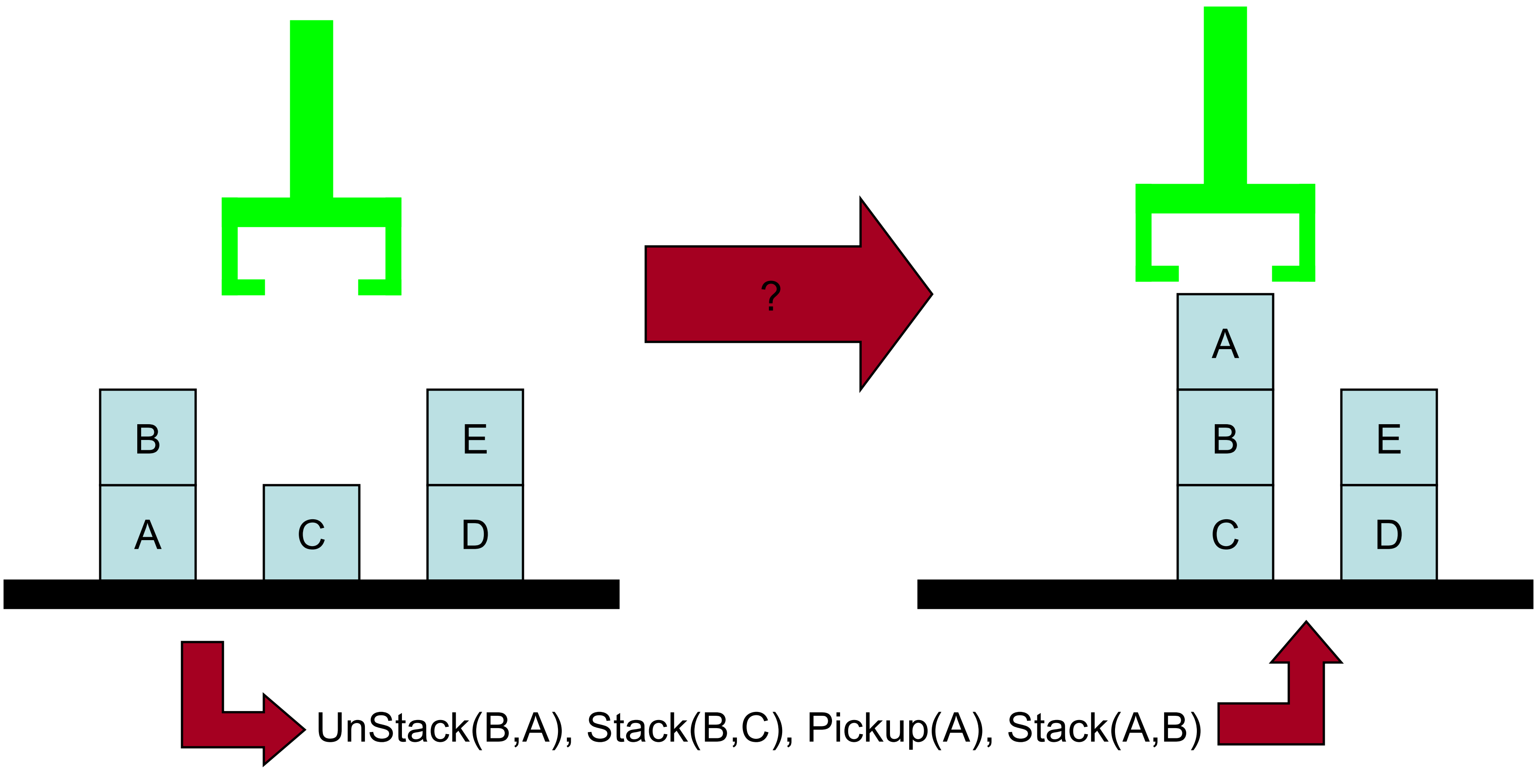
Motion Planning

CPSC 470 – Artificial Intelligence

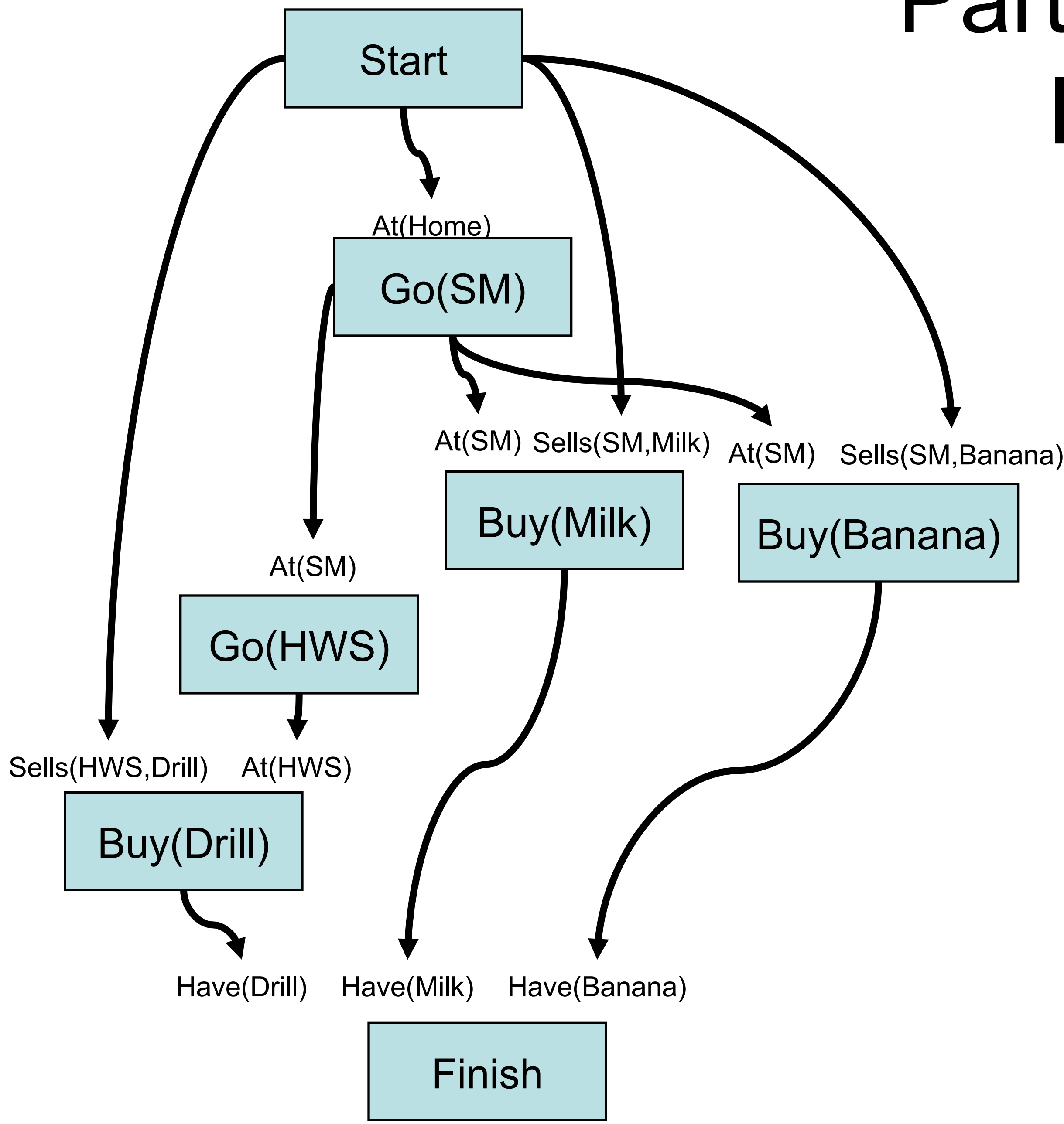
Brian Scassellati

Planning in Blocks World

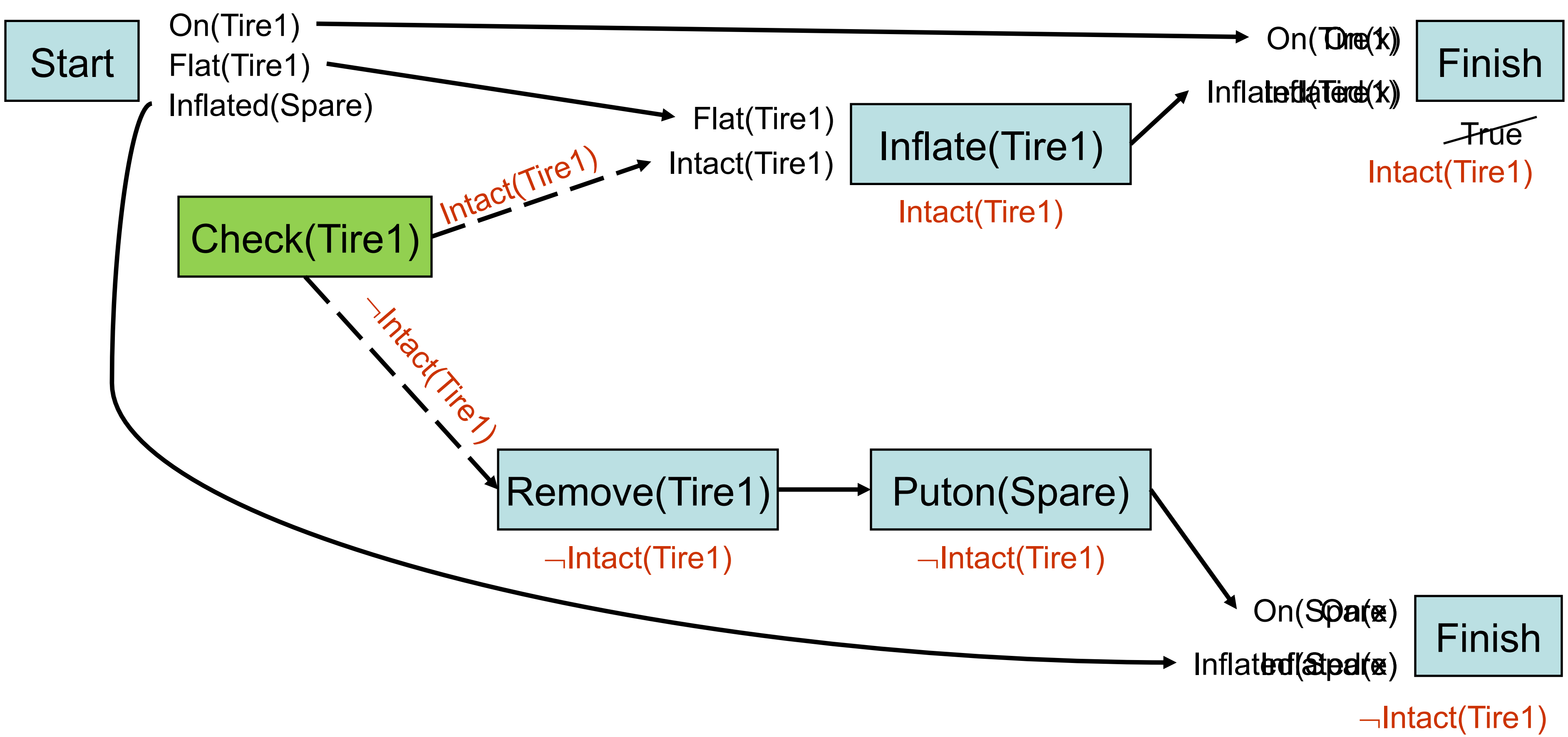


Partial-Order Plans

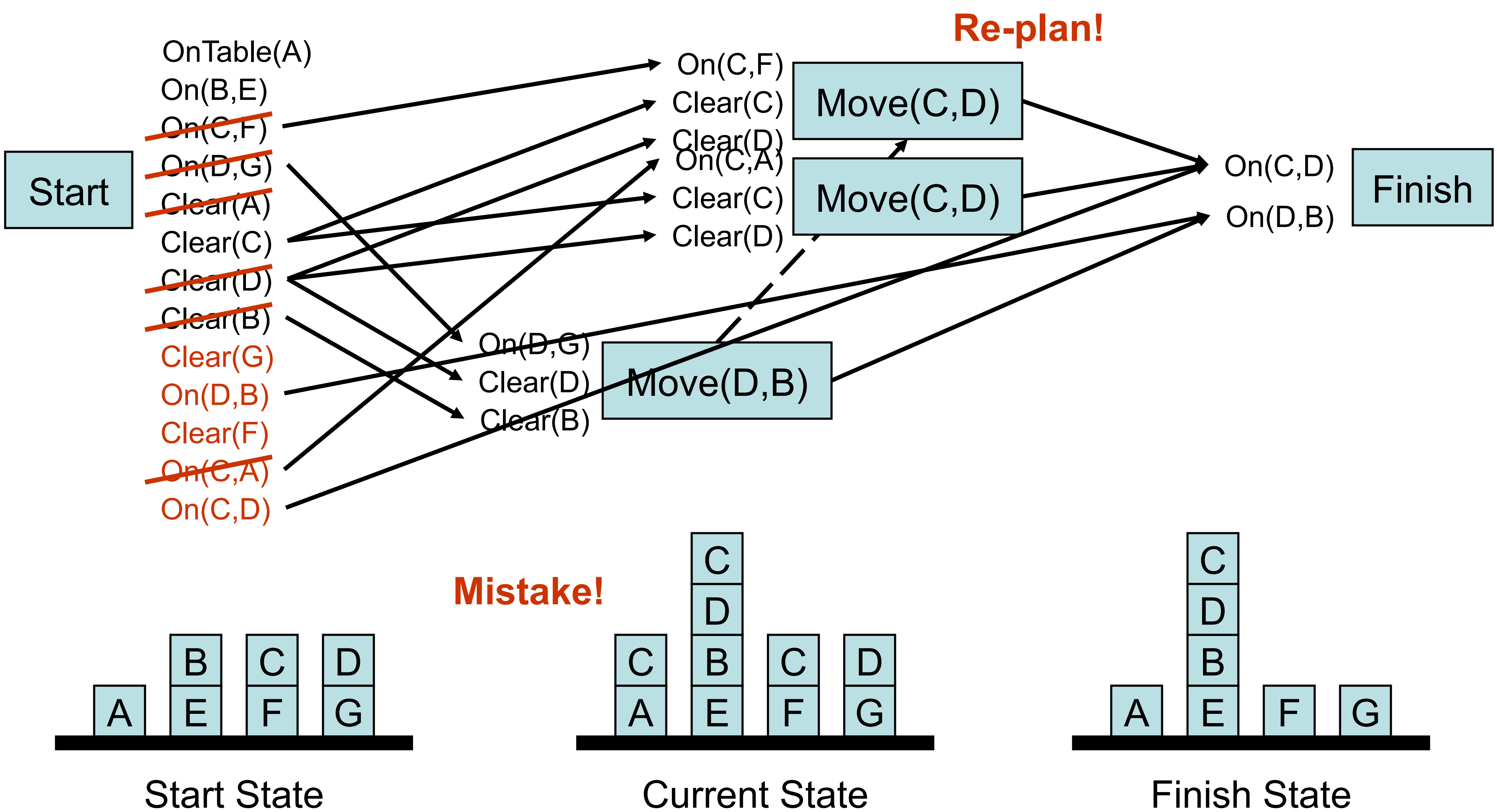
- Solution showing only causal links



Conditional Planning Example



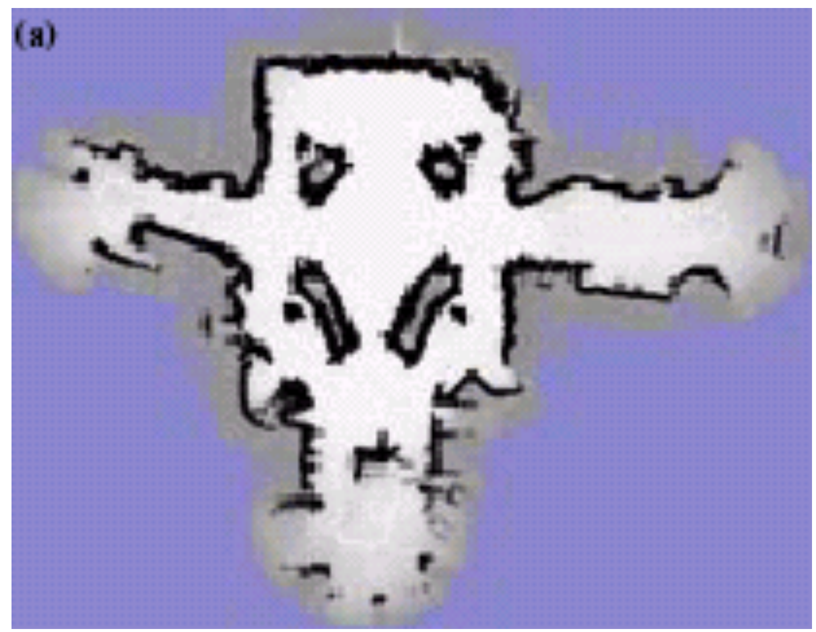
Executing a Blocks World Plan (mistakes happen…)



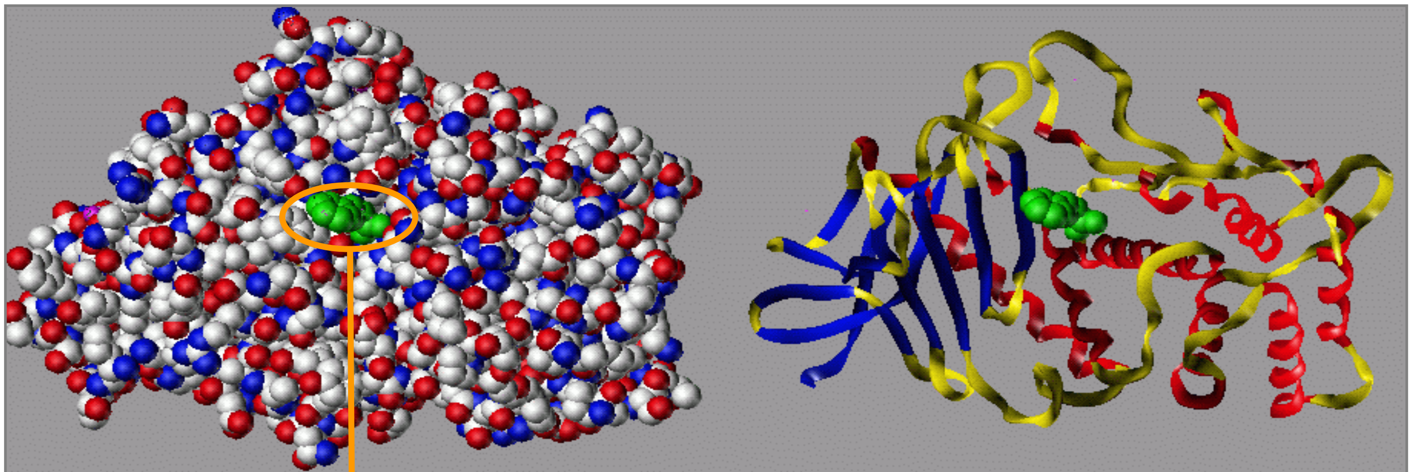
Navigation and Map Building

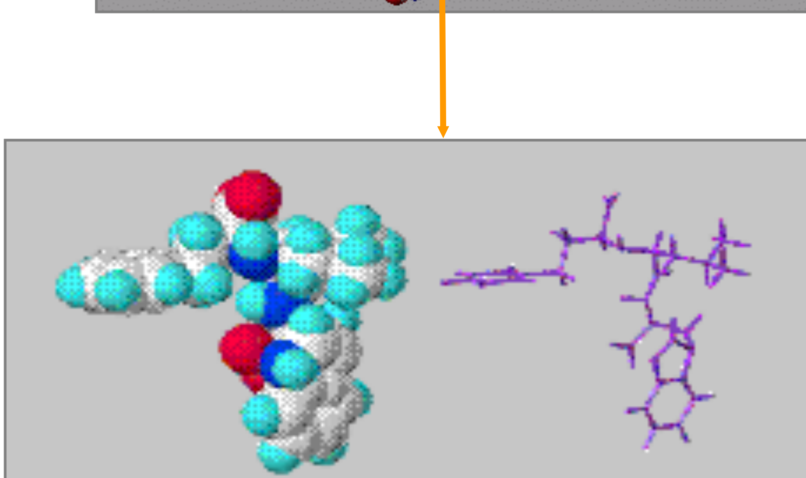
Minerva, a tour-guide robot
(Thrun, CMU)

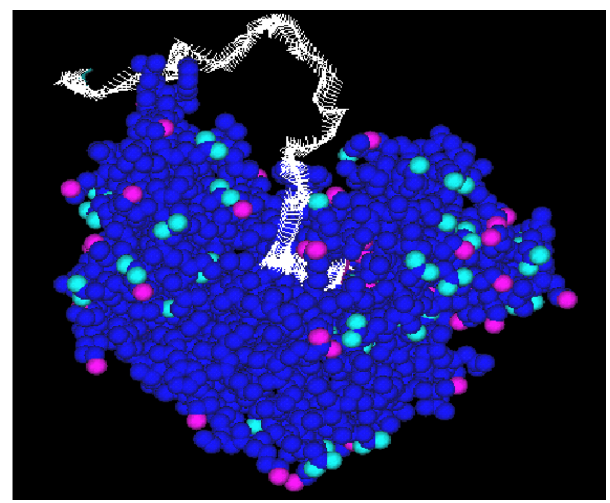
Where to move next?



Ligand-Protein Binding







Surgical Planning

DaVinci Surgical System from IntuitiveSurgical

Path Planning Input

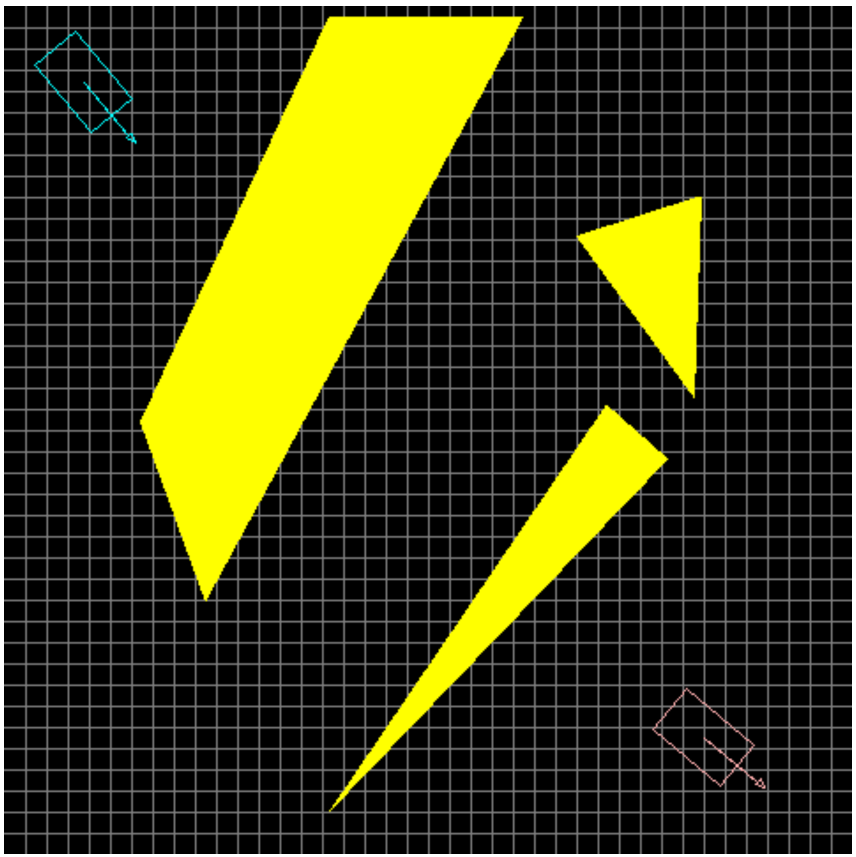


Image from Baldi, Pardini, & Pistelli

- Geometry of robot and obstacles
- Kinematics of robot (degrees of freedom)
- Starting position (blue)
- Goal position (pink)

Path Planning Goal

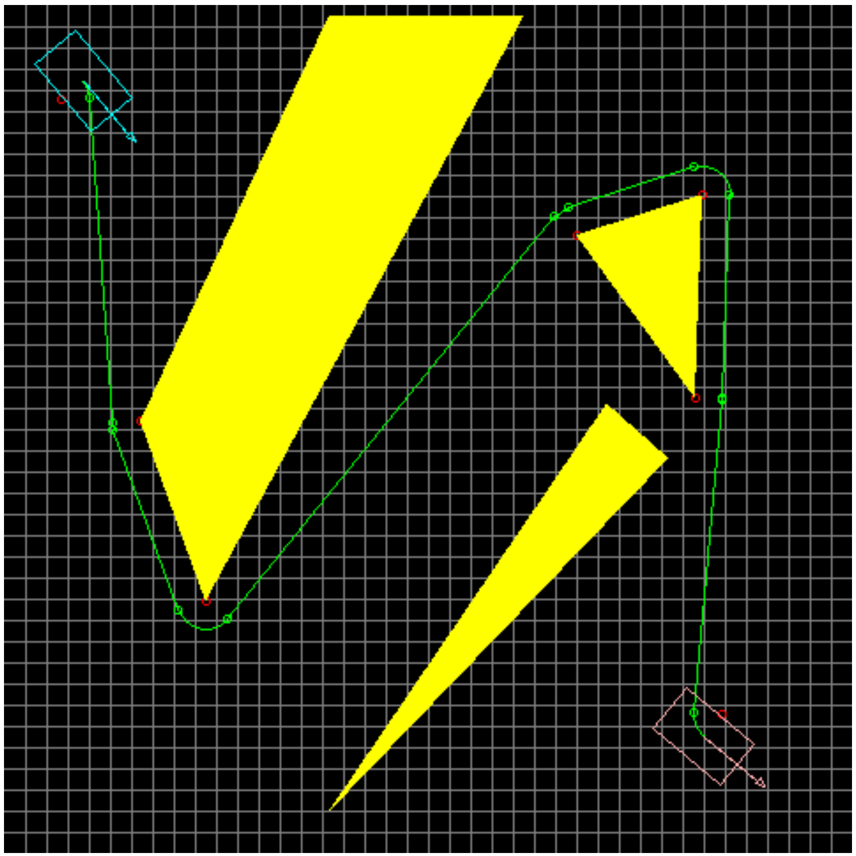


Image from Baldi, Pardini, & Pistelli

- Move the robot from the start position (blue) to a goal position (pink) while avoiding obstacles (yellow)
- Compute a collision-free path for a rigid or articulated object (the robot) among static obstacles

Path Planning Outputs

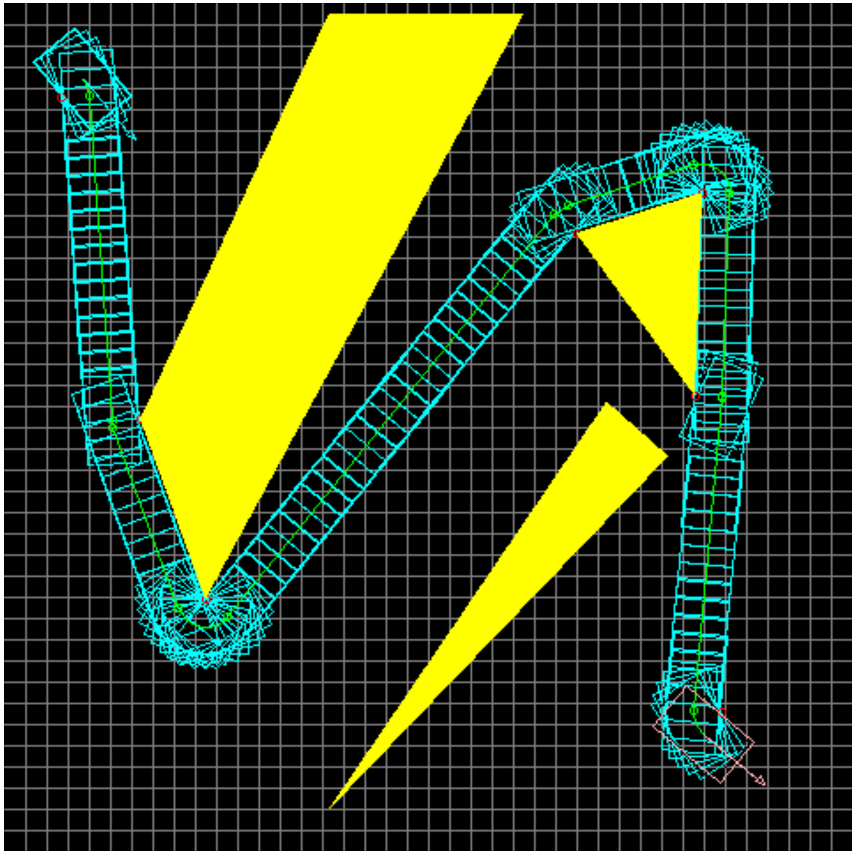
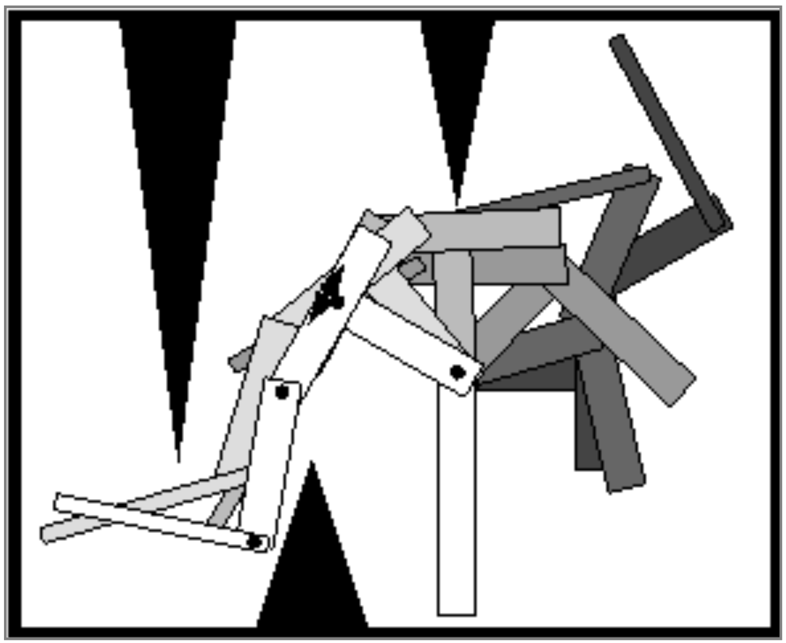
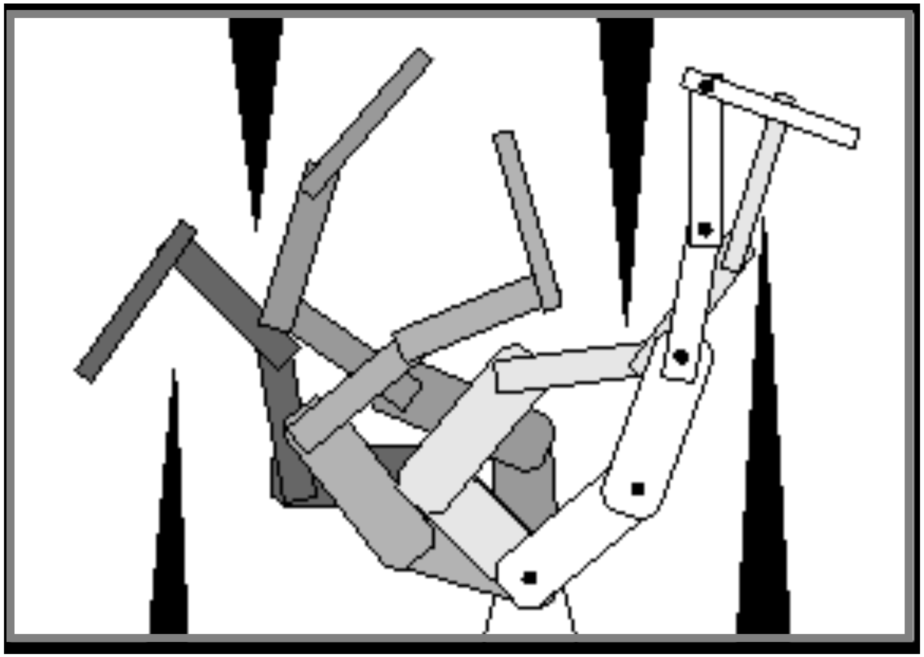


Image from Baldi, Pardini, & Pistelli

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

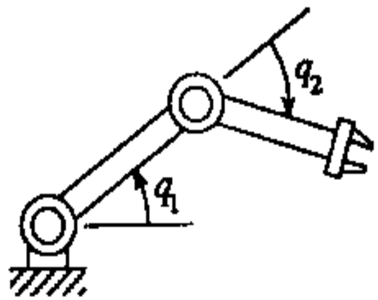
Motion Planning with an Articulated Robot Arm





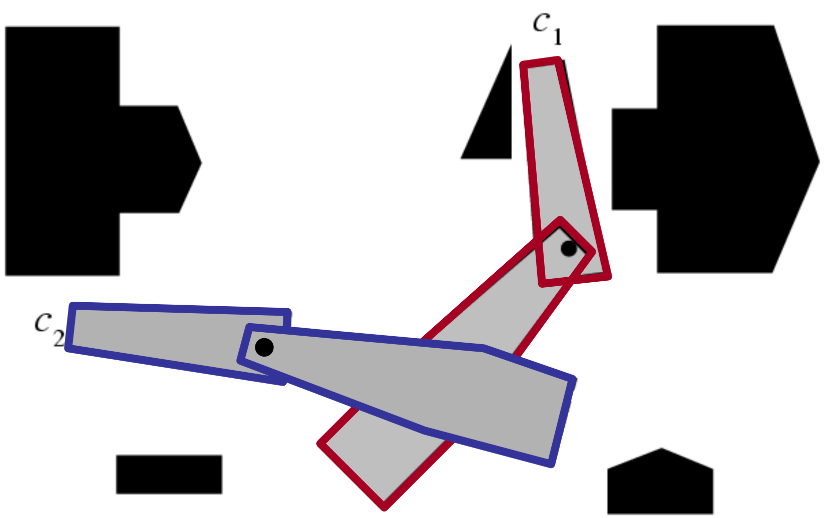
Planning the path that the endpoint needs to take is insufficient… it does not tell us how to move the individual joints

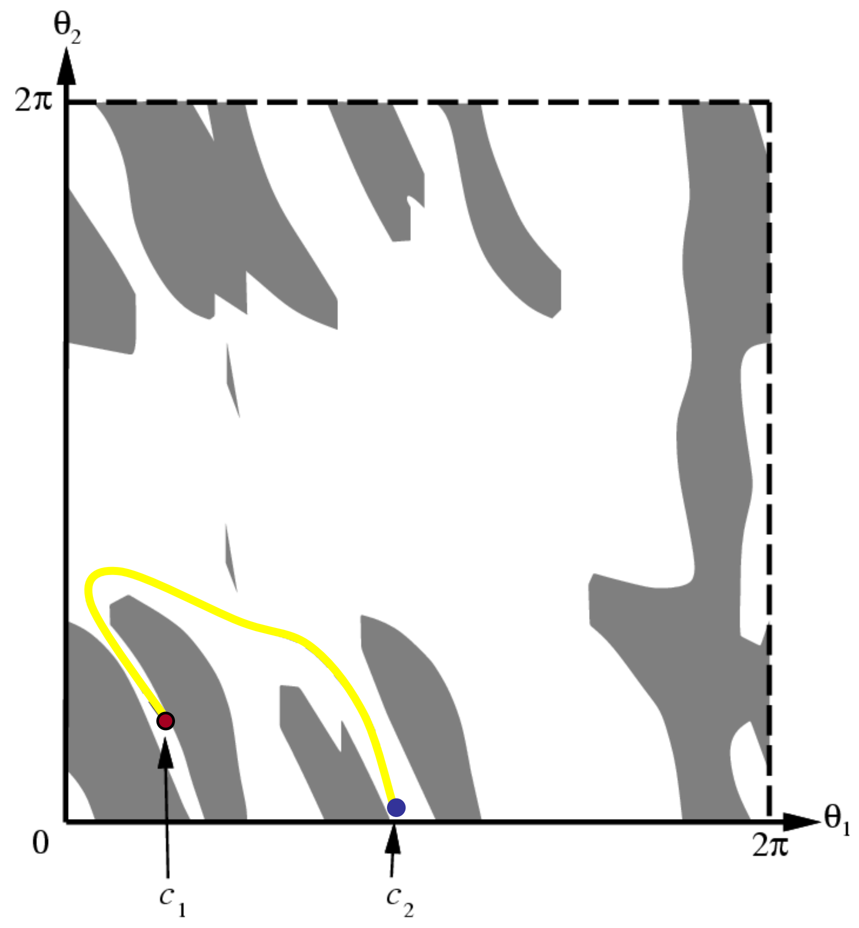
Configuration Space of a Robot



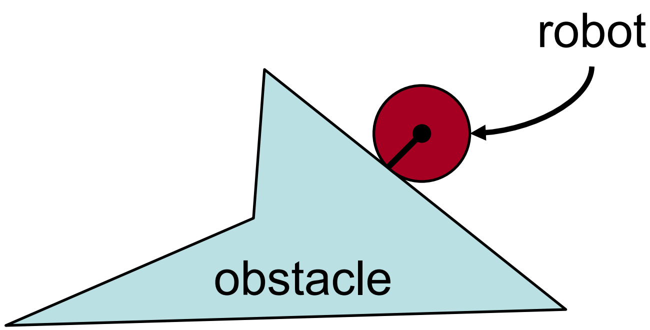
- Space of all possible configurations
- Represent within a Cartesian frame
- Allows us to treat the robot as a single point
- But the topology of this space is usually not that of a Cartesian space

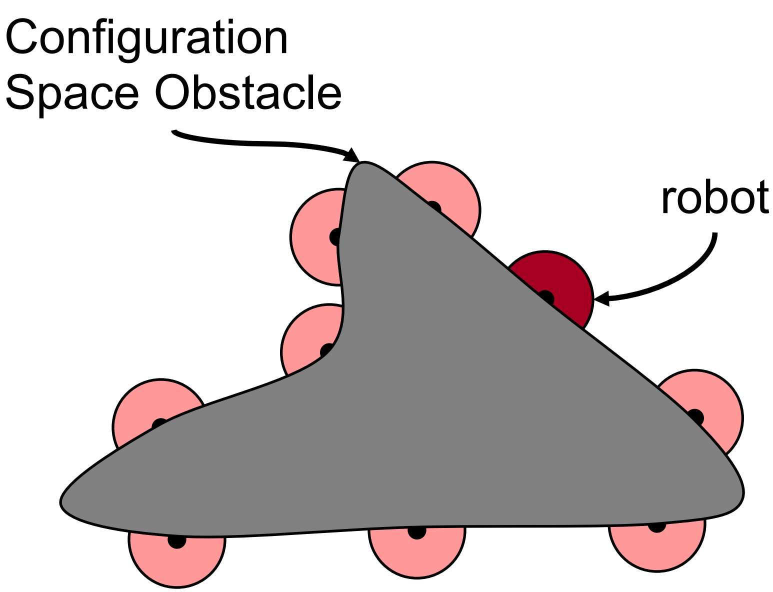
Motion Planning within Configuration Space





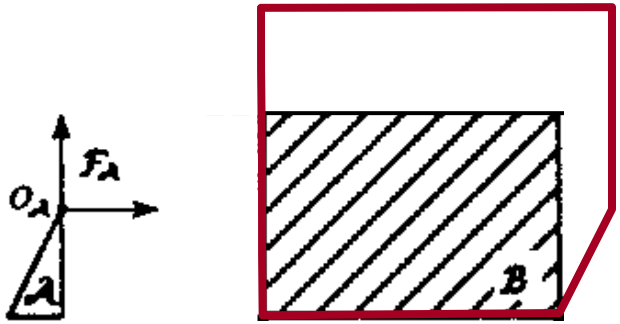
Configuration Space also allows for Obstacle Representation



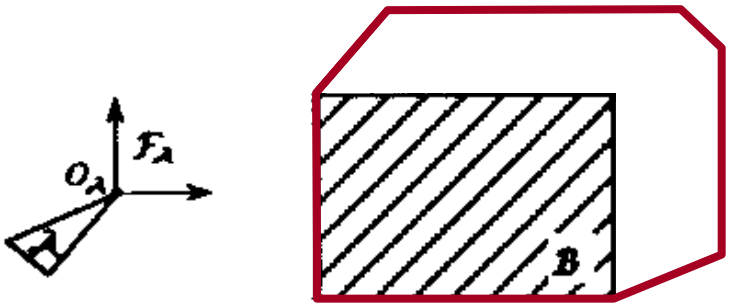


- Disc Robot in 2-D Workspace
- Define a reference point
- Because the robot has a non-zero diameter, we cannot come arbitrarily close to an obstacle
- Increase configuration space obstacles by the diameter of your robot

Rigid Robot Translating in 2-D

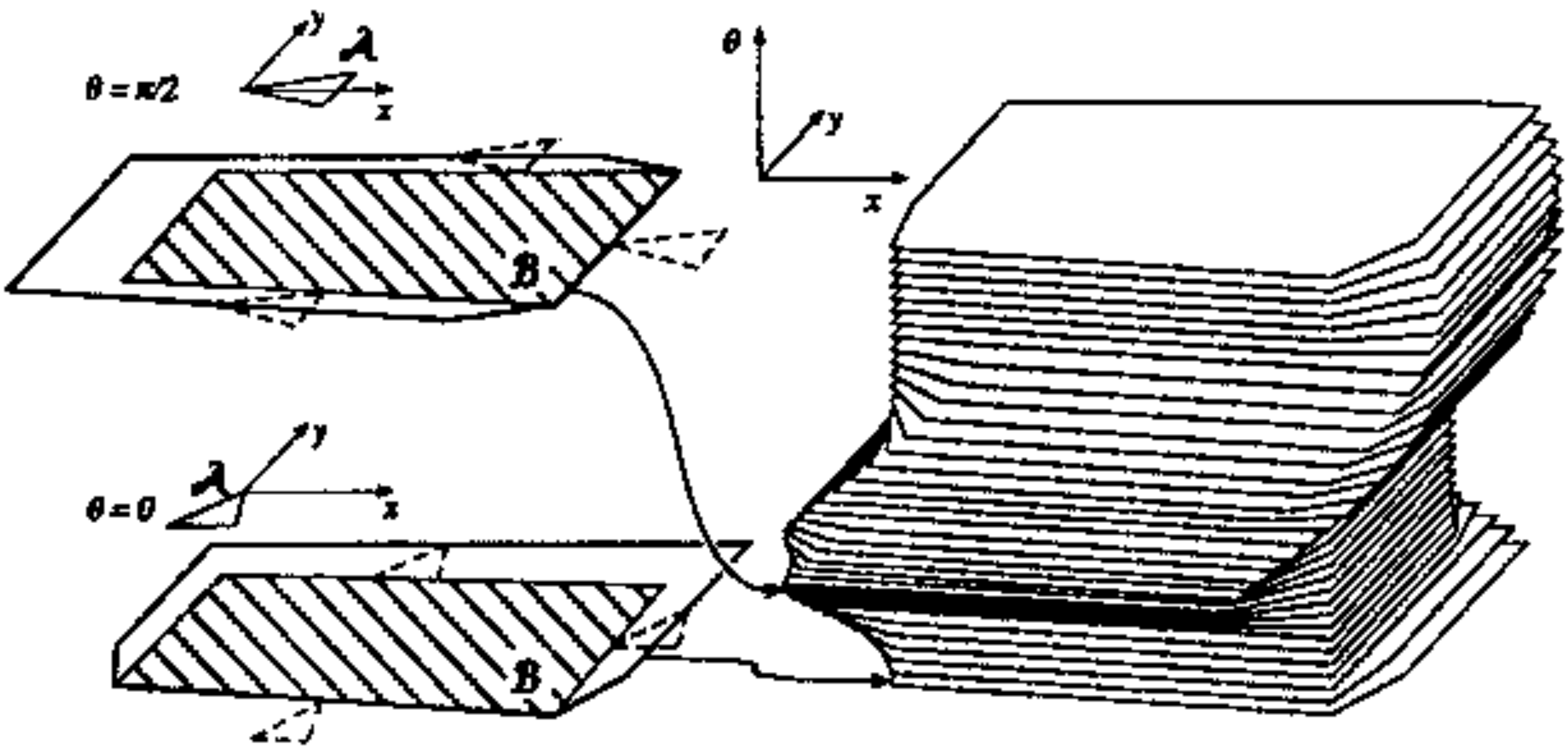


$\theta = \theta_1$



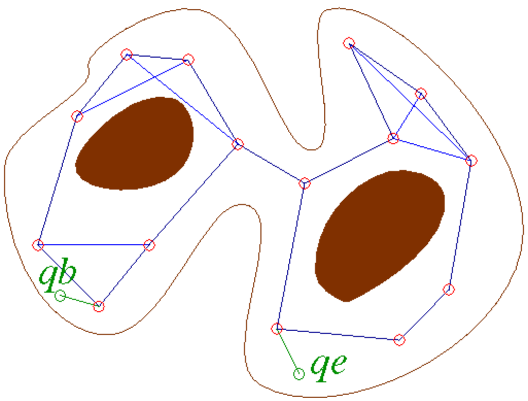
$\theta = \theta_2$

Rigid Robot Translating and Rotating in 2-D

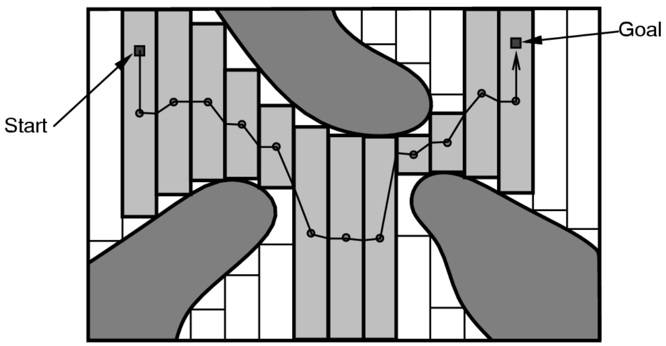


Four Methods for Path Planning in Configuration Space

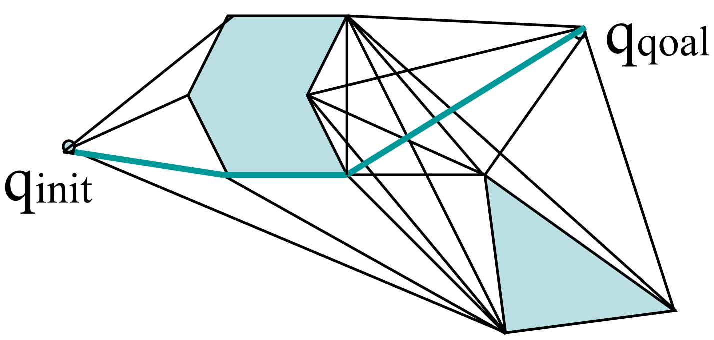
Probabilistic Roadmap



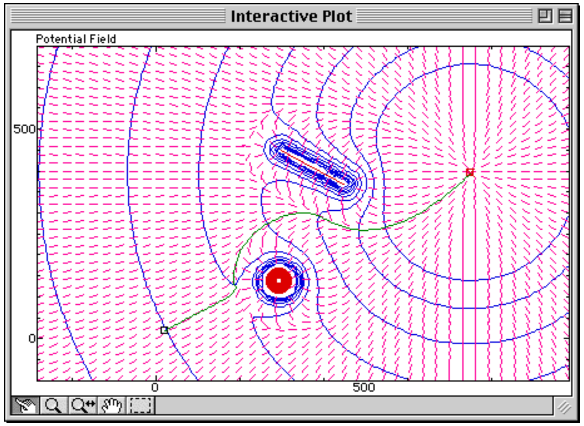
Cell Decomposition



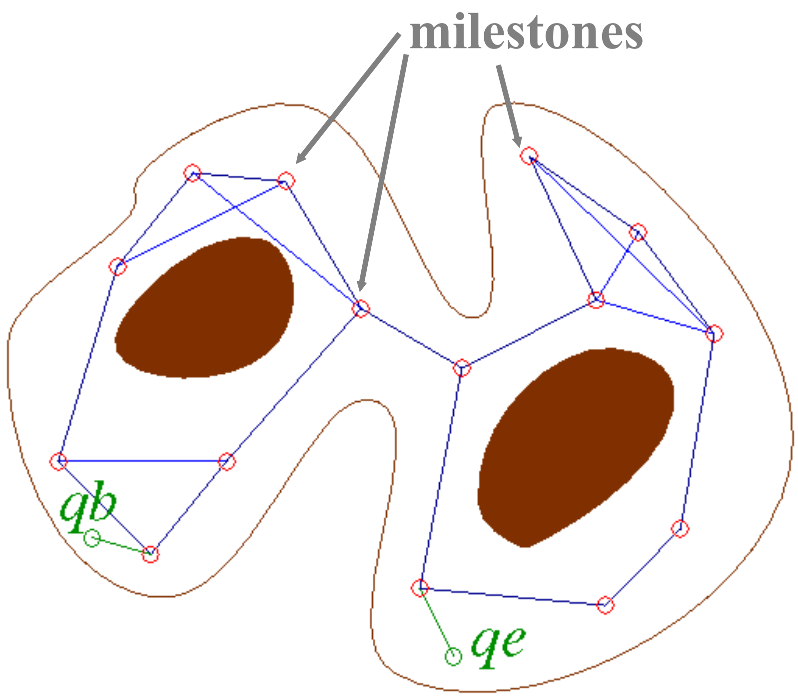
Visibility Graphs



Potential Fields



Probabilistic Roadmap

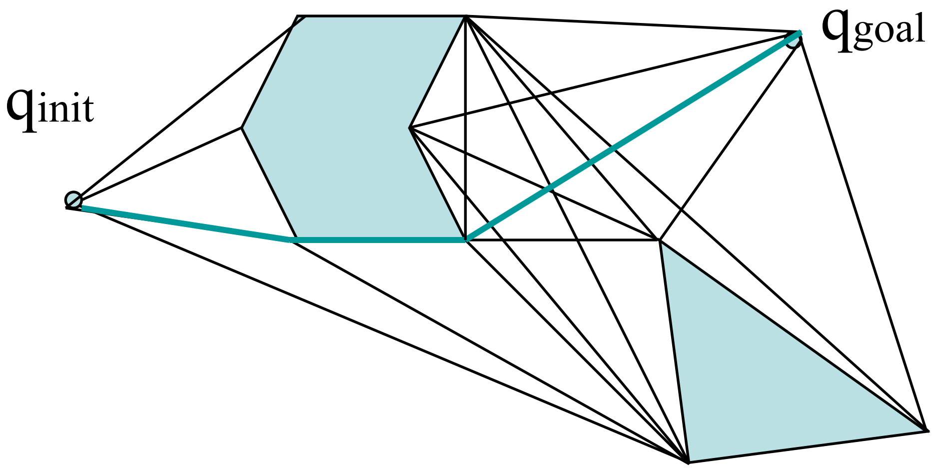


- Provide an effective computational framework to plan motions of robots with many degrees of freedom.
- Principle is very simple:
  - sample the configuration space at random
  - keep the samples in the free space
  - connect pairs of samples by simple paths

Problems with Probabilistic Roadmaps

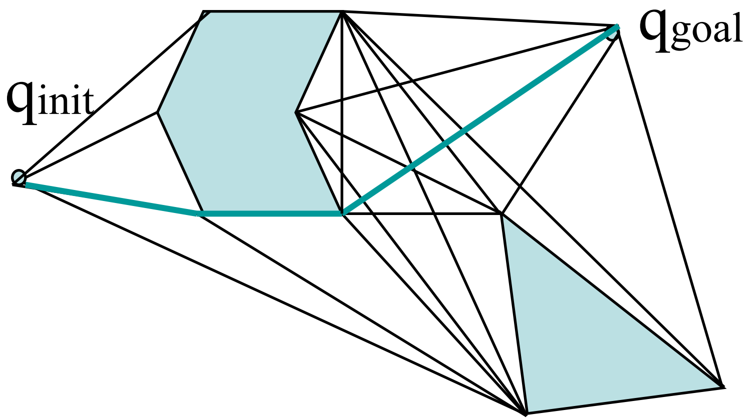
- Sampling strategy is critical
- Incomplete
  - No guarantee of performance
  - How fast does it converge?
- Still can be an effective tool for robots with many degrees of freedom (high dimensional configuration spaces)

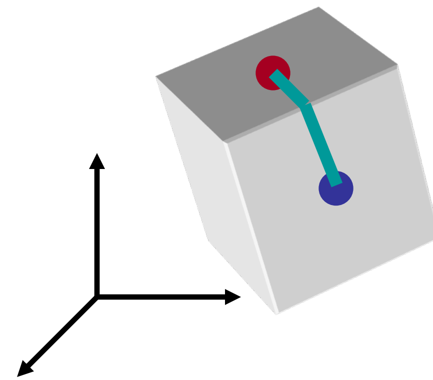
Visibility Graphs



- Create a non-directed graph *G*
- Vertices include
  - Start position
  - Goal position
  - Vertices of all C-space obstacles
- Edges
  - Straight-line segments connecting all pairs of vertices that don't go through obstacles
- Graph search problem

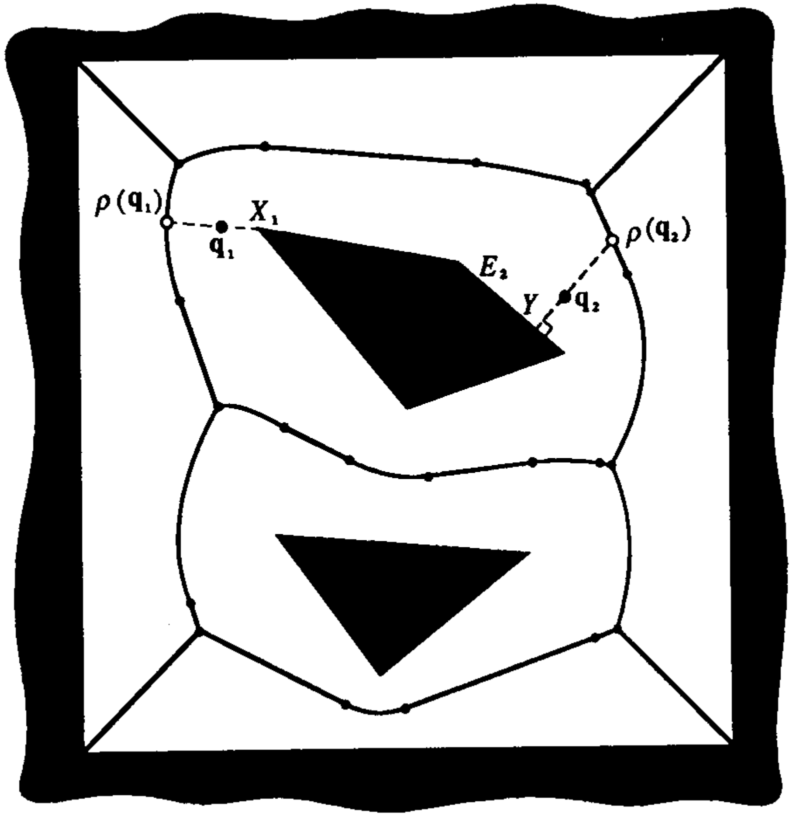
Why do Visibility Graphs Work?





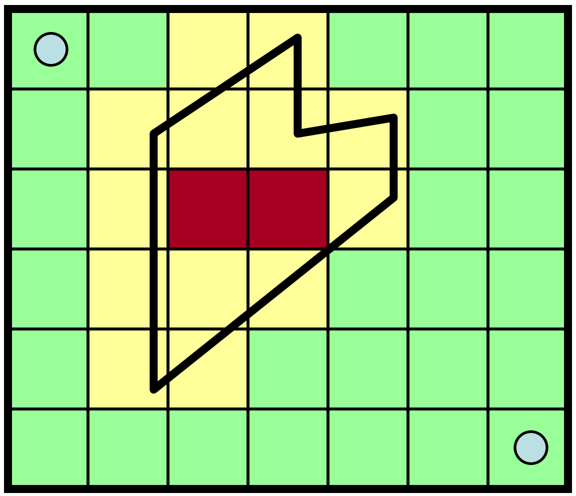
- If the path is not obscured, the shortest path is a straight line
- If you must go around an obstacle, the shortest path will be from one of the vertices
- In 3 dimensions, does this also hold?
- Shortest path is still a straight line
- But if you need to go around an obstacle, the shortest path might be from an edge

Other Skeletonization Methods: Voronoi Diagrams



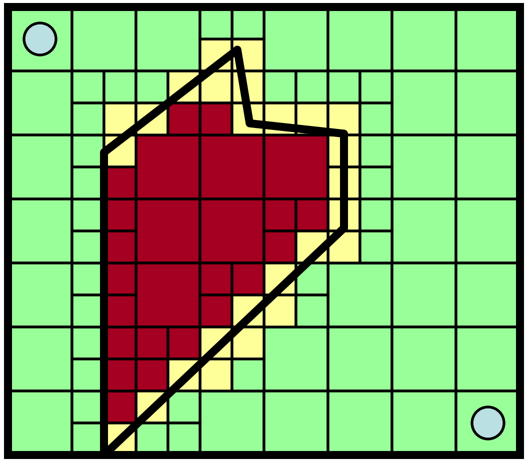
- Consists of curves in C-space that are equidistant from two or more points in the blocked space
- Maximize free space between the robot and obstacles

Cell Decomposition: Rasterization



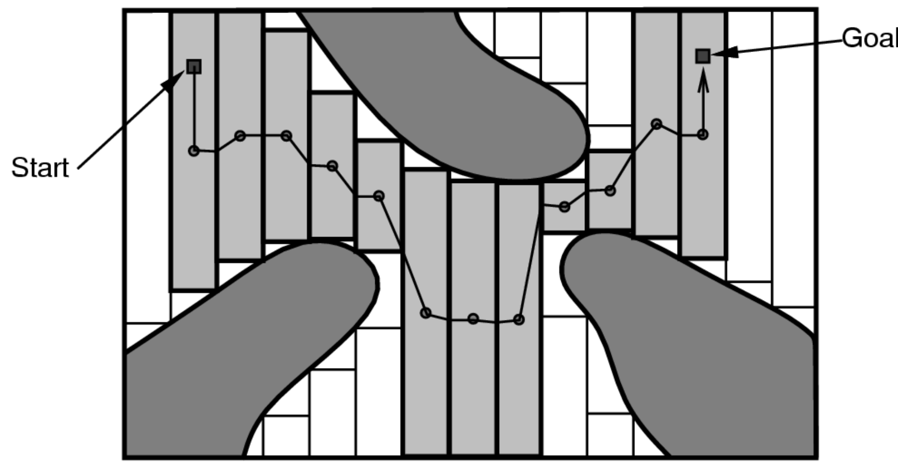
- Choose a pixel resolution
- Mark each pixel
  - *Blocked*
  - *Free*
  - *Partially-blocked*
- Only generate paths that go through free squares
- Limits the dimensionality… becomes a basic search problem

Iterative Cell Decomposition



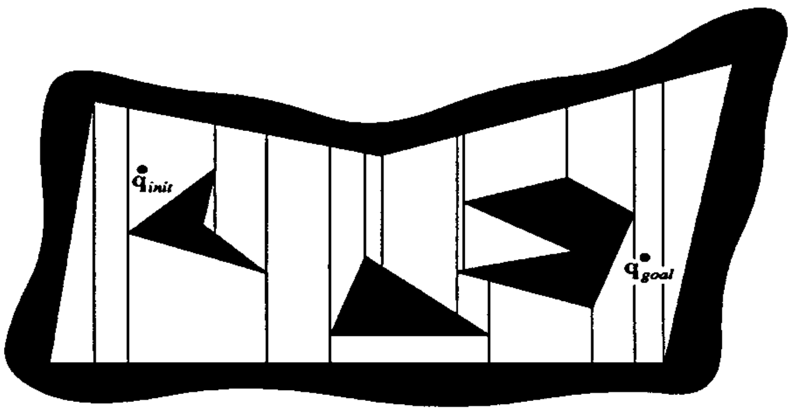
- Rasterization is non-optimal
- For partially-blocked squares, iteratively decompose them into smaller units
- Main problem: when do you stop?

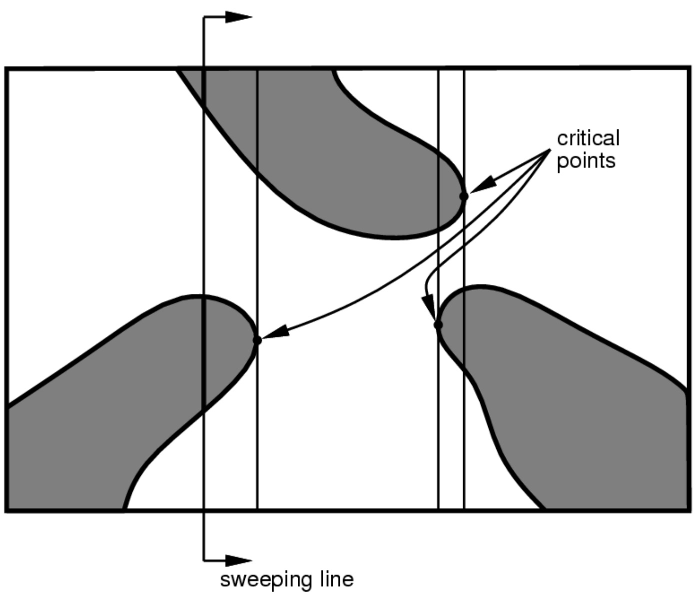
Cell Decomposition with Strips



- Use non-square cells of variable size
- Any cells that share an edge are adjacent
- Robot can navigate between adjacent cells by moving from the cell center to the midpoint of the shared edge

Cylindrical Decomposition





- Decomposition is based upon critical points
  - To find critical points, sweep a line across c-space
  - Any points where the boundary curve is vertical marks a critical point
- These points mark locations where the topology changes

Potential Field Techniques: A Way to Avoid Planning?

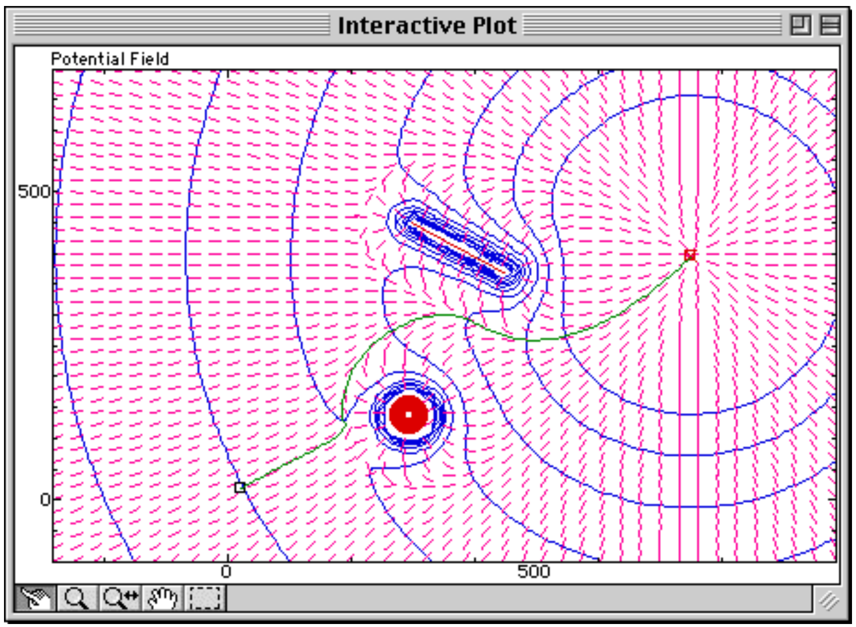


Image from calgera.com

- Potential field: scalar function over the free space
- Ideal field is smooth, with a global minimum at the goal, no local minima, and grows to infinity near obstacles
- Robot moves along with the gradient

Difficulties with Potential Fields

- Computing an ideal potential field is likely to be at least as hard as path planning itself.
- Potential fields are computed by combining forces applied to selected points, called control points, in the robot.
- Such potential fields may have local minima and must be completed by search techniques, e.g., best-first (up to 4 or 5-D configuration spaces) or random (for more dimensions).

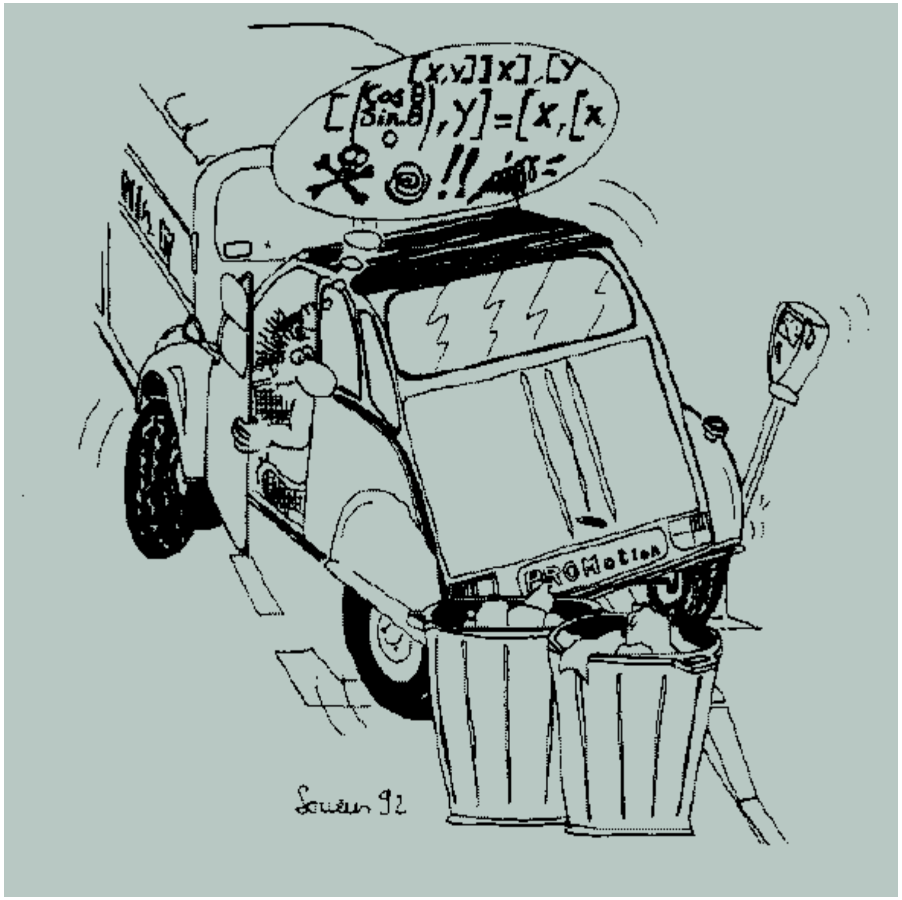
What makes this even harder?

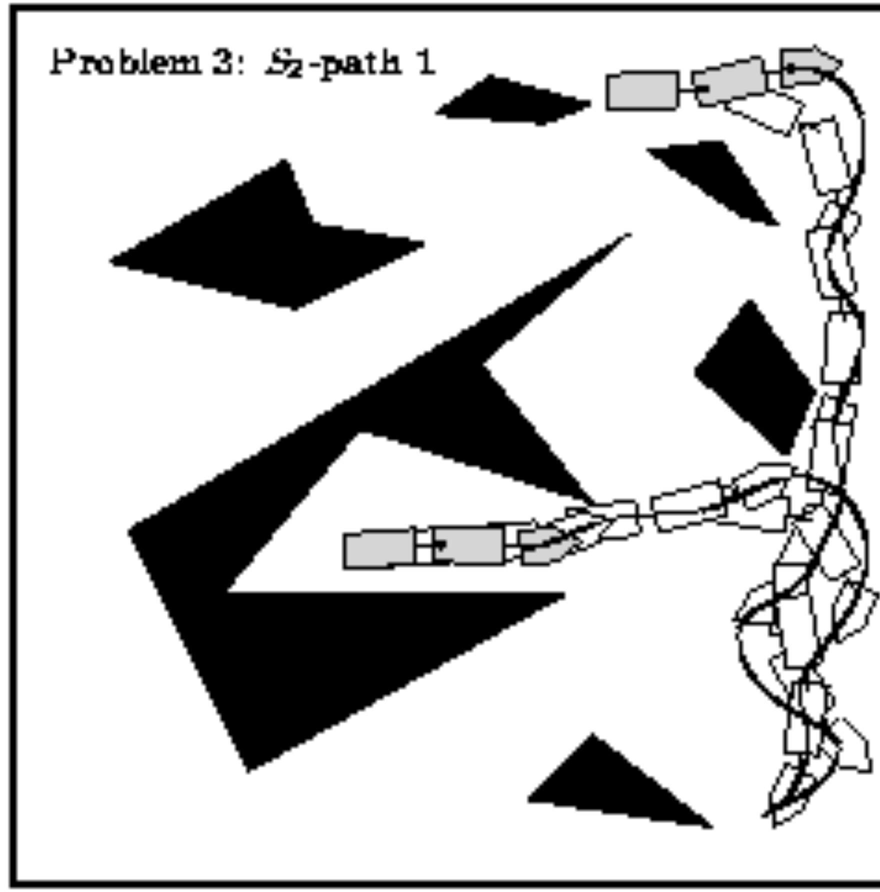
Non-holonomic robots,

Dynamic Environments,

And Uncertainty

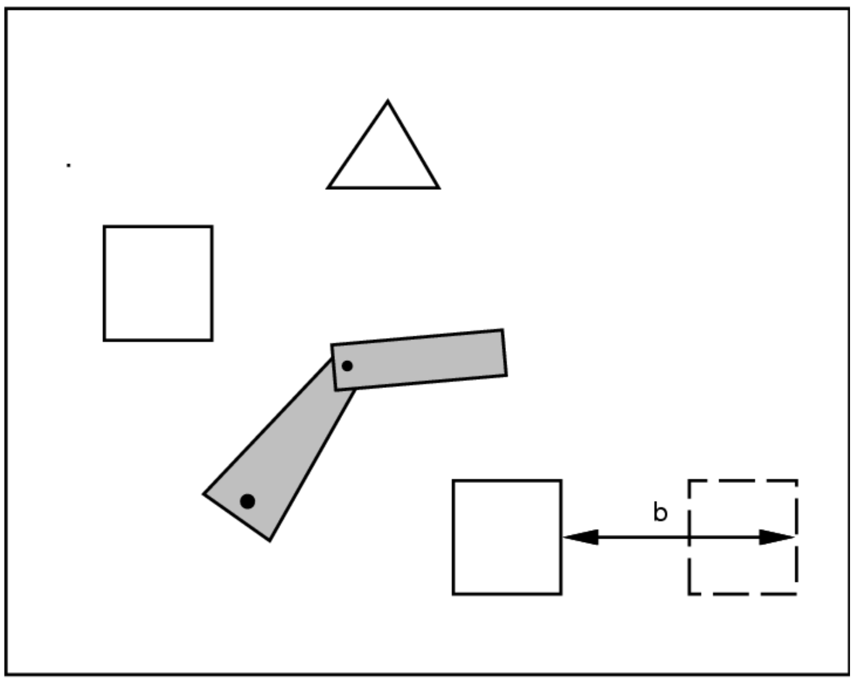
Planning for Non-holonomic Robots





Nonholonomic robots: Number of controlled DOF exceeds actual DOF

Dynamic Environments



- Impact on configuration space?
  - Increase in the dimensionality
- Solutions:
  - Convert back to a logical planning problem using abstraction
  - Plan object motions, then plan the robot's motion
  - Restrict object motions

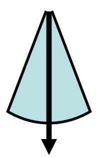
Perfect Knowledge of the World?

- What happens when you don’t know the locations of all the objects?
- What happens when your sensory systems are unreliable?
- What happens when your actuators are unreliable?

- And many more problems….

Uncertainty and Motion Planning

- What if I don't know exactly my position or velocity exactly?



Velocity uncertainty cone