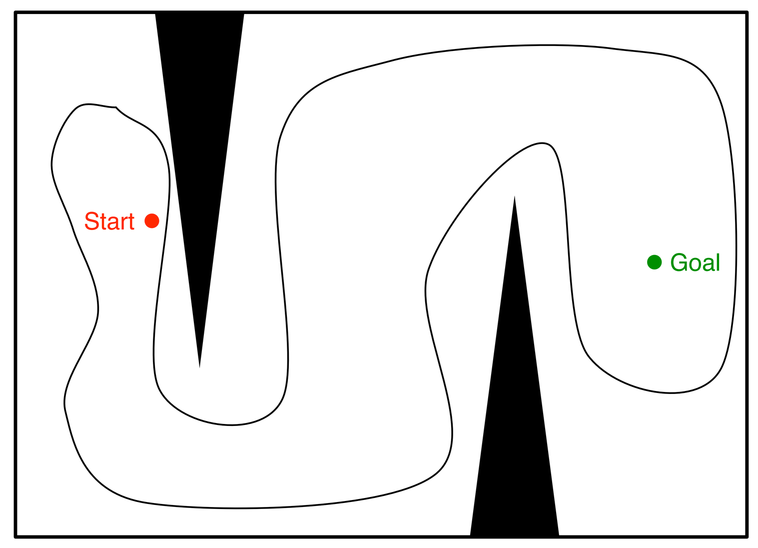
## Planning practical paths

- **Goal**
  Find paths that not only satisfy *hard constraints* (collision-free) but also meet *soft constraints* (smoothness, energy, etc.)
- **Existing algorithms**
  Integrate path practicality issue during:
  - pre-processing phase
  - learning phase
  - post-processing phase



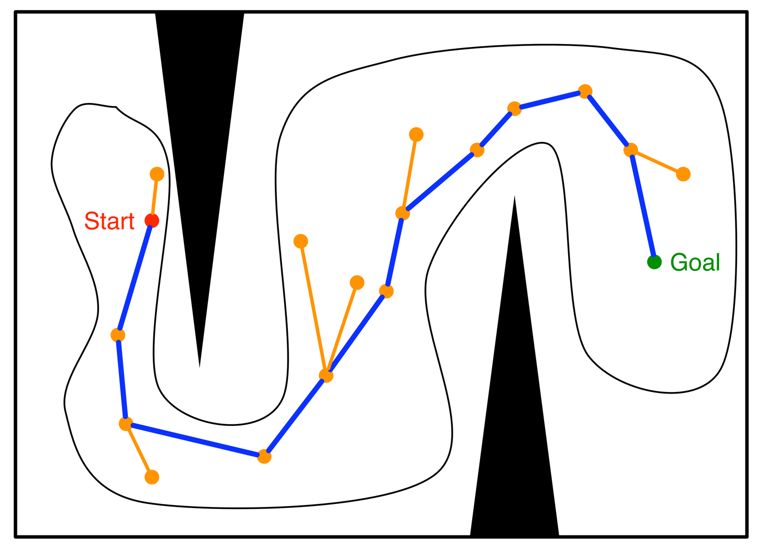


- **Our proposed algorithm**
  (1) Build a graph (roadmap or tree) to represent the general geometric structure of the search space





(2) Carve out a sub-space of the search space by hierarchically maximizing the free space around nodes and edges in the graph





(3) Find 'optimal' paths with maximum practicality by running a complete or randomized path planner within the sub-space

- **On-going work**
  - Implementation of the algorithm
  - Formalization of the definition of practicality

**Acknowledgements:**
The financial support of NSERC is gratefully acknowledged