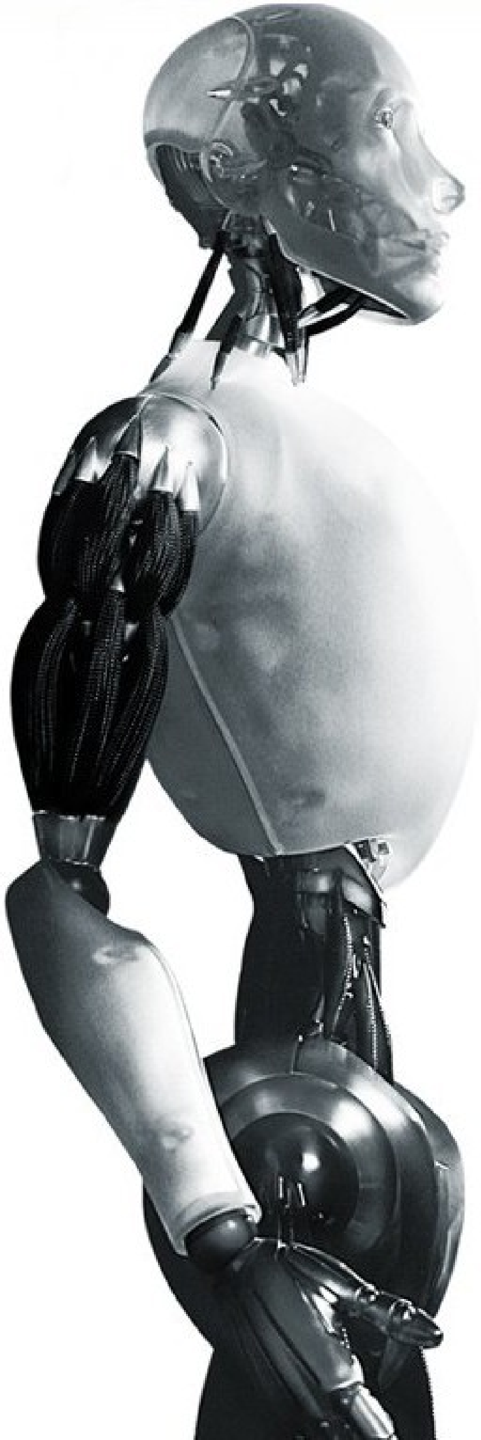
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.



Robotics

Where do we stand now?

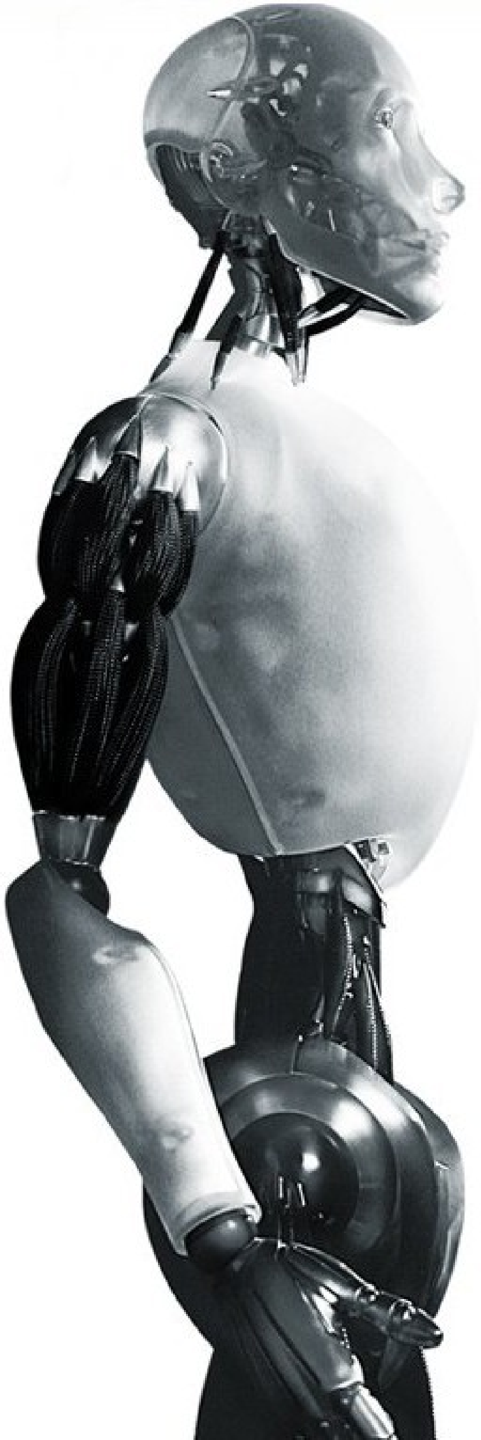
TU Berlin
Oliver Brock

Reading for this set of slides

- Planning Algorithms (Steve LaValle)
  - 4 The Configuration Space (4.1 – 4.3)
  - 5 Sampling-based Motion Planning (5.1, 5.5, 5.6, also skim the remaining sections)

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.

Problem	Stationary Robots	Mobile Robots
Move to a given position (ignoring obstacles)	Control, artificial potential fields; almost always holonomic kinematics	Control; artifical potential fields; often nonholonomic kinematics
Modeling disturbances	No slip; dynamics (gravity, inertia, centrifugal and Coriolis forces); friction, stiction	Slip (difficult/impossible to model); dynamics almost always ignored
Knowing obstacles around you	3D perception problem basically unsolved (Kinect? Dense stereo?); 3D SLAM visual servoing	Laser range finder solves 2D perception problem; SLAM; visual servoing
Knowing where you are	Use special sensors (encoders);	Localization, Mapping, SLAM
Move to a given position (considering obstacles)	Local artificial potential fields, navigation functions, **global motion planning**	Local artificial potential fields, global navigation functions (NF1, NF2, harmonic potentials)



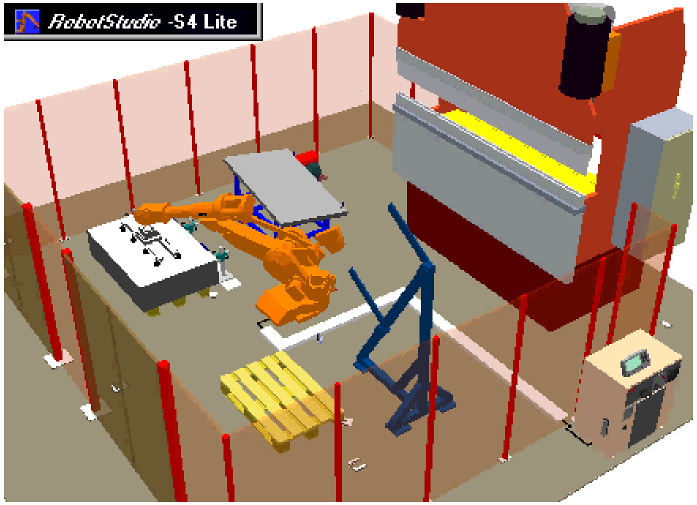
Robotics

Motion Planning

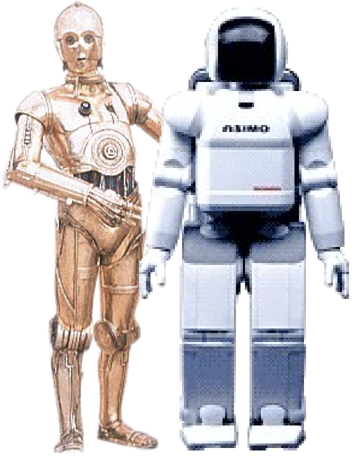
TU Berlin

Oliver Brock

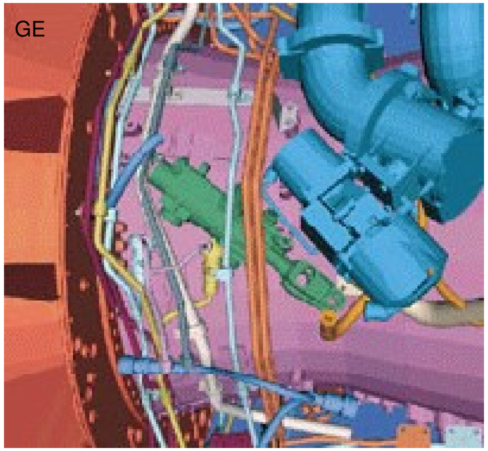
Why Motion Planning?



Robert Bohlin

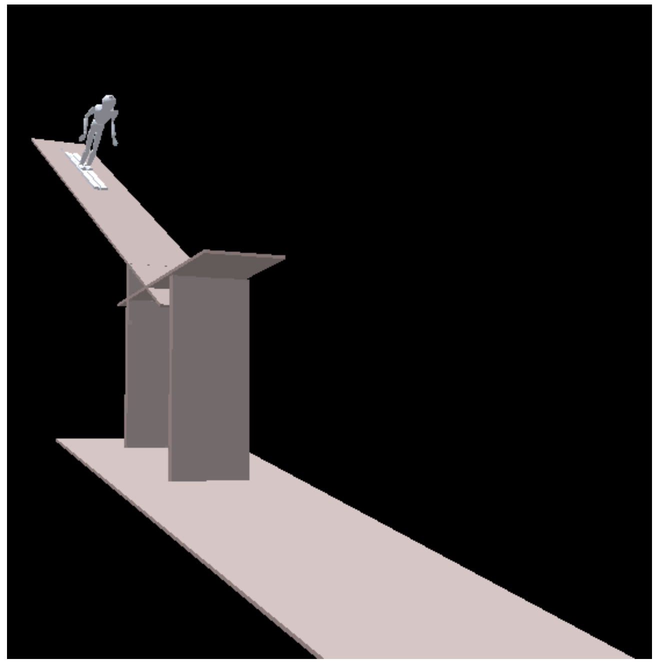


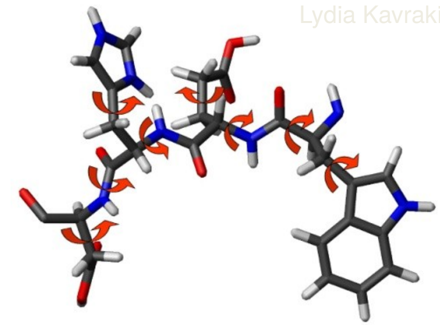
Why Motion Planning?



Virtual Prototyping

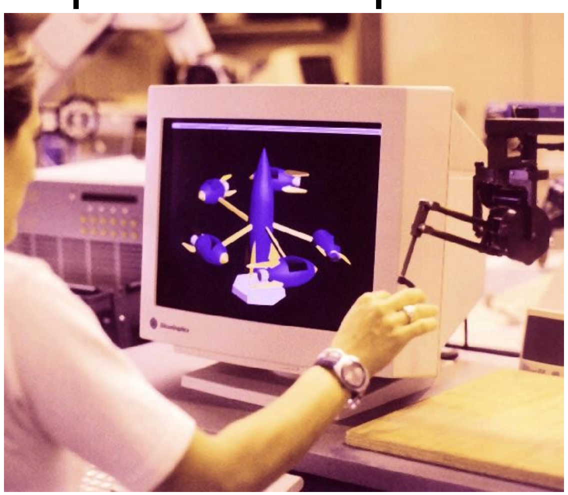
Dynamic Simulation





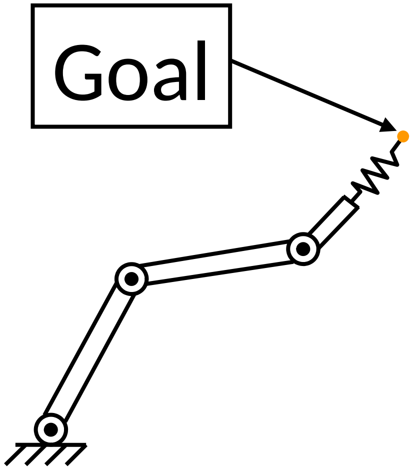
Molecular Biology

Haptics/Teleoperation



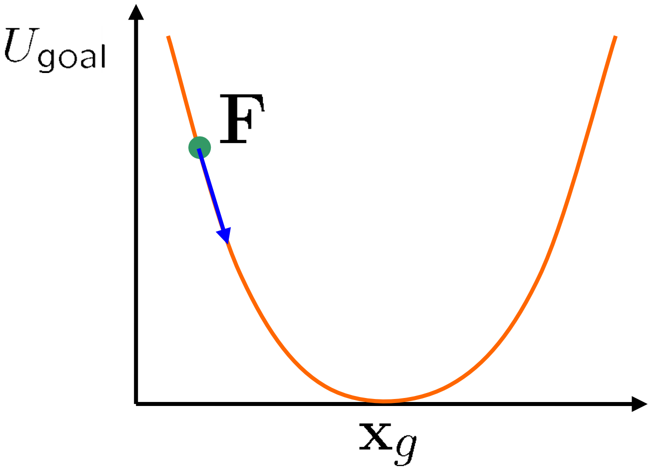
Character Animation
for Games and Movies

Reminder: Potential Function U_{goal}



$$U_{\text{goal}}(\mathbf{x}) = \frac{1}{2}\, k_p \, (\mathbf{x} - \mathbf{x}_g)^T \, (\mathbf{x} - \mathbf{x}_g)$$

$$\mathbf{F} = -\nabla U_{\text{goal}} = -k_p \, (\mathbf{x} - \mathbf{x}_g)$$



The Motion Planning Problem

- R : robot with n degrees of freedom
- $B_{1,\ldots,m}$: obstacles, rigid objects
- W : workspace R^2 or R^3
- C : configuration space
- $R, B_{1,\ldots,m} \in W$
- $q_{\text{initial}}, q_{\text{final}}$
- Find a free path τ so that R moves from q_{initial} to q_{final} without touching any B_i

Reminder: C-space

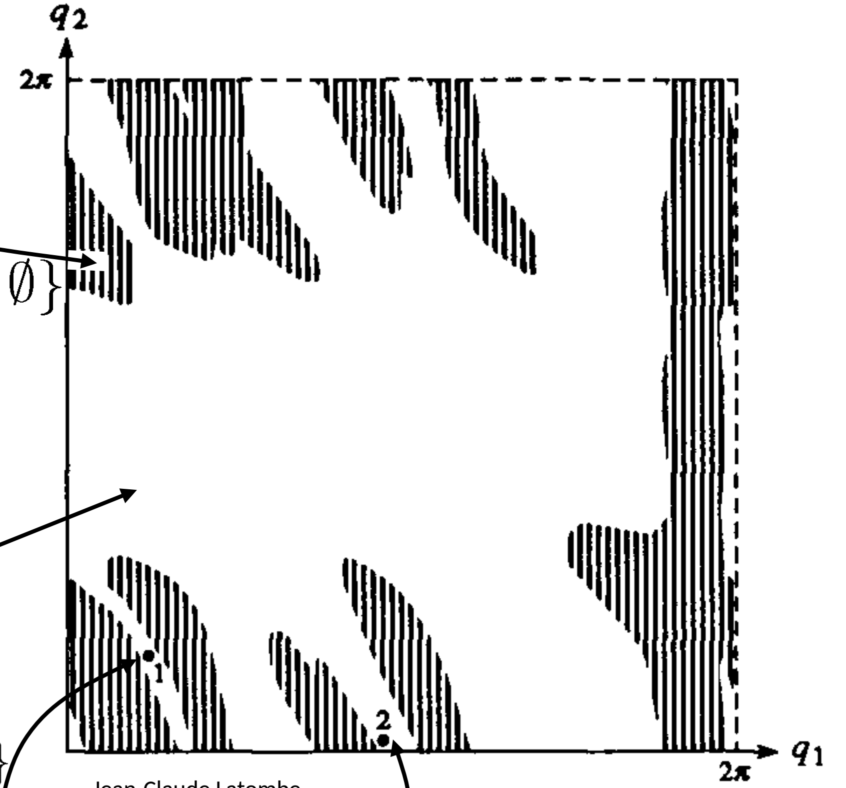
How can we obtain C_{obst}?

$$C_{\text{obst}} = \{\mathbf{q} \in C \mid \exists i : R(\mathbf{q}) \cap B_i \neq \emptyset\}$$

$$C_{B_i} = \{\mathbf{q} \in C \mid R(\mathbf{q}) \cap B_i \neq \emptyset\}$$

How can we obtain C_{free}?

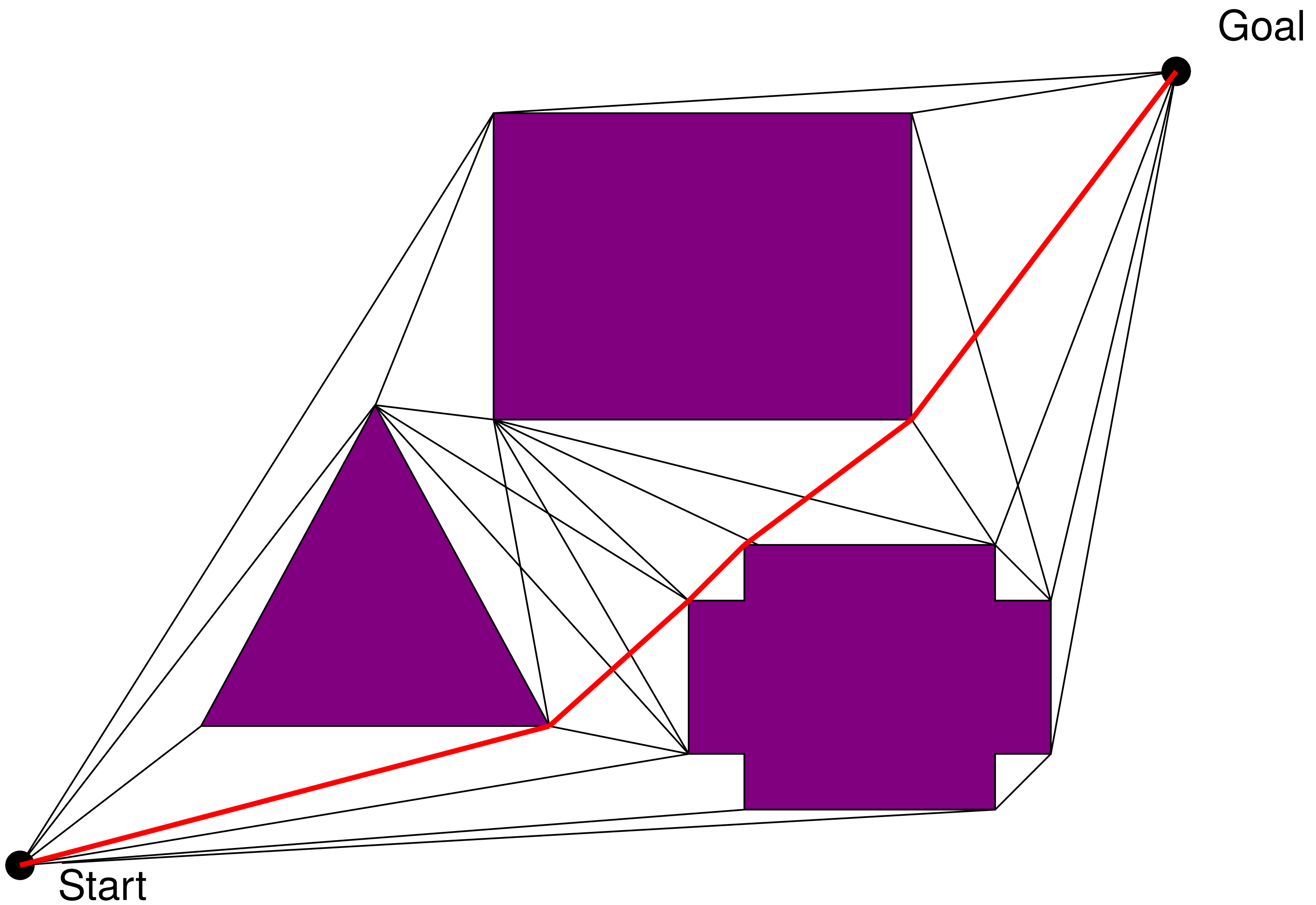
$$\begin{aligned} C_{\text{free}} &= C \setminus C_{\text{obst}} \\ &= \{\mathbf{q} \in C \mid R(\mathbf{q}) \cap (\cup_i B_i) = \emptyset\} \end{aligned}$$



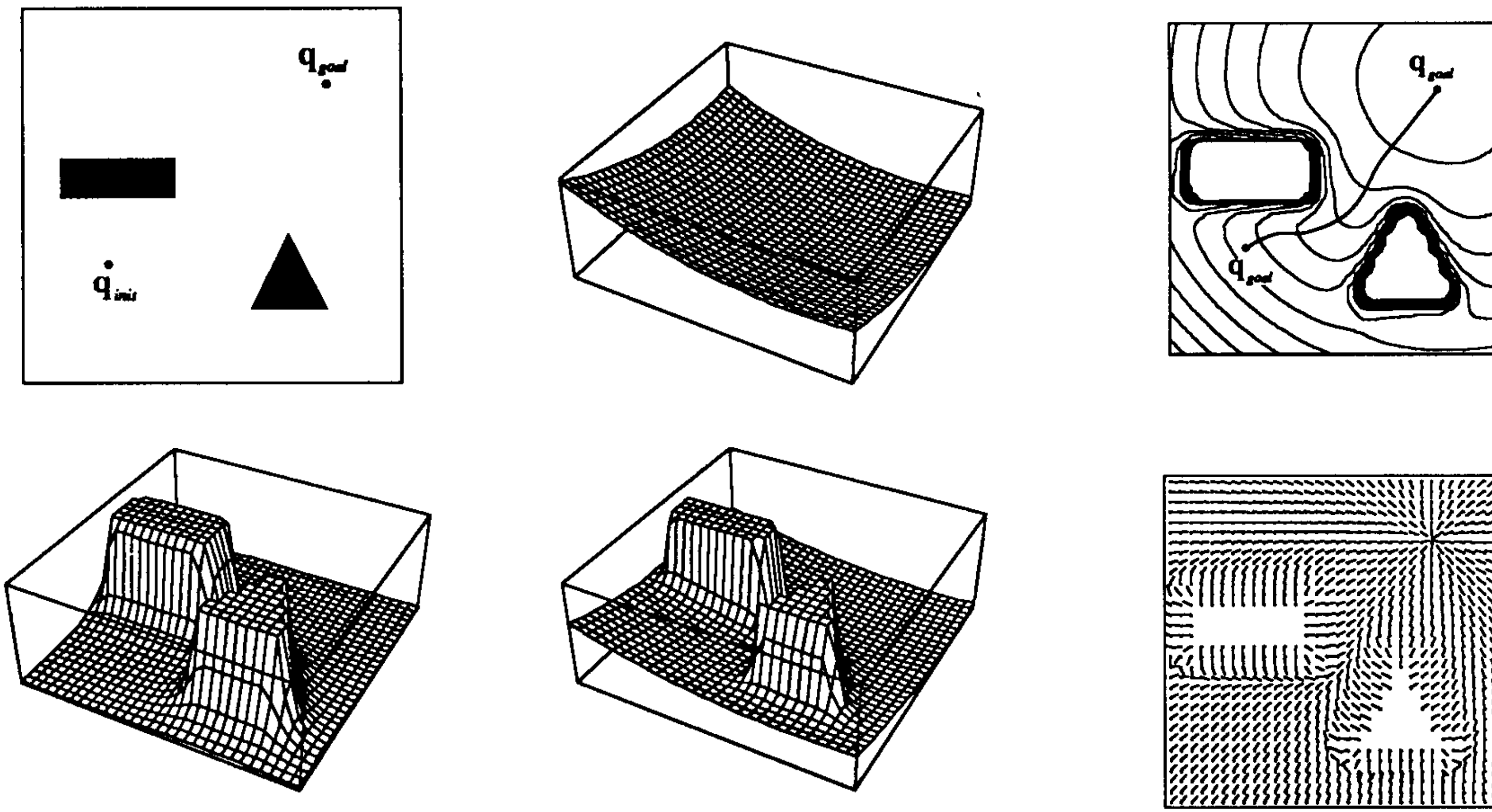
Jean-Claude Latombe

How can we obtain a path from 1 to 2?

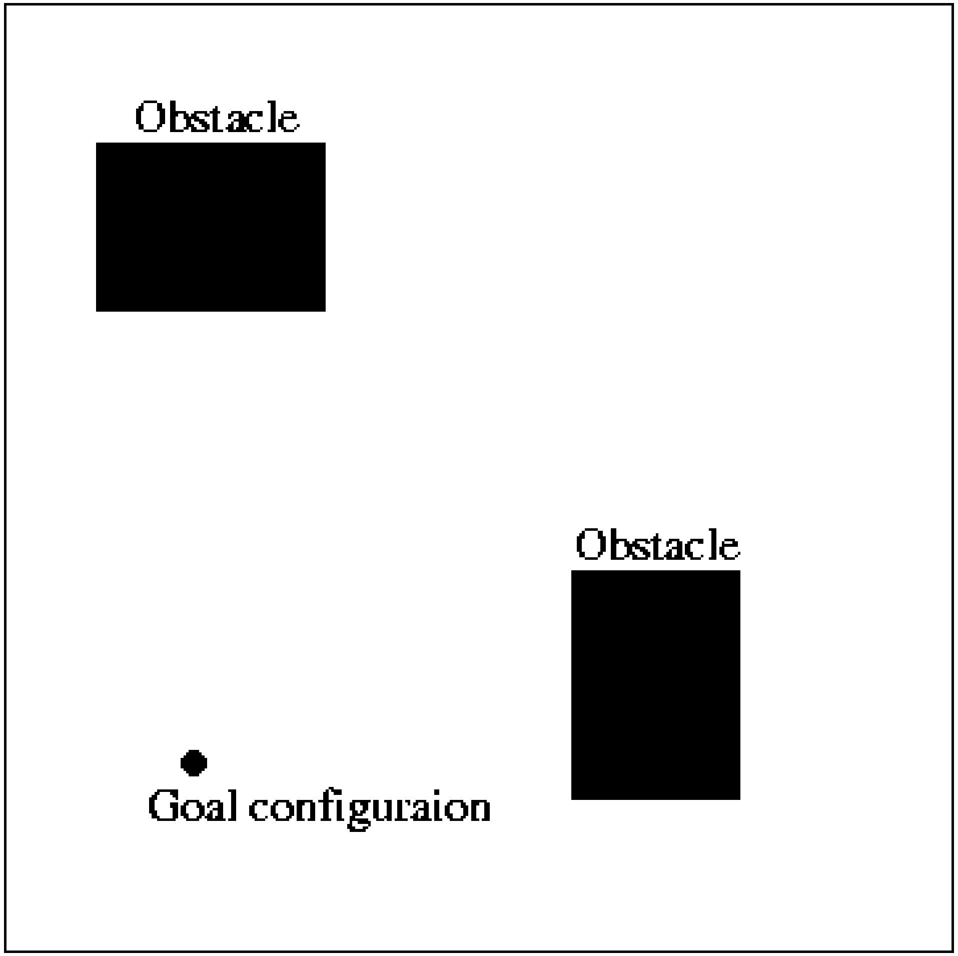
Visibility Map



Potential Field Approach

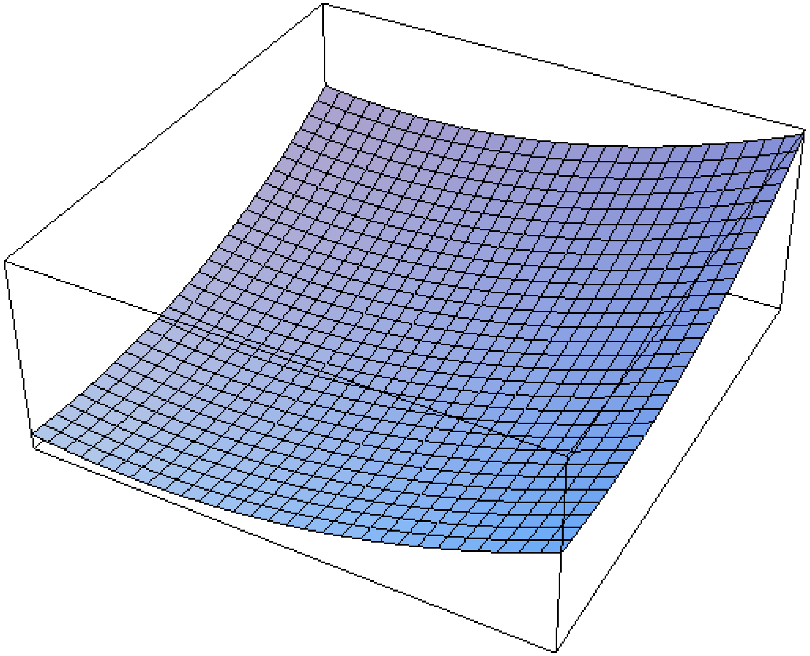


Potential Field Approach



Attractive Potential

$$U_{\mathrm{att}}(q) = \frac{1}{2} k \, \delta^2_{\mathrm{goal}}(q)$$

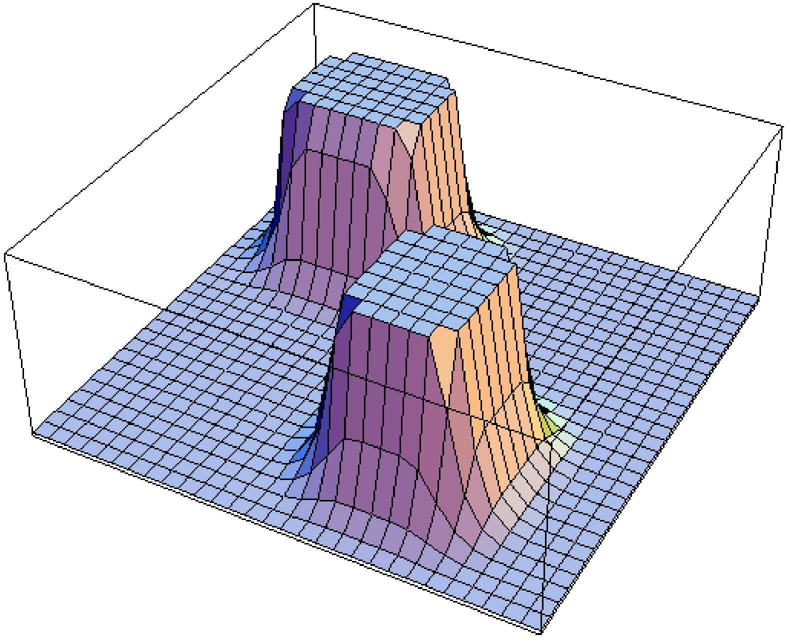


$$F_{\mathrm{att}}(q) = -\nabla U_{\mathrm{att}}(q)$$

$$= - k \, \delta_{\mathrm{goal}}(q)$$

Repulsive Potential

$$U_{\text{rep}}(q) = \frac{1}{2} k \left(\frac{1}{\delta_{obst}(q)} - \frac{1}{\delta_0} \right)^2$$

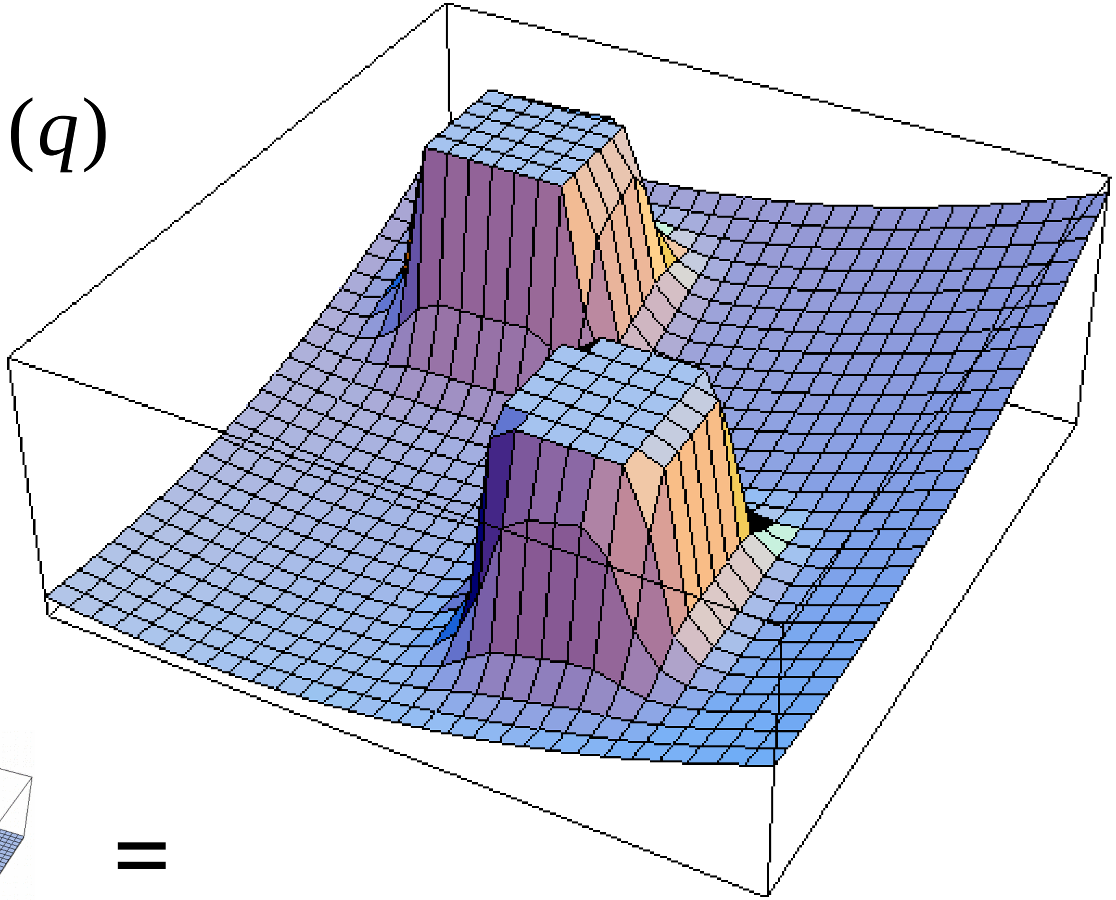


$$F_{\text{rep}}(q) = -\nabla U_{\text{rep}}(q)$$

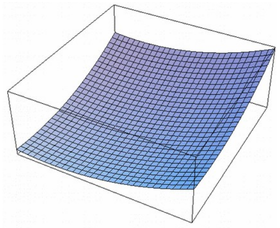
$$= -k \left(\frac{1}{\delta_{obst}(q)} - \frac{1}{\delta_0} \right)$$

Total Potential Function

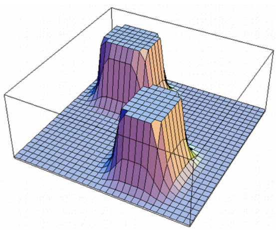
$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q)$$



$$F(q) = -\nabla U(q)$$

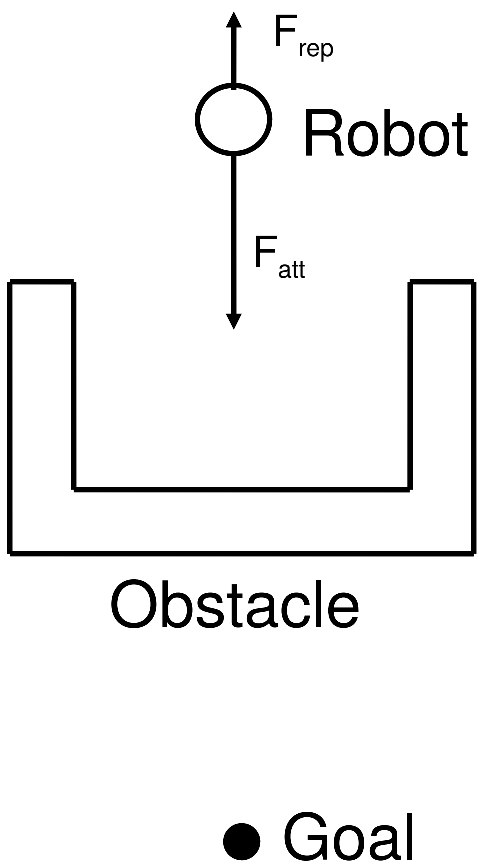


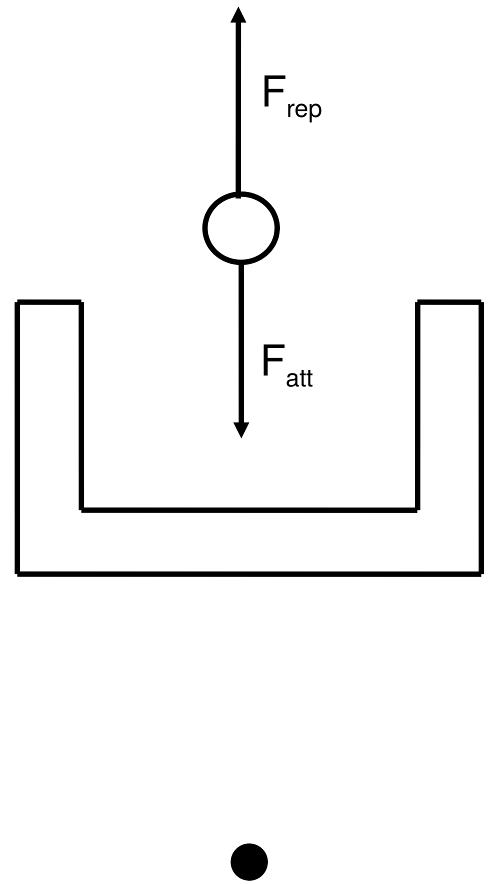
+



=

Local Minimum



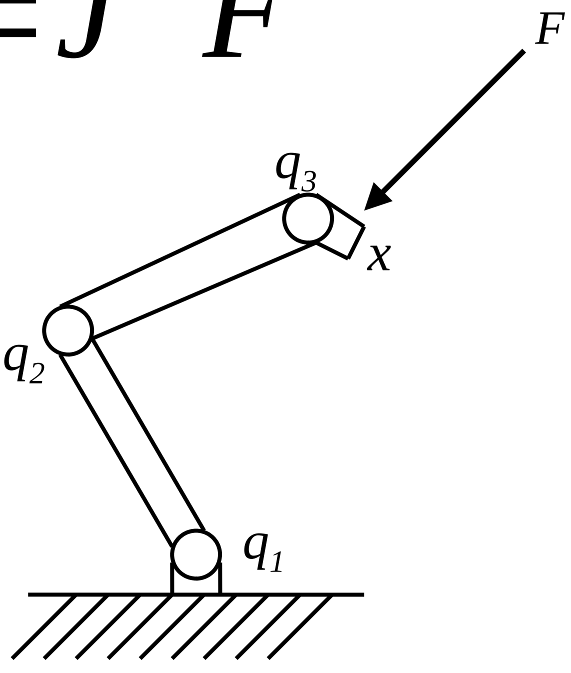


In High-Dimensional C-Spaces

- In principle the same as in lower dimensions
- Do we need to construct C-obstacles?
- Repulsive
  - How to map from workspace into configuration space?
  - Where to apply on robot?
- Attractive
  - In workspace? Mapping into C-space required!
  - In C-space? Might not be intuitive.

Mapping into C-Space

$$\tau = J^T F$$

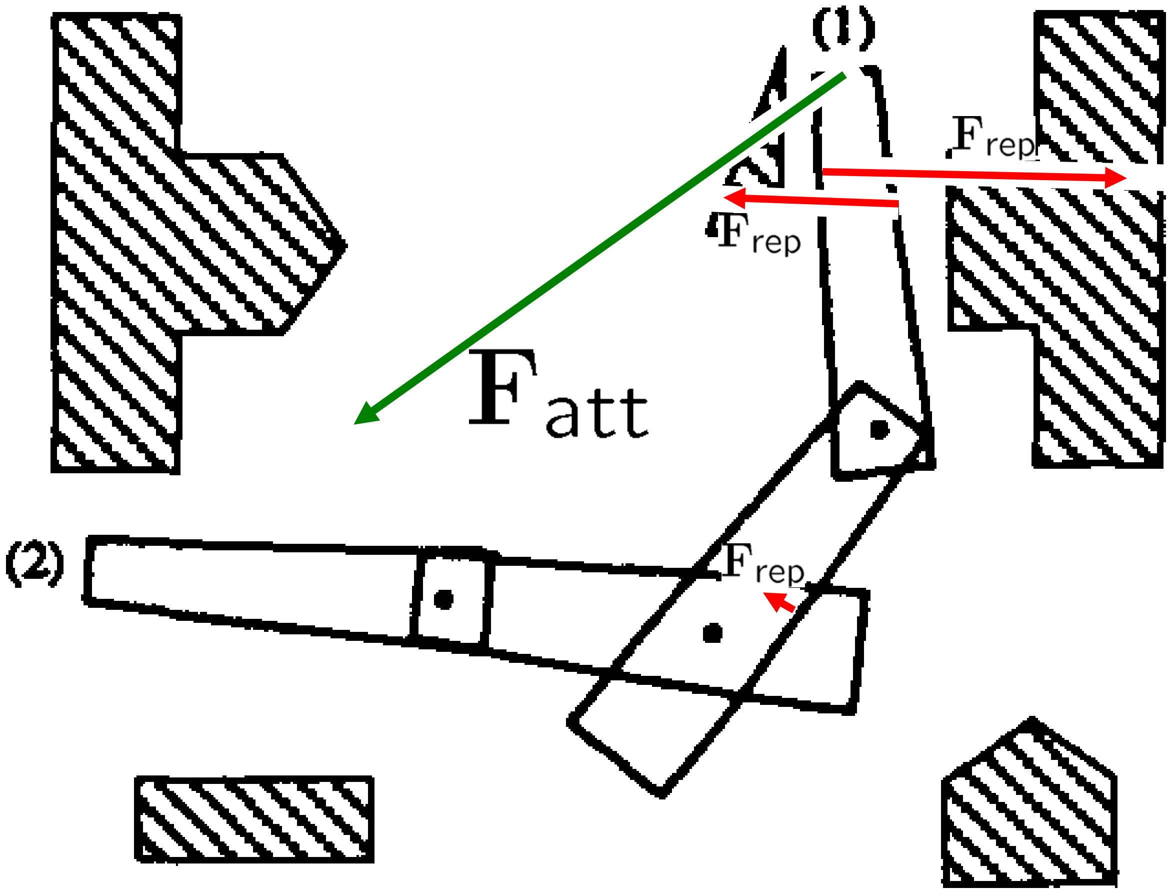


$$\delta x = J \; \delta q$$

$$\dot{x} = J \; \dot{q}$$

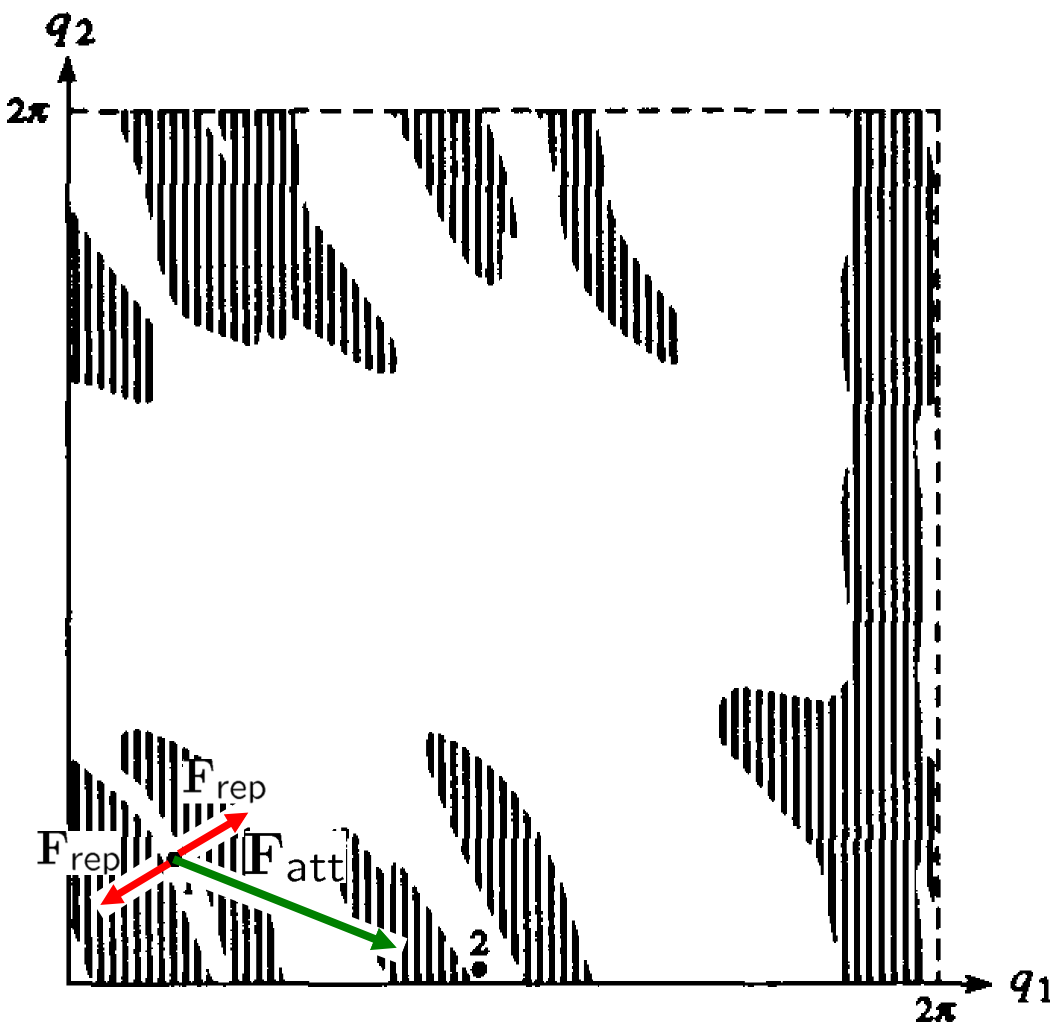
$$\dot{q} = J^{-1} \dot{x}$$

Potential Field in the Workspace



Jean-Claude Latombe

Potential Field in the C-Space

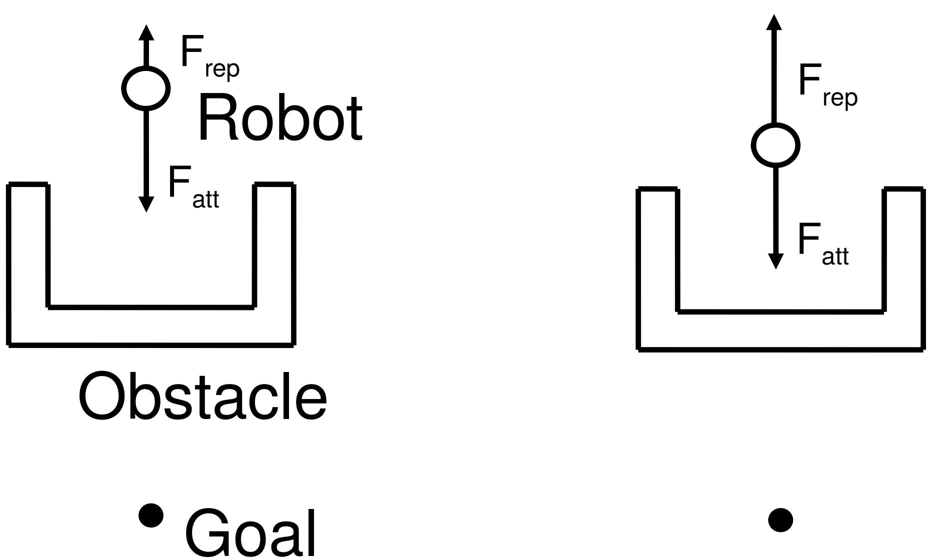


Summary: Potential Field Approach

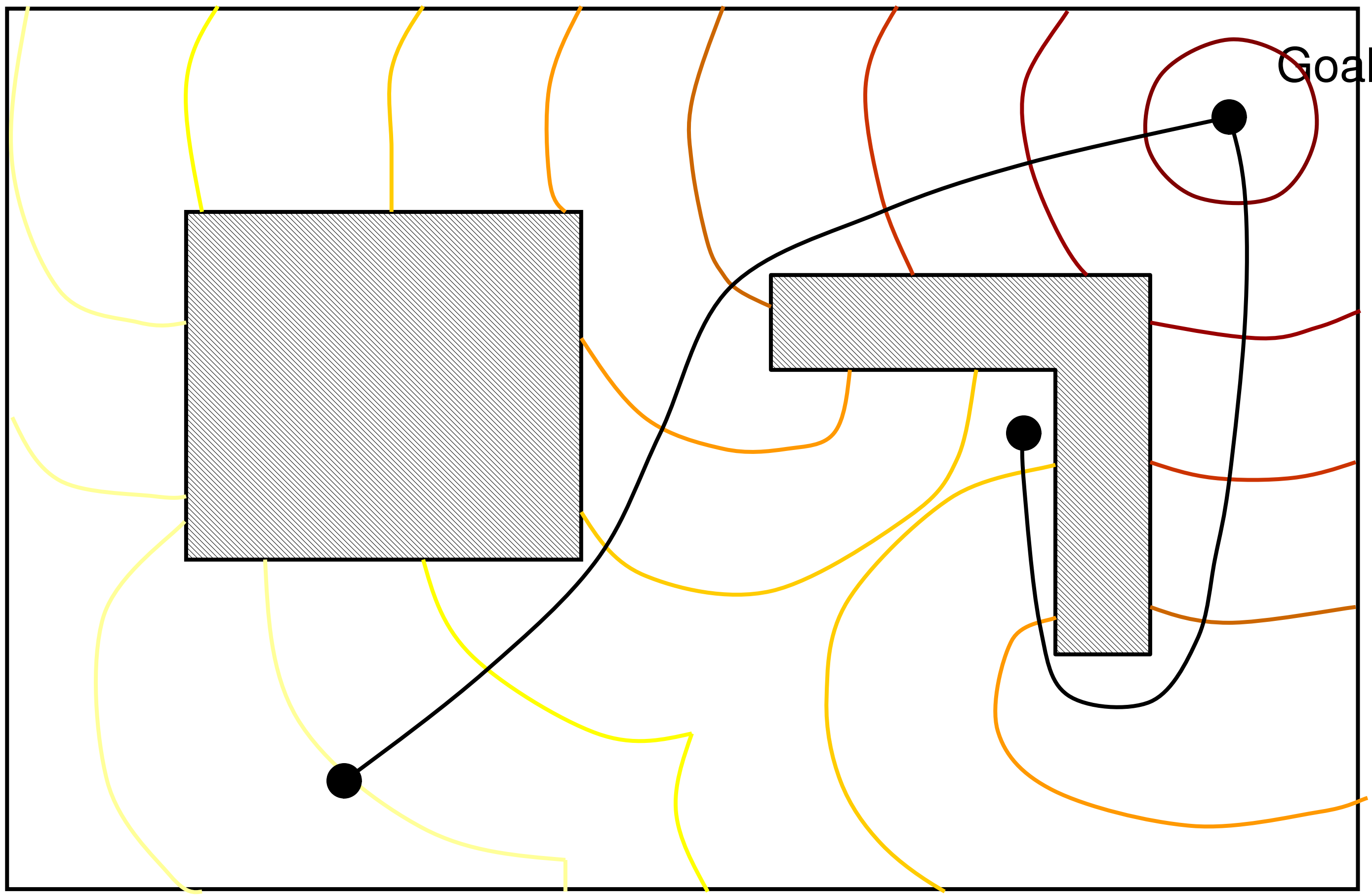
- Repulsive Forces
  - C-space
    - Requires C-obstacle computation
    - Mapping forces is 1:1
  - Work space
    - No computation of C-obstacles
    - Repulsive forces are mapped into C-space
- Attractive Forces
  - Might be undesirable in C-space
  - Might only describe partial behavior in work space
- Local approach and susceptible to local minima
- Very fast in **any** number of dimensions

Global Potential Functions

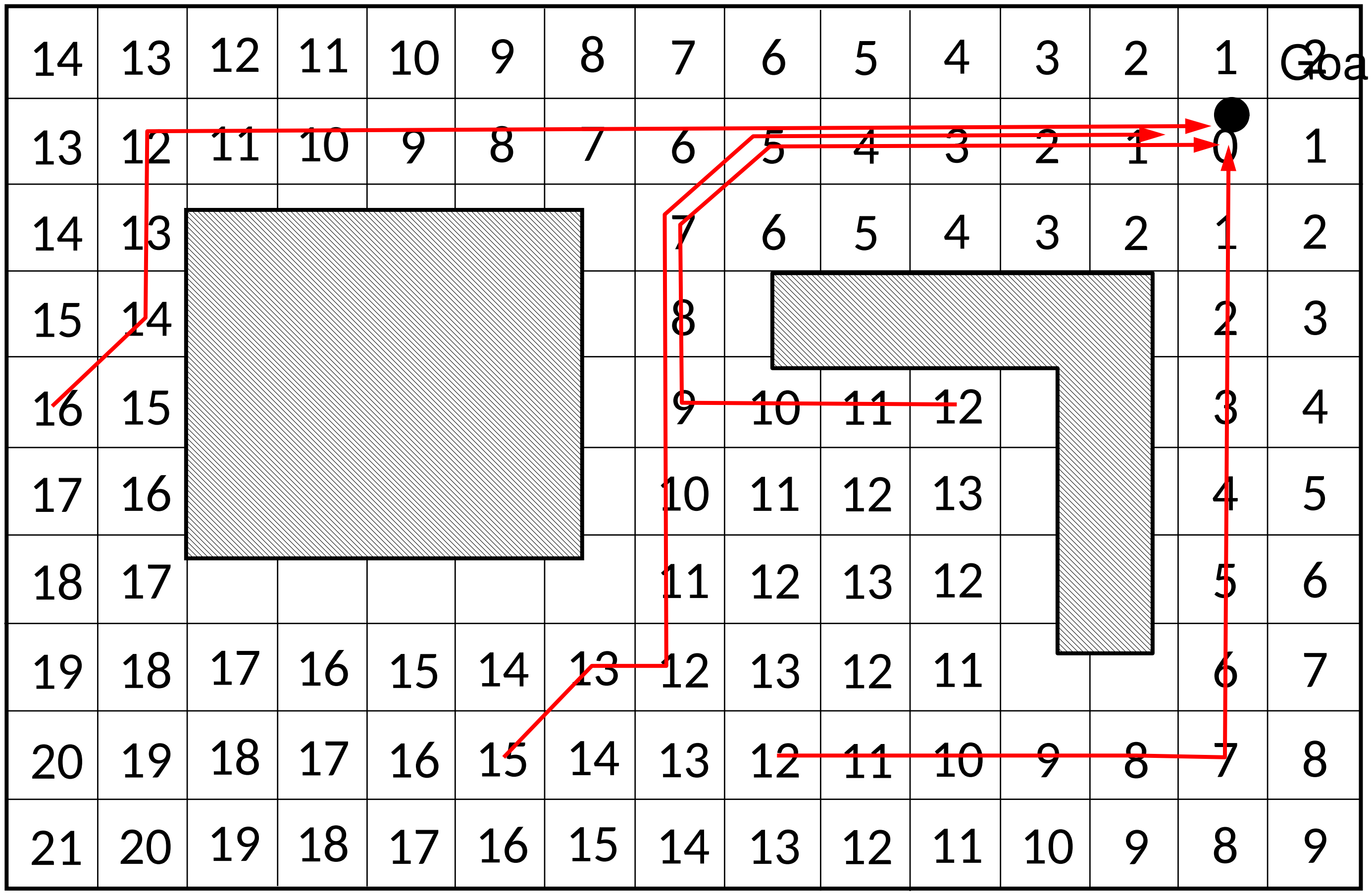
- Goal: avoid local minima
- Problem: requires global information
- Solution: **Navigation Function**



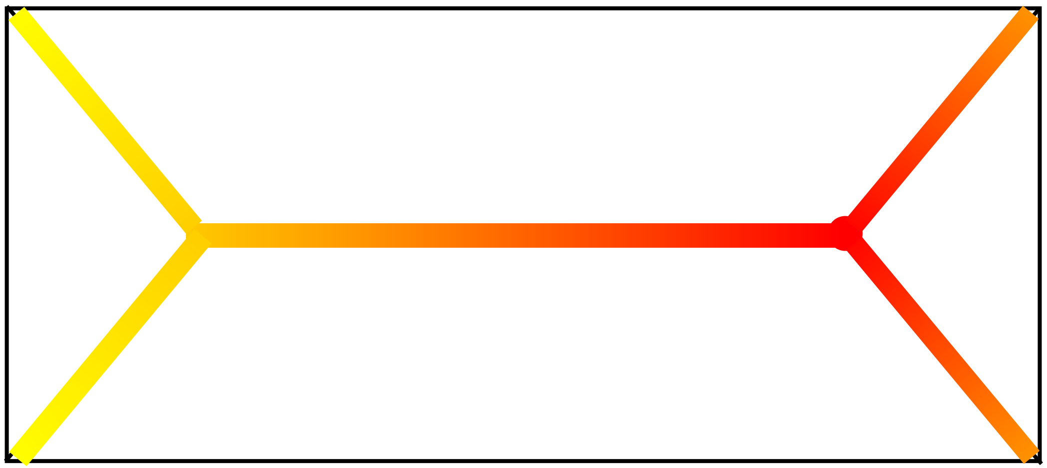
NF1



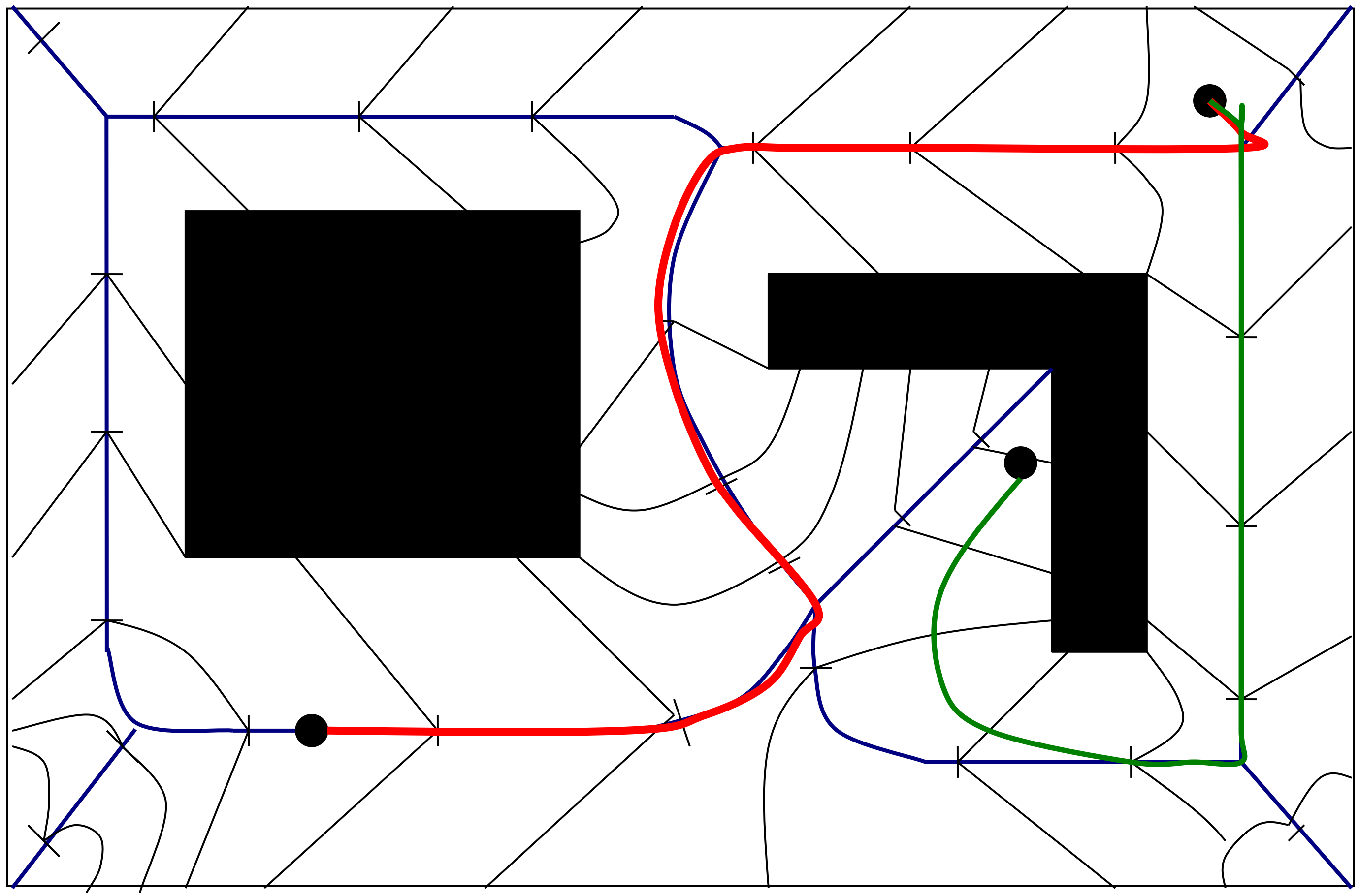
NF1 Real-World Scenario



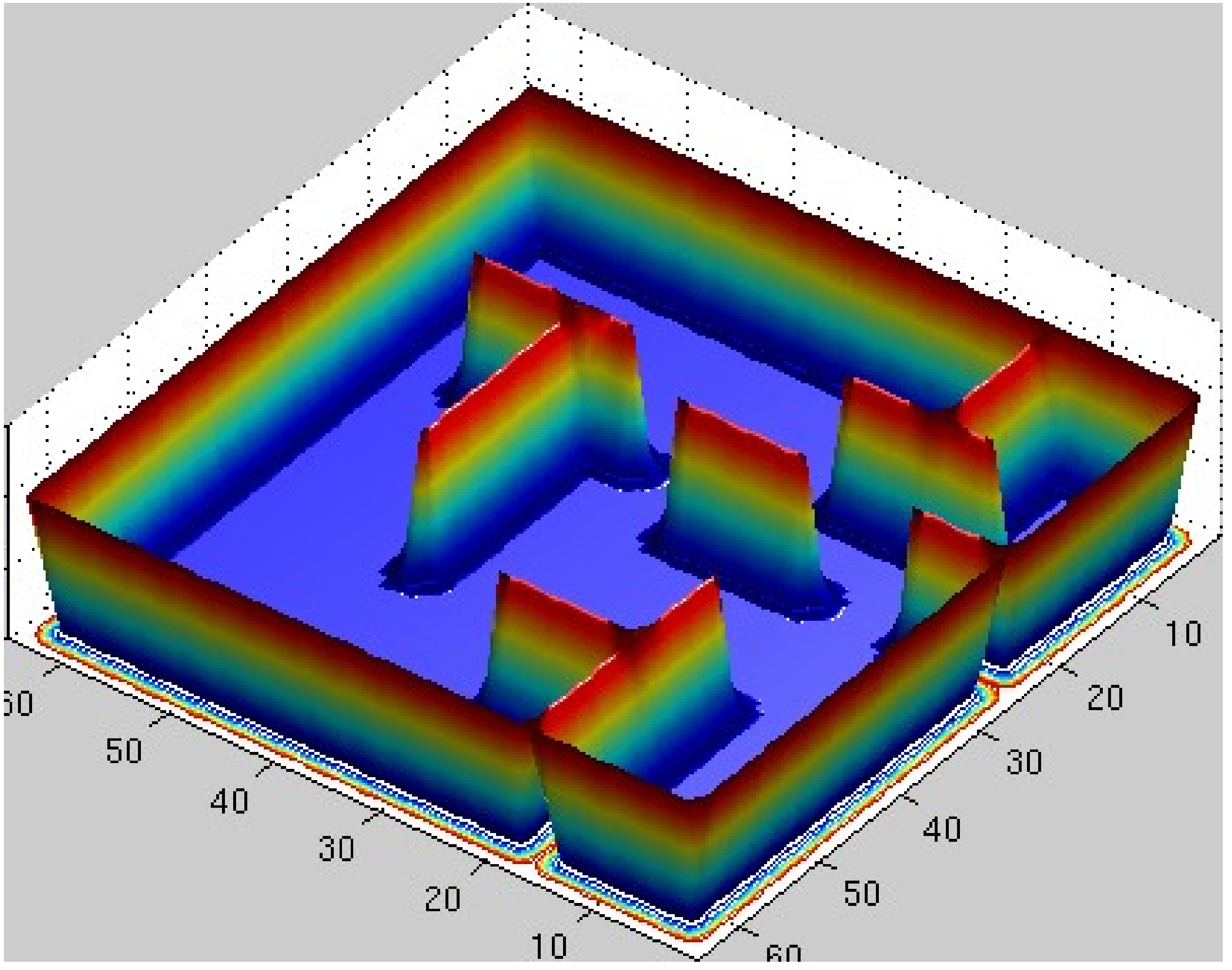
NF2



NF2



Harmonic Potentials

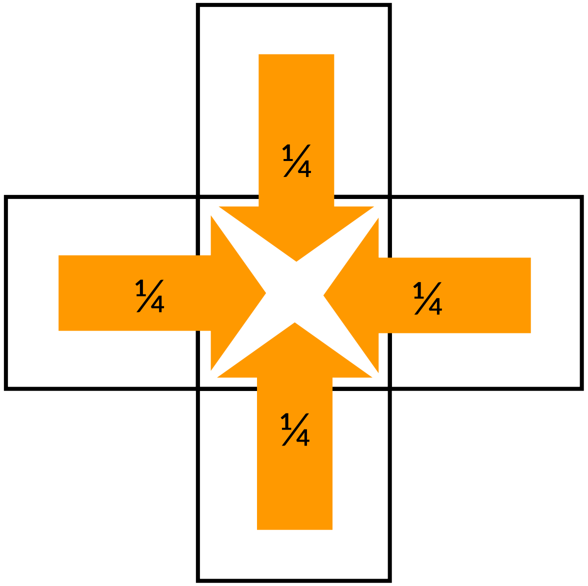


Courtesy of John Sweeney

Harmonic Potentials

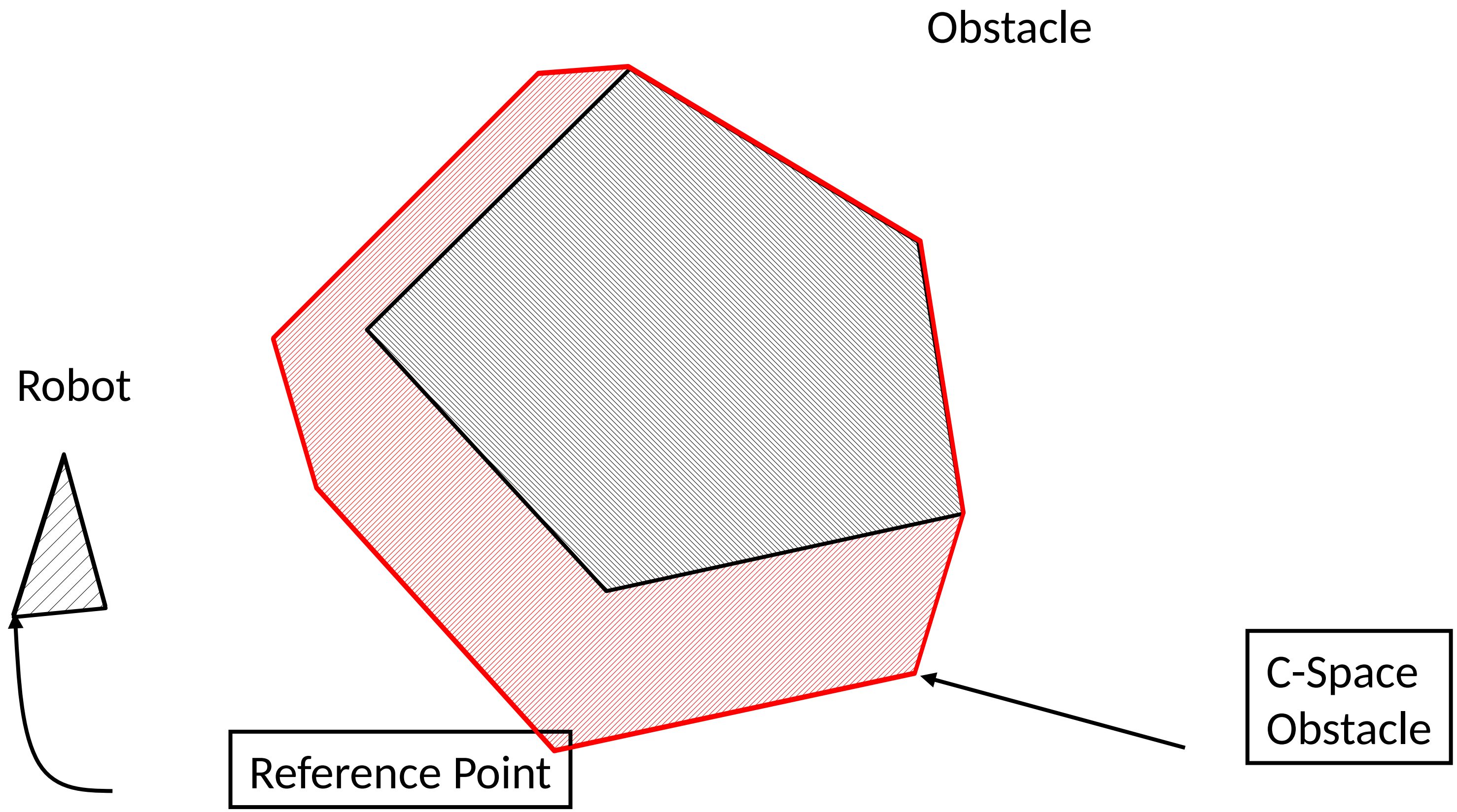
- Harmonic functions
- Solutions to **Laplace's equation** (PDE)
- Intuition: heat transfer
- Numerical solutions: dynamic programming / relaxation
- No local minima
- $O(n^{\mathrm{d}})$ computation

Example: Jacobi Iteration



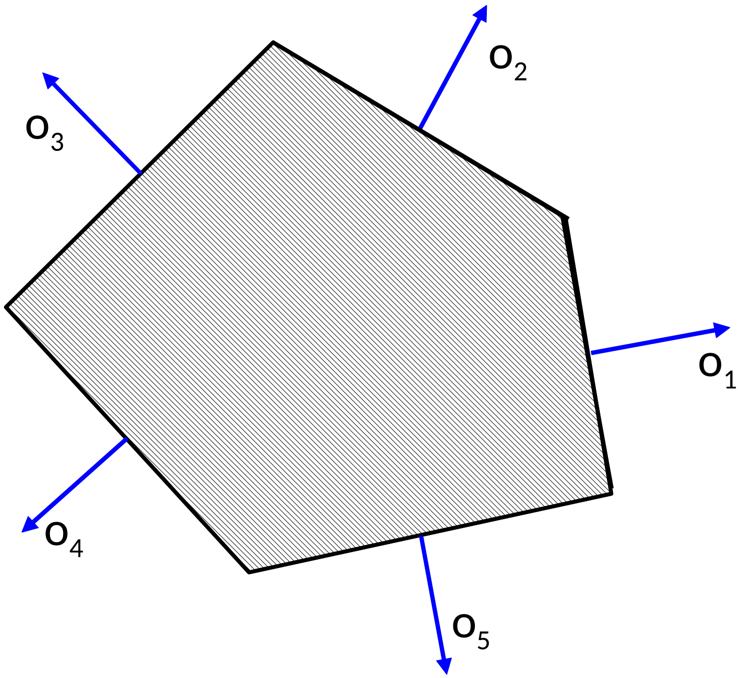
the new value of a cell for the next iteration
is ¼ of the sum of its 4-neighbors

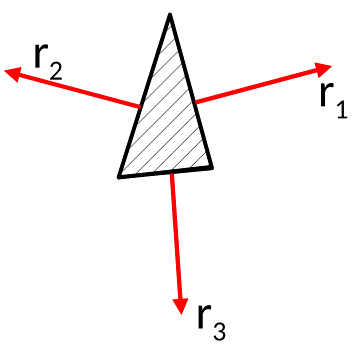
Translational Case

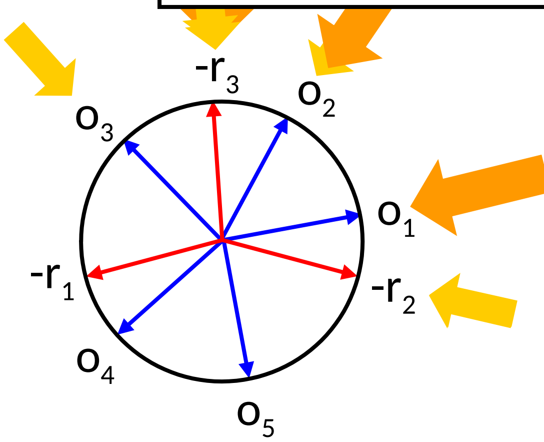


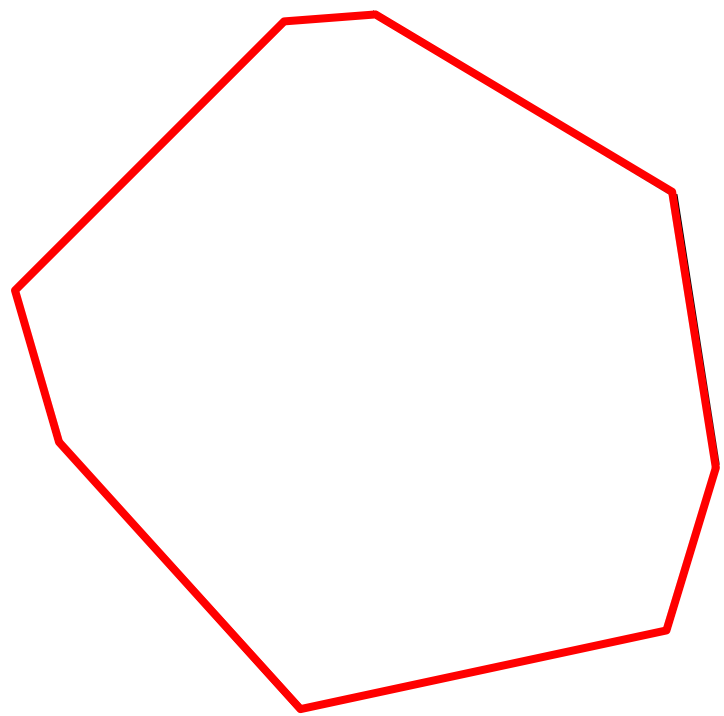
C-Obstacle Const

positive linear combination

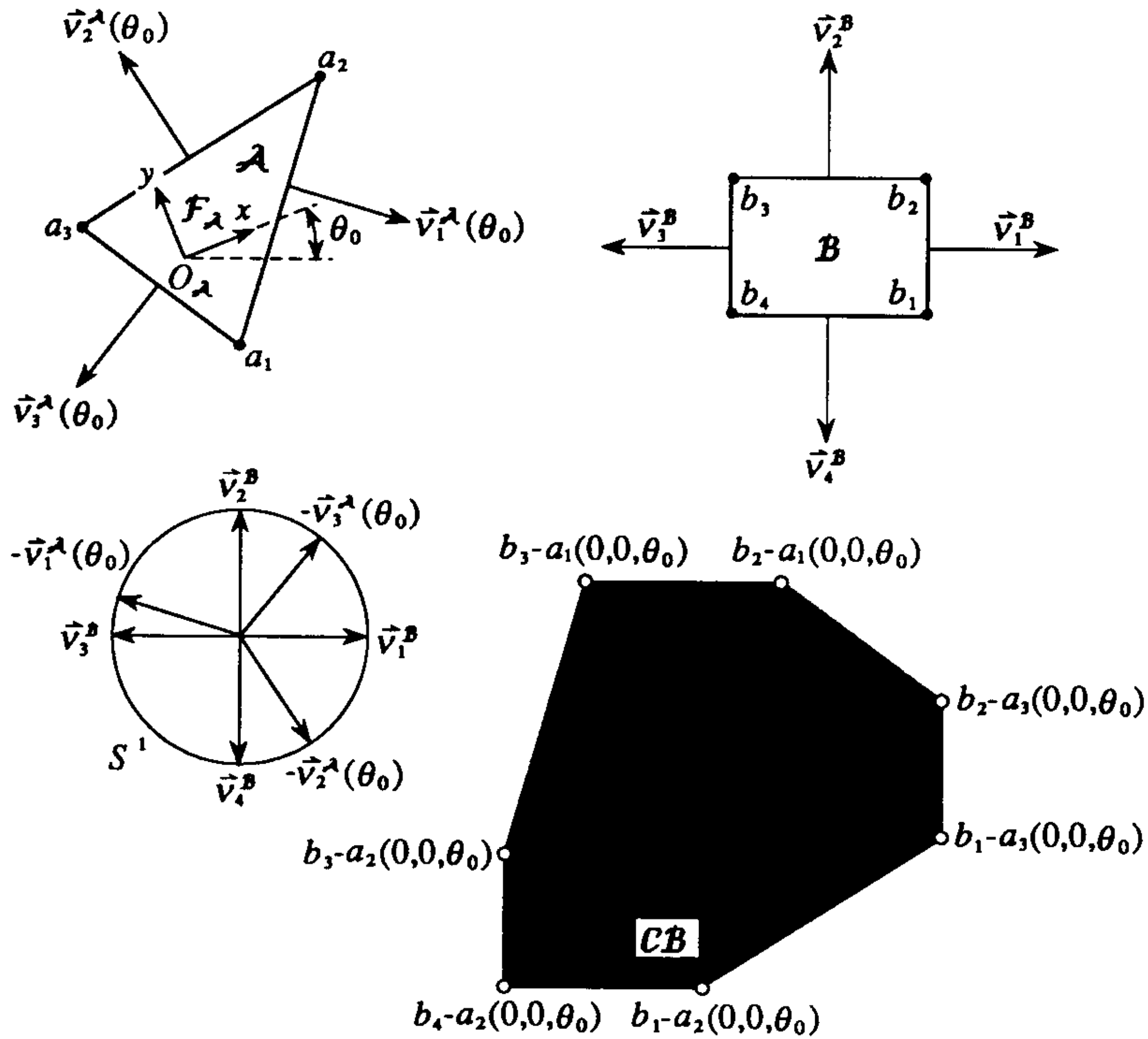




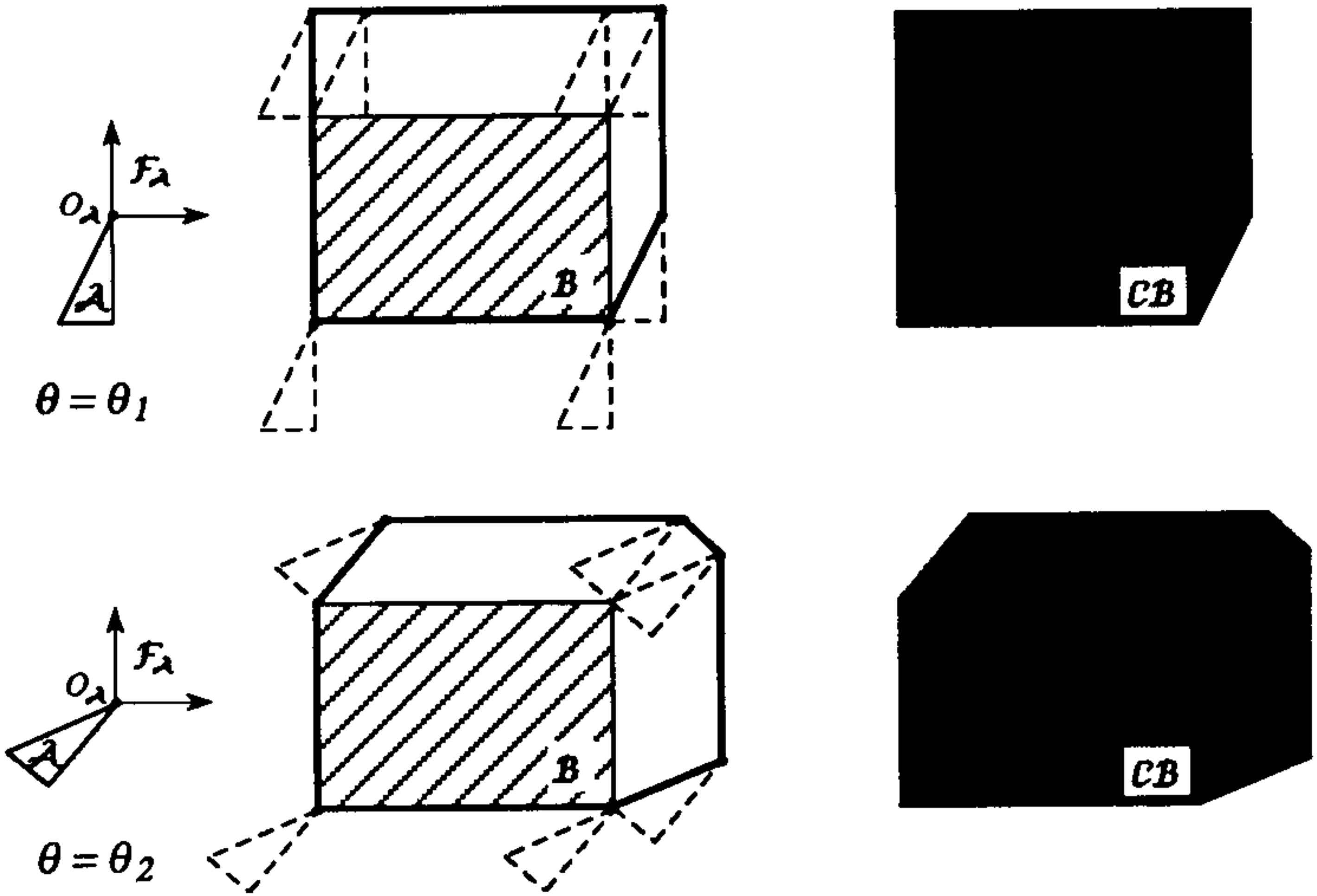




C-Obstacle Construction

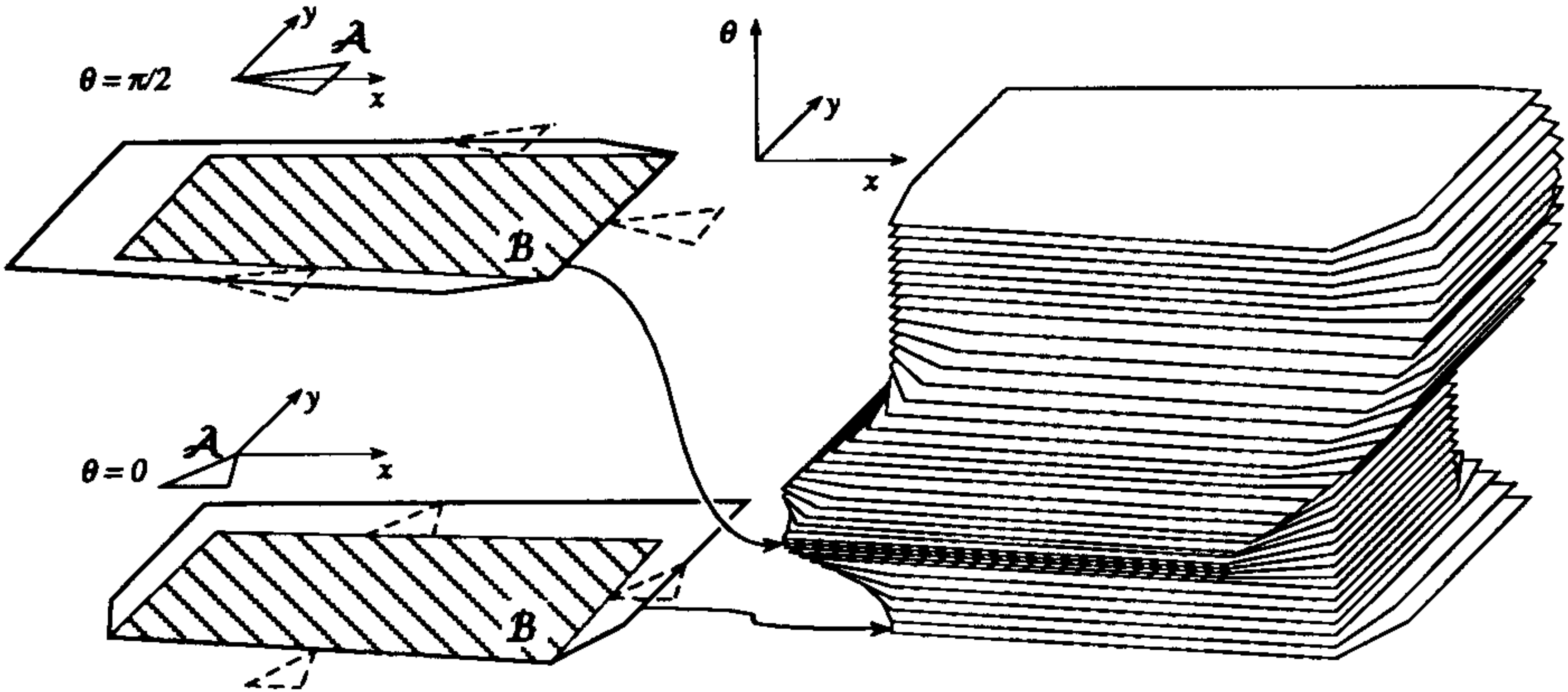


C-Obstacle Construction in 3D



Jean-Claude Latombe

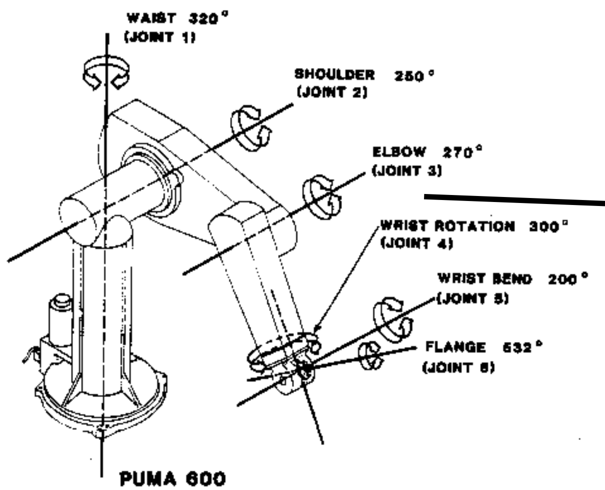
C-Obstacle in 3D



C-Space Construction

- We have seen methods for C-space computation under the following assumptions:
  - planar
  - polygonal obstacles
  - polygonal robot
    - translating
    - translating and rotating
  - holonomic motion

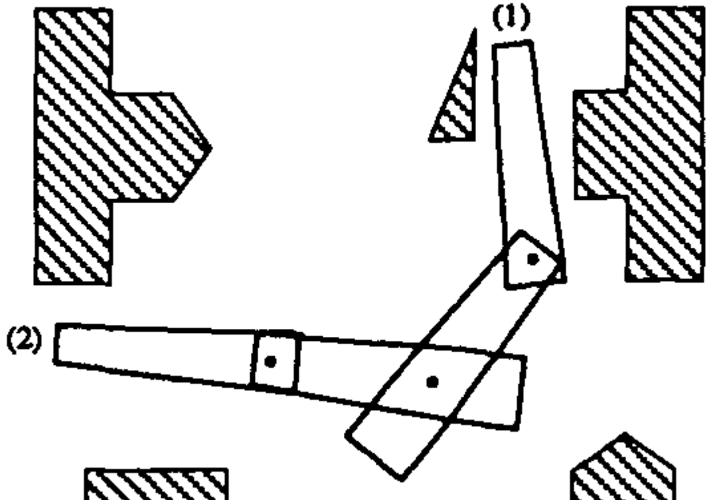
Reminder: Configuration Space

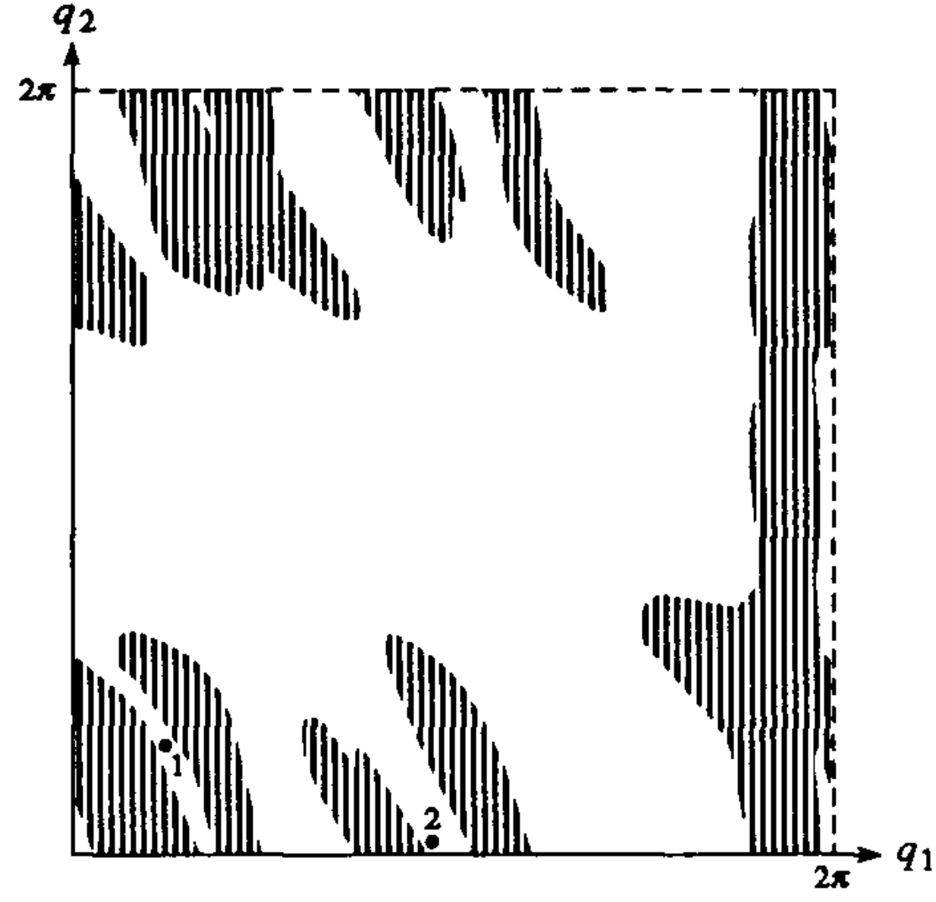


$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}$$

- Configuration: *Minimal* set of parameters *uniquely* describing R
- $q \in R^n$ so that robot is represented as a point
- How do we represent the world in R^n?

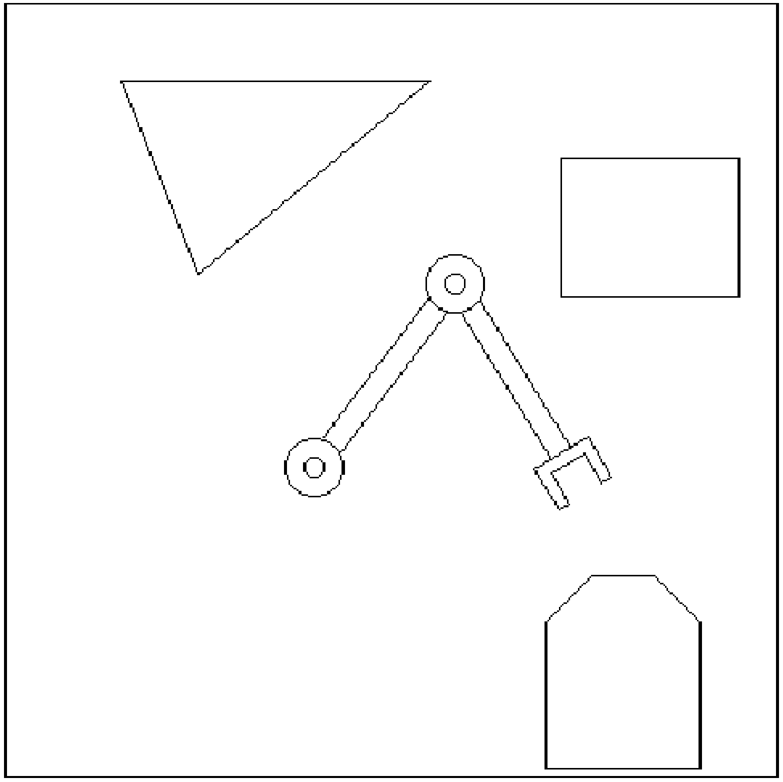
World in C-Space

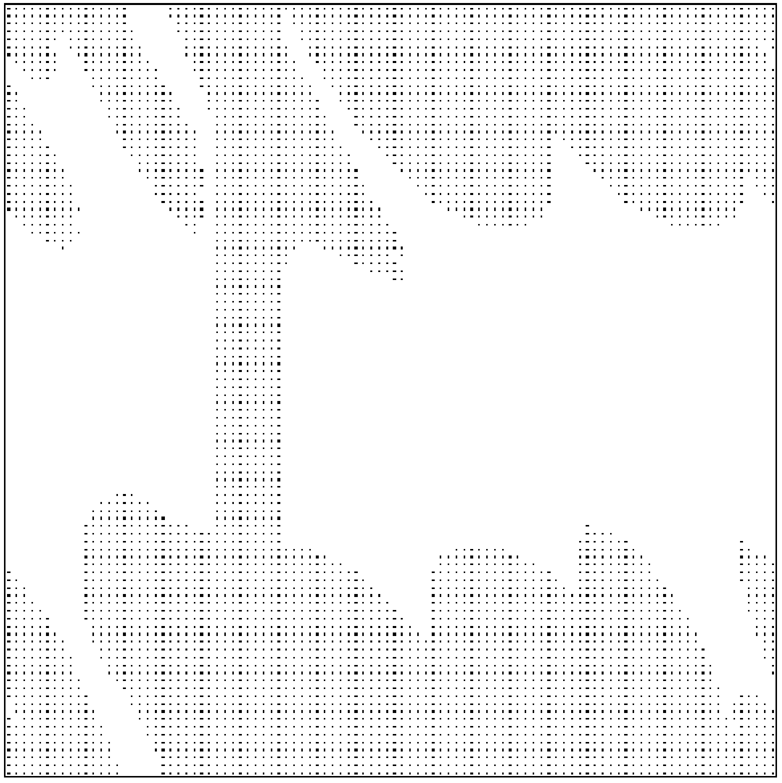




Jean-Claude Latombe

Another World in C-Space





Sean Quinlan

C-Space Obstacles

- Good representation: robot is a point!
- But we are left with the **BIG** problem of Motion Planning:

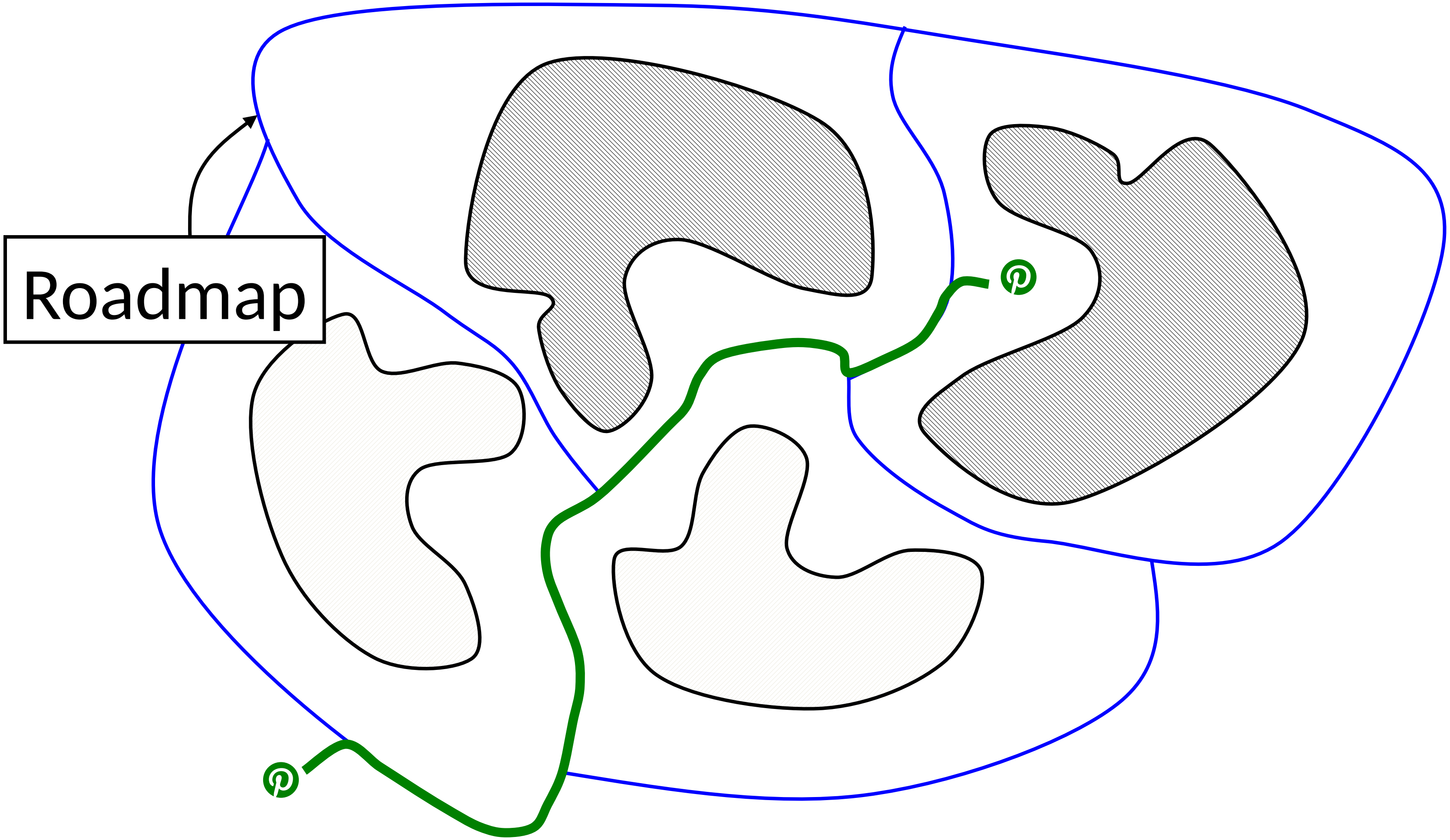
  C-obstacle computation

- How to translate the world into C-space?

C-Space Construction for *n* dof

- Don't underestimate the problem:
  - 6 rotational degrees of freedom (e.g. Puma)
  - 360 degrees of orientation per joint
  - 2 x 10^15 configurations
  - 1 million checks per second
  - 69 years of computation
- Naïve grid methods are **computationally intractable!**

Roadmap Methods



How to obtain the roadmap?

Jean-Claude Latombe

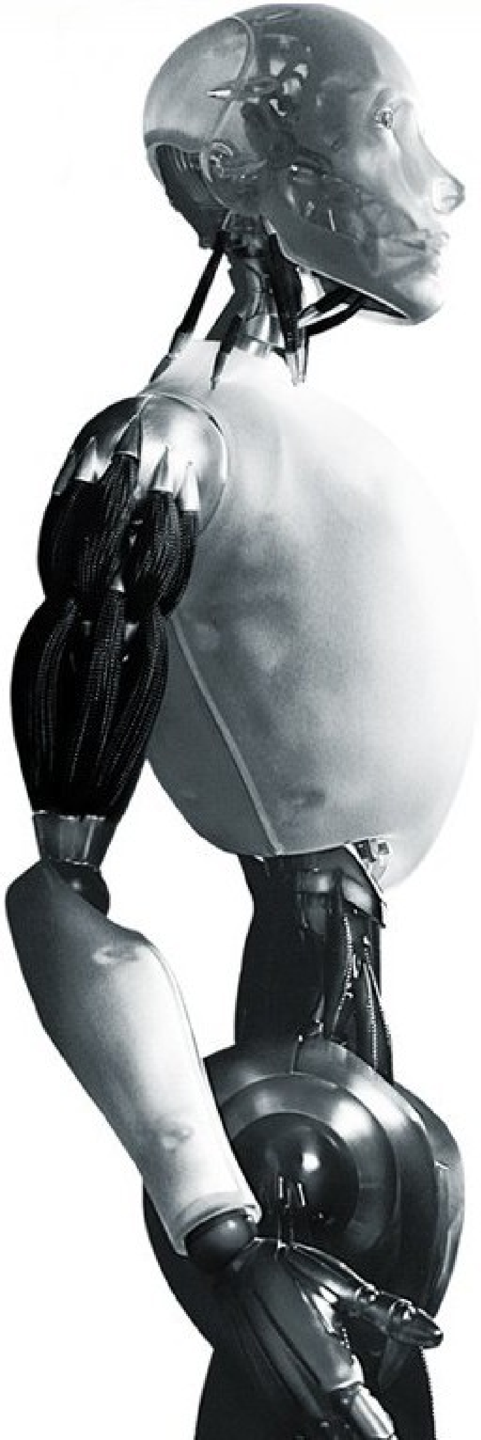
Motion versus Path

- “Path” refers to continuous sequence of configurations
- “Motion” or “trajectory” includes time parameterization
- Most people just refer to Motion Planning
- Here: Path Planning = Motion Planning
- We ignore time parameterization!

Free Path / Connected Component

A **free path** between two configuraions q_{init} and q_{goal} is a continuous map $\mathcal{P} : [0,1] \rightarrow C_{\text{free}}$ with $\mathcal{P}(0) = \mathrm{q}_{\text{init}}$ and $\mathcal{P}(1) = \mathrm{q}_{\text{goal}}$.

Two configurations belong to the same **connected component** of C_{free} if and only if they are connected by a free path.



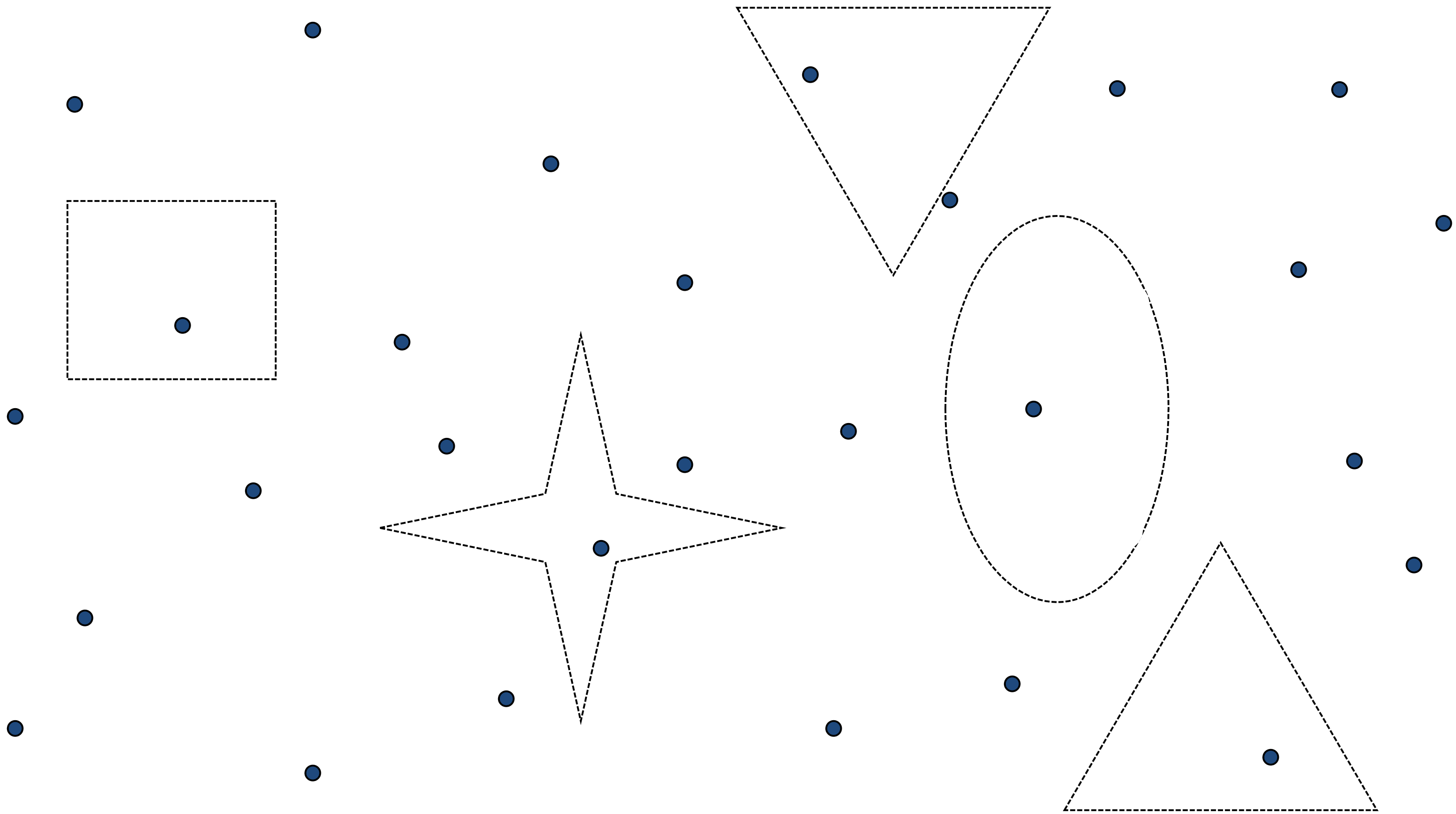
Robotics

Sampling-based Motion Planning

TU Berlin

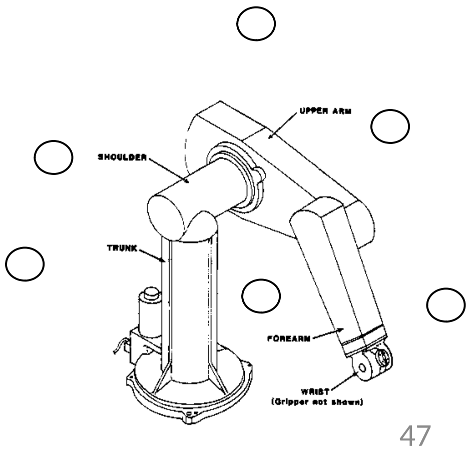
Oliver Brock

Sampling Configuration Space



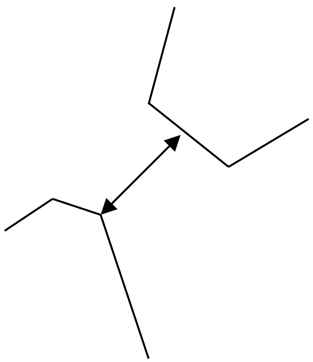
Basic Primitive: Collision Detection

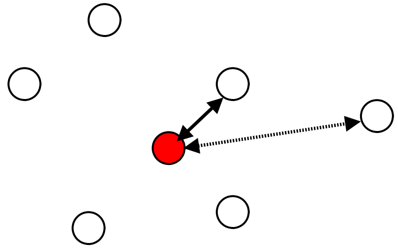
- Computational complexity collision detection
  - n objects have $O(n^2)$ interactions
  - Robot with l links and n obstacles has $O(l * n)$
  - Each object has many features – 1000s!!!
  - In practice a **costly** operation



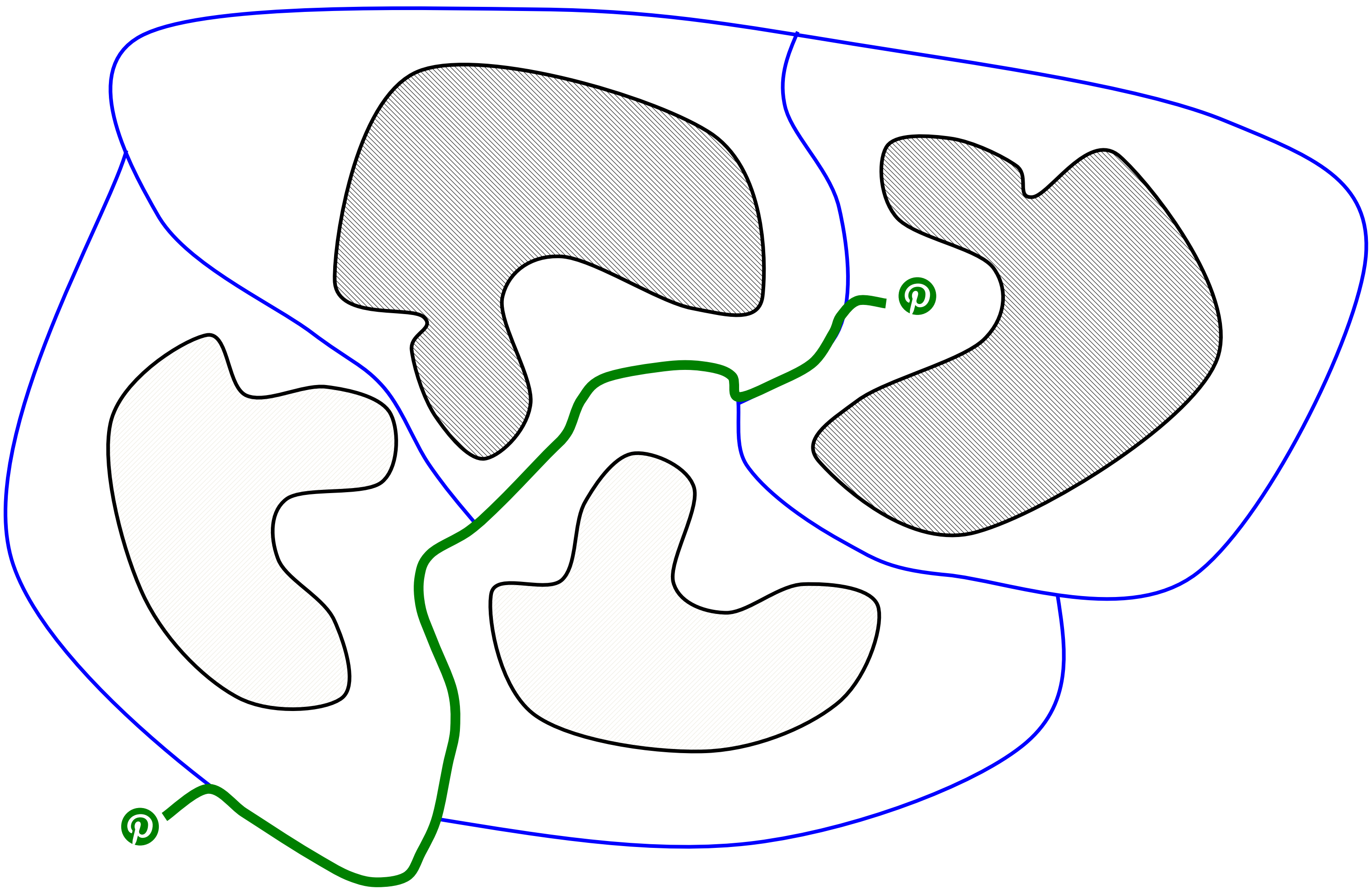
Tricks for Distance Computation

- Exploit spatial coherency
  - Group primitives hierarchically
  - Exploit adjacency (on single object)
- Exploit temporal coherency
  - Exploit former relation between multiple objects
- Heuristics are good
- Problem remains computationally complex





What is the perfect roadmap?

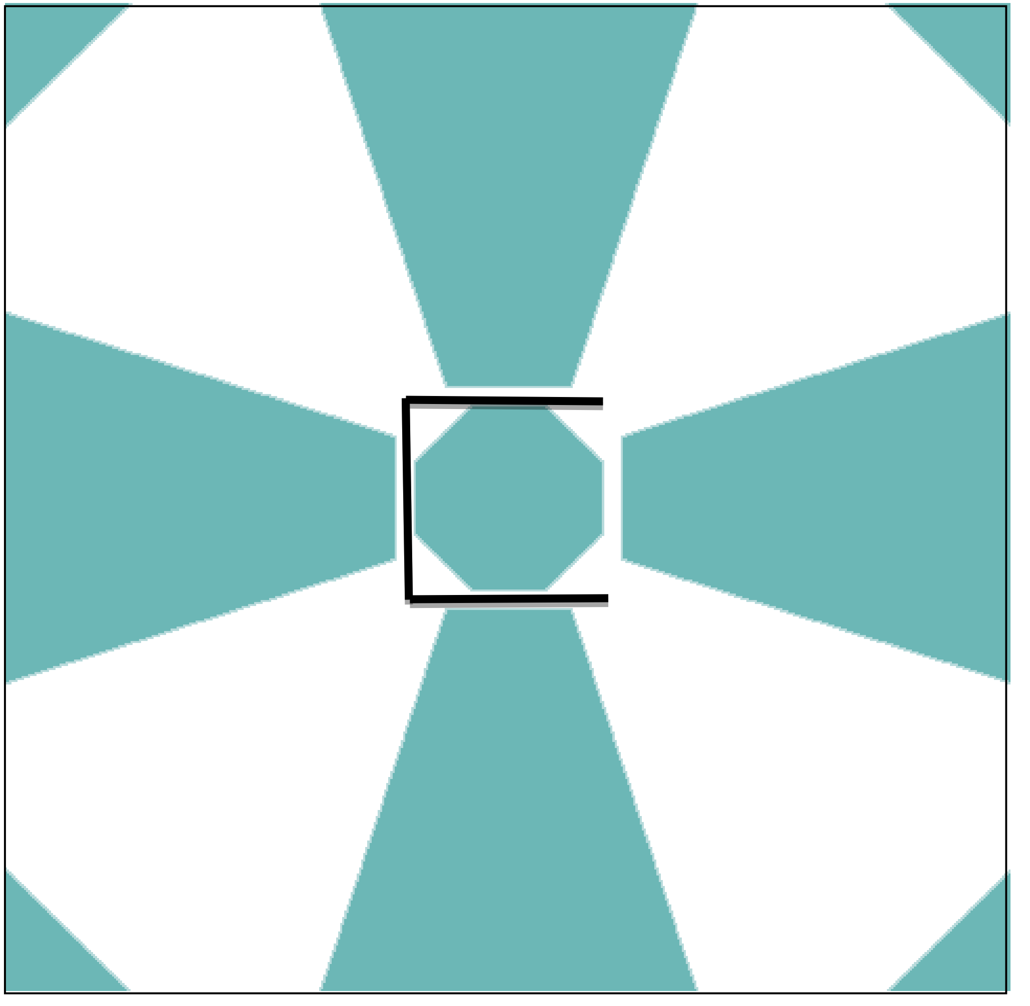


An Ideal Roadmap

- Any point in C-space should be connectable to the roadmap
- If there is a path between two points in C-space the roadmap should contain a path between them after they were connected to the roadmap

- How can such a roadmap be obtained through sampling?

Perfect Roadmap

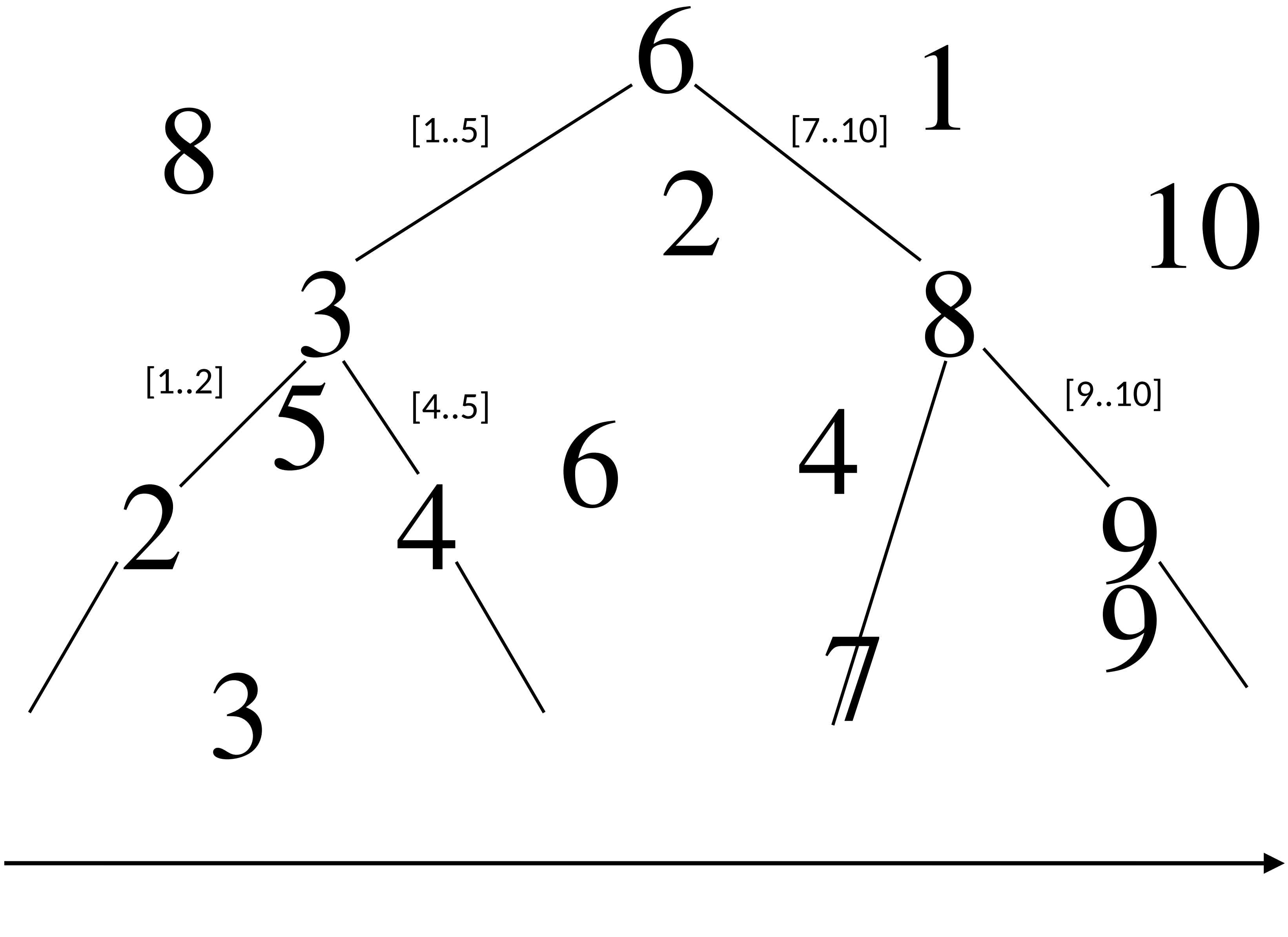


Exploration versus Exploitation

Exploration seeks **understanding of the state space**, irrespective of a particular task.
In motion planning, the process **exploration** seeks to understand the connectivity of the configuration space, irrespective of solving a particular motion planning problem.

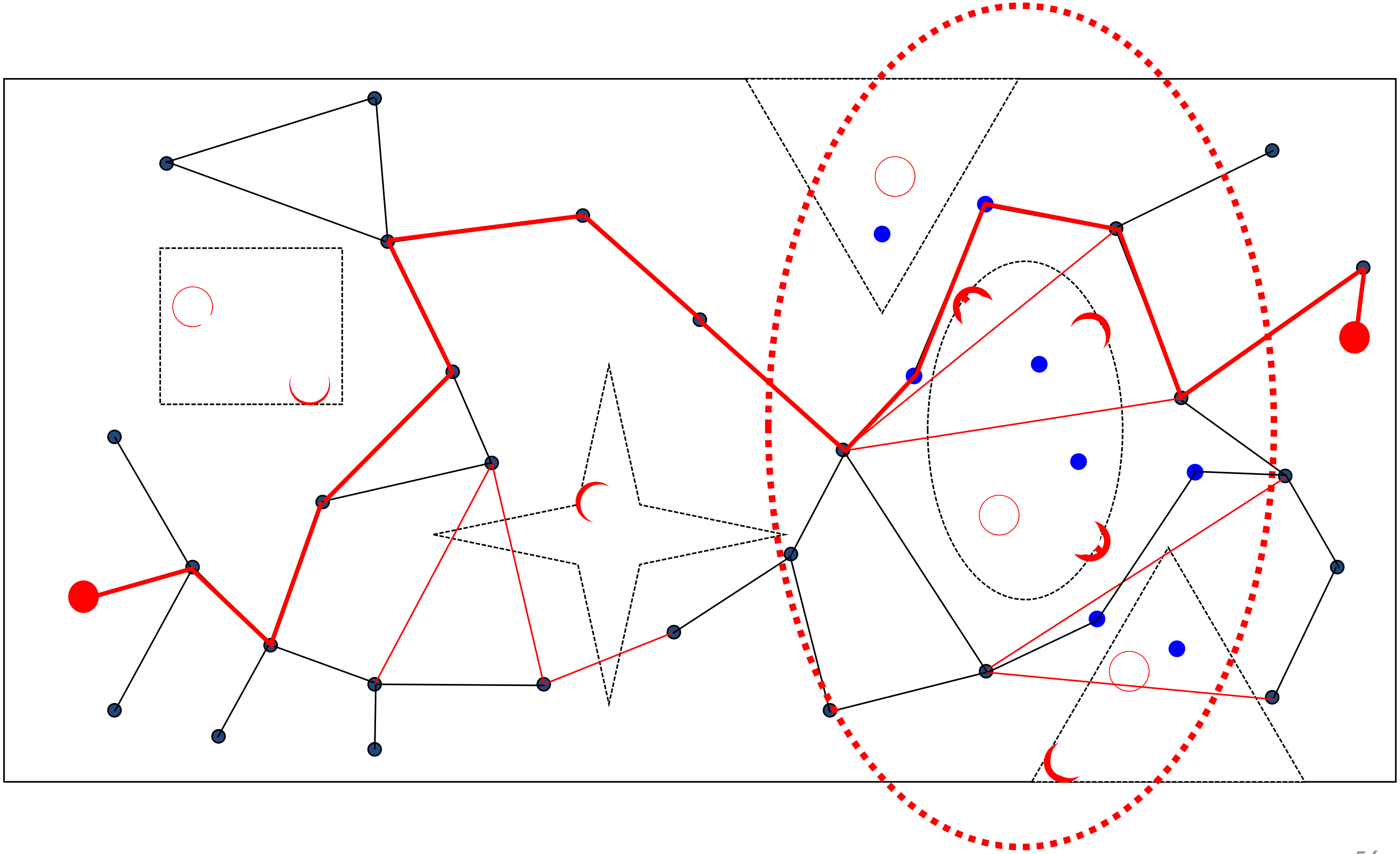
Guided ***exploration*** seeks efficient **understanding of the state space**, irrespective a particular task, by **leveraging available information**.

Exploitation strives to **accomplish a particular task** as **efficiently as possible** by **leveraging available information**.
In motion planning, **exploitation** seeks a valid path for a **particular task**, based on available information.



6
1
8
[1..5]
[7..10]
2
10
3
8
[1..2]
[4..5]
[9..10]
5
6
4
2
4
9
9
7
3

Probabilistic Roadmap (PRM) Planner



Probabilistic Roadmap Planner

- Construction
  - Generate random configurations
  - Eliminate if they are in collision
  - Use local planner to connect configurations
- Expansion
  - Identify connected components
  - Resample gaps
  - Try to connect components
- Query
  - Connect initial and final configuration to roadmap
  - Perform graph search

Learning

Learning Phase

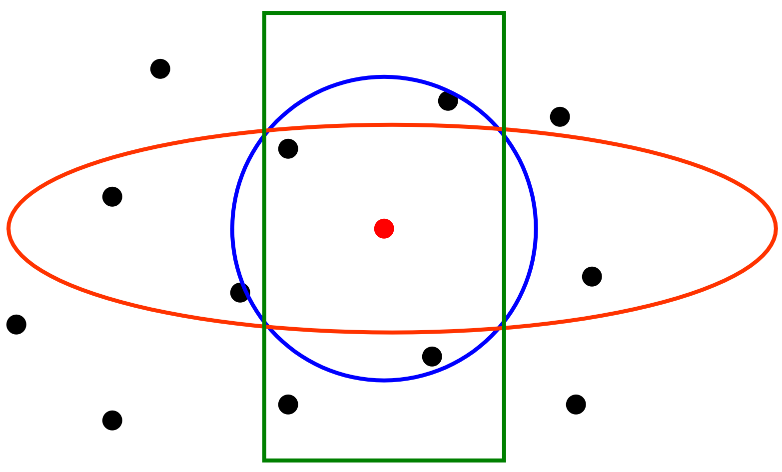
- Construction
  - $R = (V,E)$
  - repeat n times:
    - generate random configuration
    - add to V if collision free
    - attempt to connect to *neighbors* using *local planner*, unless in same connected component of R
- Expansion
  - repeat k times:
    - select *difficult* node
    - attempt to connect to neighbors using *another local planner*

Query

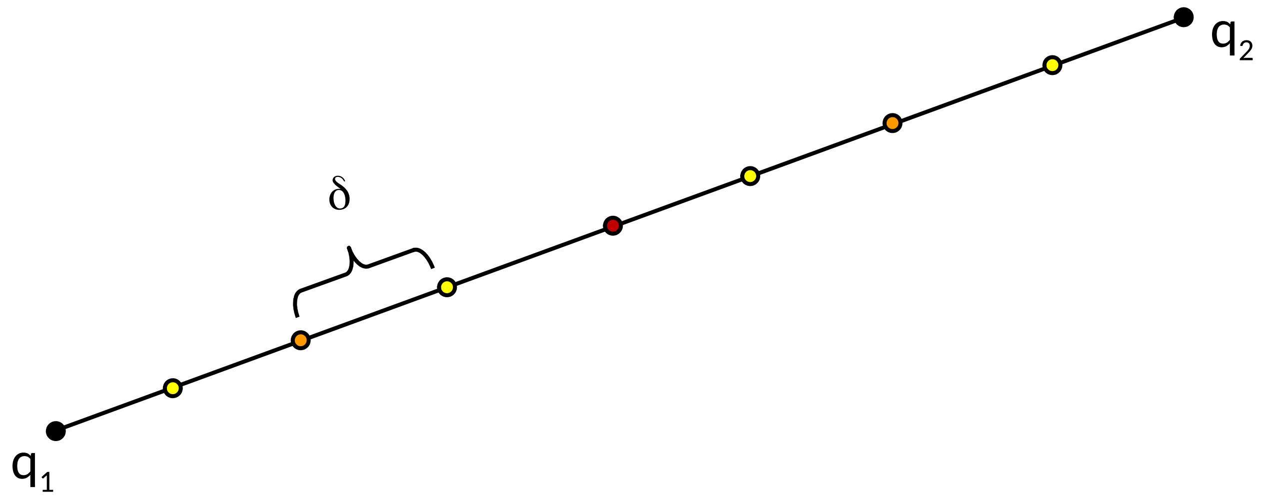
- Connect start and goal configuration to roadmap using local planner
- Perform graph search on roadmap
- Computational cost of querying neclegible compared to construction of roadmap

Neighbors

- Use distance metric to determine neighbor
- Euclidian distance oftentimes used
- Others possible:
  - maximum Euclidian distance
  - maximum joint difference

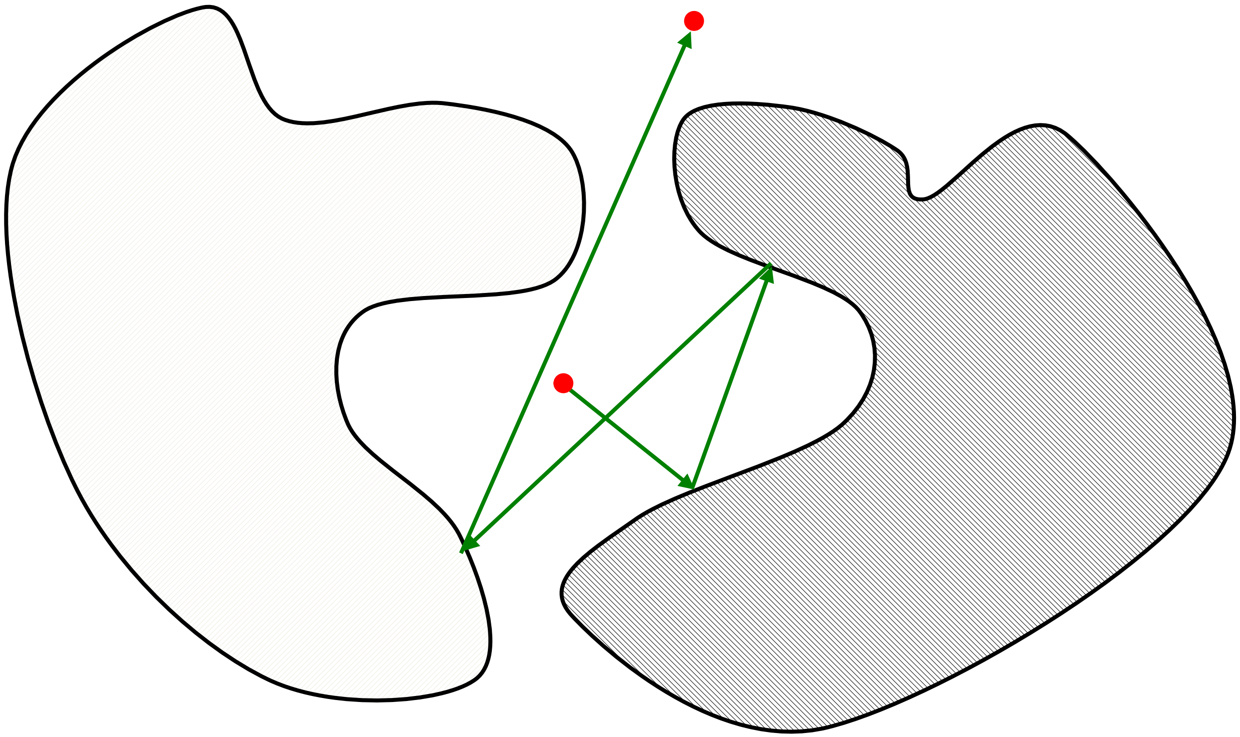


Local Planner



tests up to a specified resolution δ!

Another Local Planner

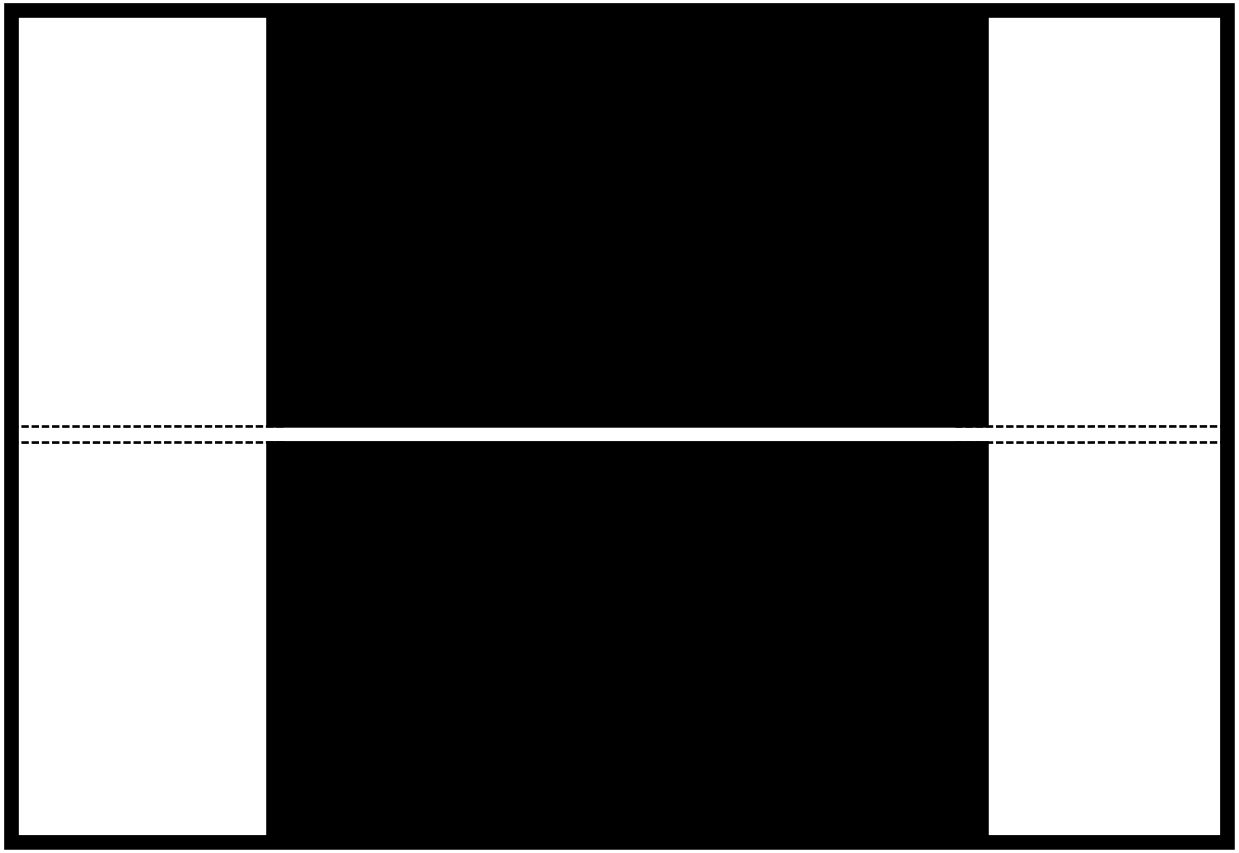


perform random walk of predetermined length;
choose new direction randomly after hitting obstacle;
attempt to connect to roadmap after random walk

PRM Limits Local Planners

- Consider car-like robot
- Connecting configurations might be difficult
- Goal: provide probabilistic method for kinematic and dynamic constraints
  - Car-like
  - Satellite
  - Plane
- Idea: Let local planner choose configurations

Narrow Passage Problem



Extensions to PRM

- The narrow passage problem
- Examples of advanced sampling methods for PRM:
  - Gaussian sampling
  - Bridge sampling
  - Free space dilatation
  - Obstacle-based PRM (OBPRM)
  - Medial Axis PRM

Summary: PRM

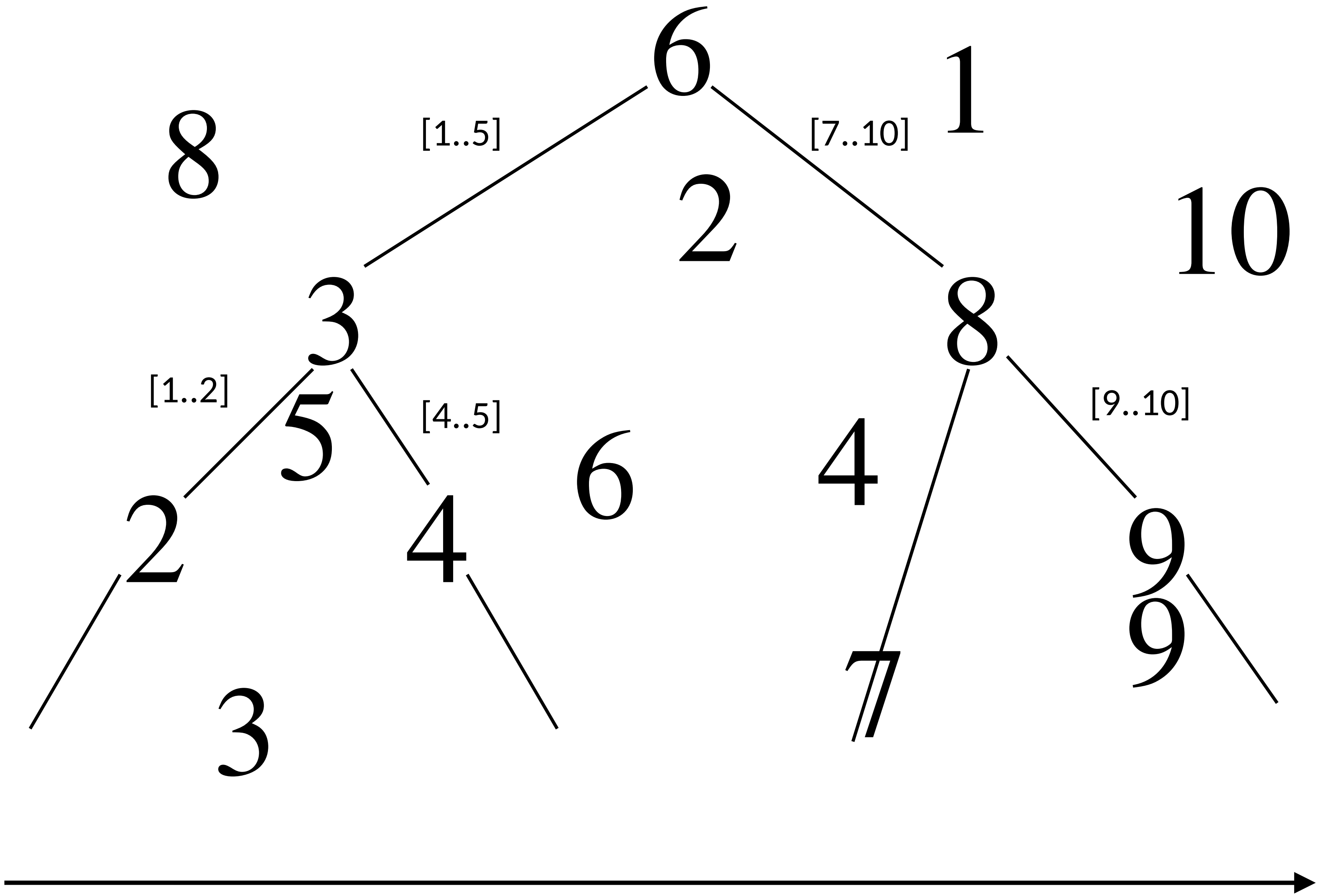
- Algorithmically very simple
- Surprisingly efficient even in high-dimensional C-spaces
- Capable of addressing a wide variety of motion planning problems
- One of the hottest areas of research
- Allows probabilistic performance guanratees
- **BUT: narrow passage problem!**

Exploration versus Exploitation

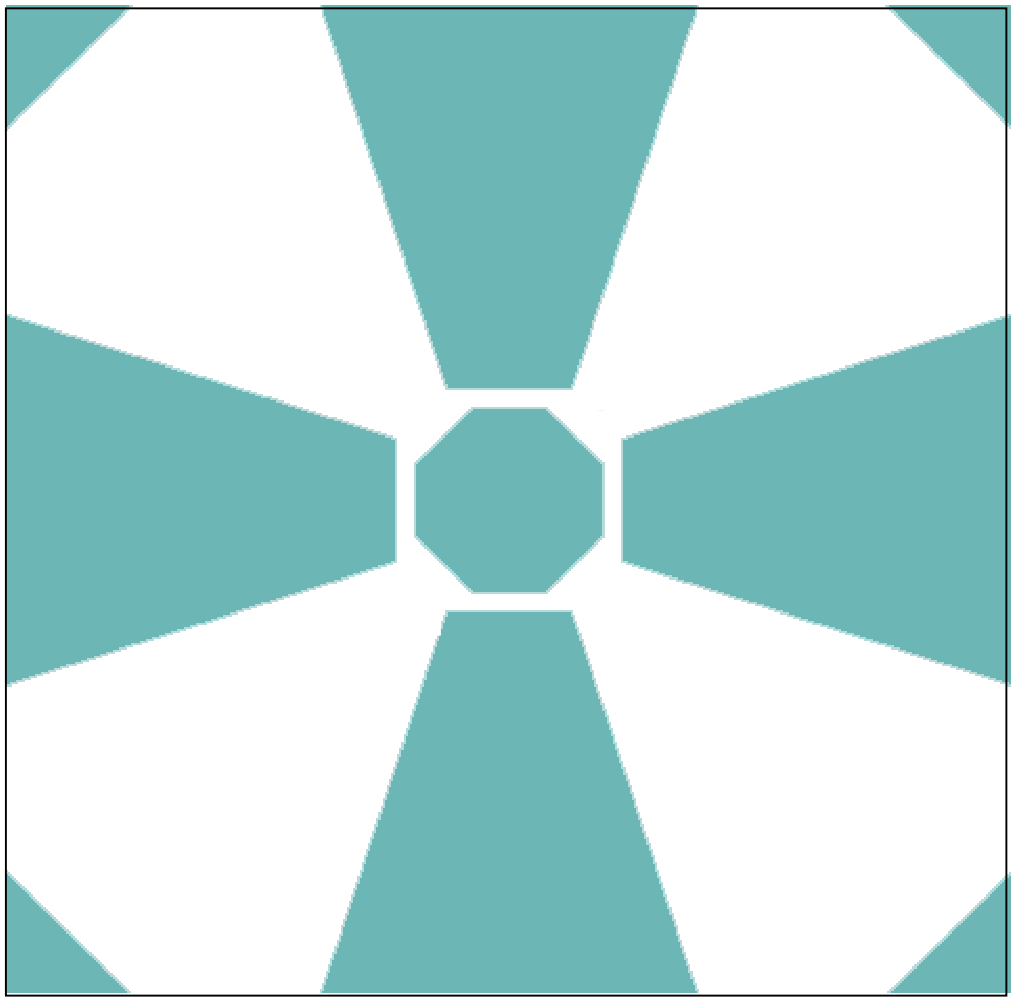
Exploration seeks **understanding of the state space**, irrespective of a particular task. In motion planning, the process **exploration** seeks to understand the connectivity of the configuration space, irrespective of solving a particular motion planning problem.

Guided exploration seeks **efficient** **understanding of the state space**, irrespective a particular task, by **leveraging available information**.

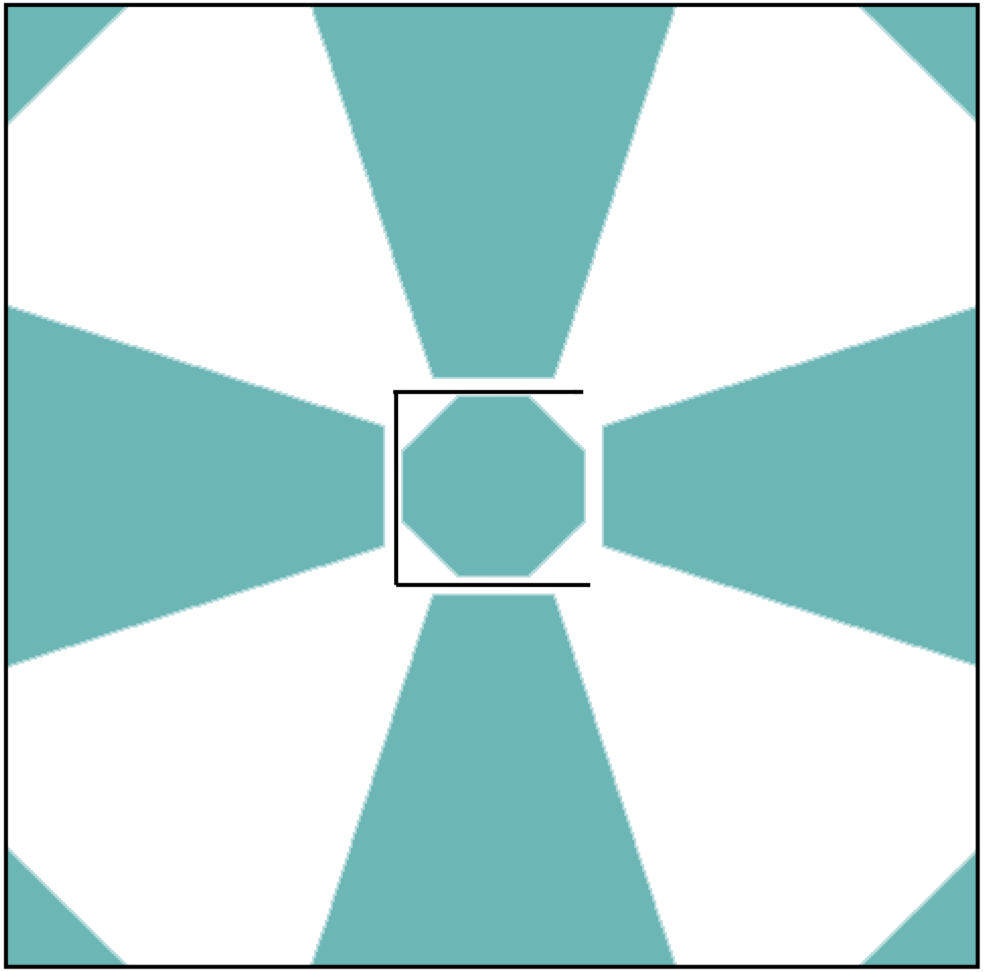
Exploitation strives to **accomplish a particular task** as **efficiently as possible** by **leveraging available information**. In motion planning, **exploitation** seeks a valid path for a **particular task**, based on available information.


6
1
8
[1..5]
[7..10]
2
10
3
8
[1..2]
5
[4..5]
[9..10]
6
4
2
4
9
9
7
3

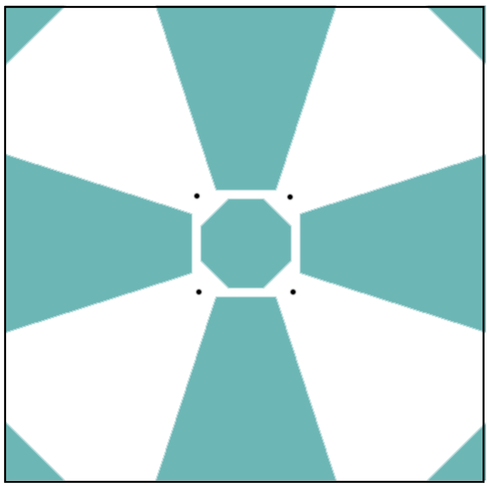
Perfect Sampling



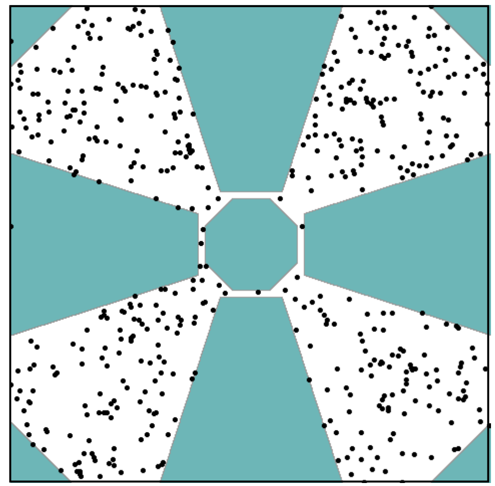
Perfect Roadmap



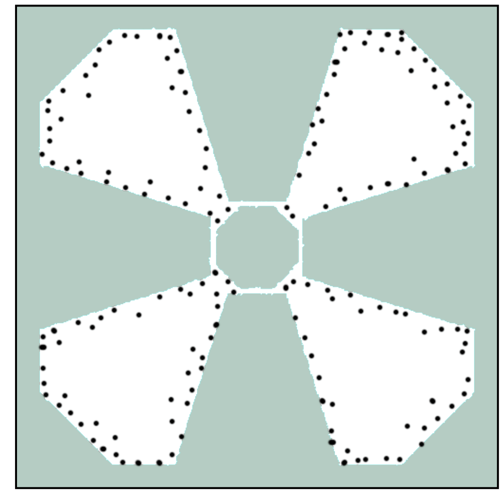
Different Sampling Strategies



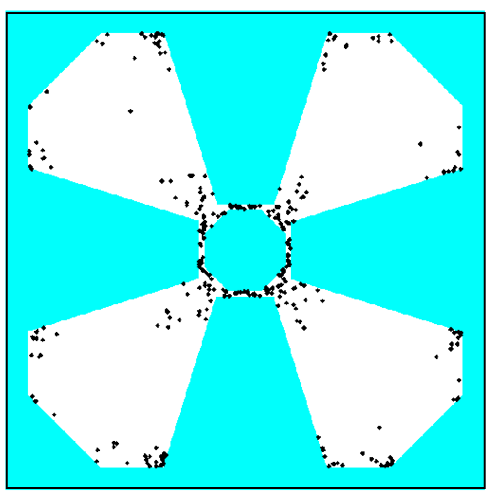
ideal



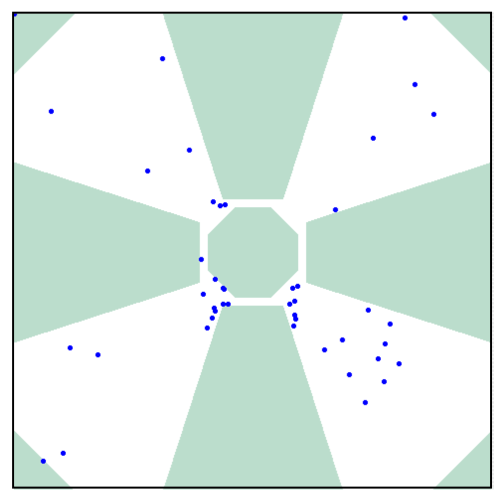
uniform



Gaussian



Bridge



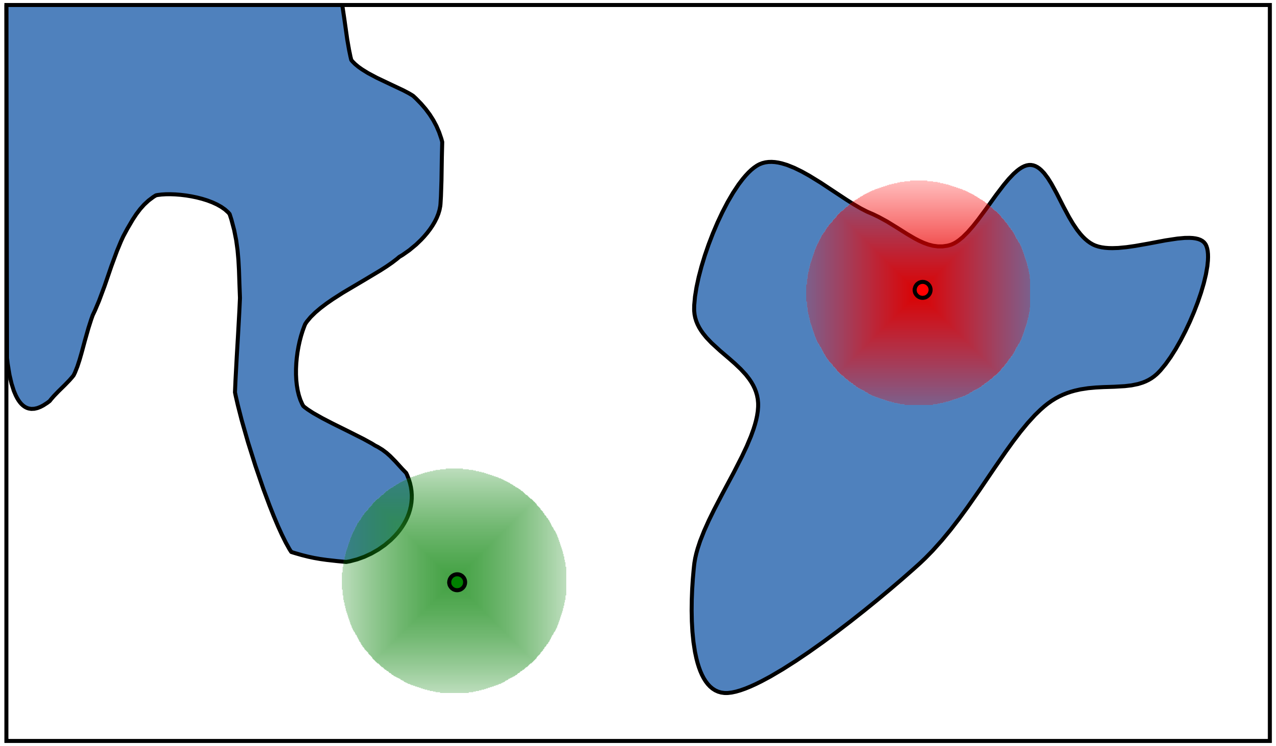
Utility

Key to Good Sampling: Exploiting Structure

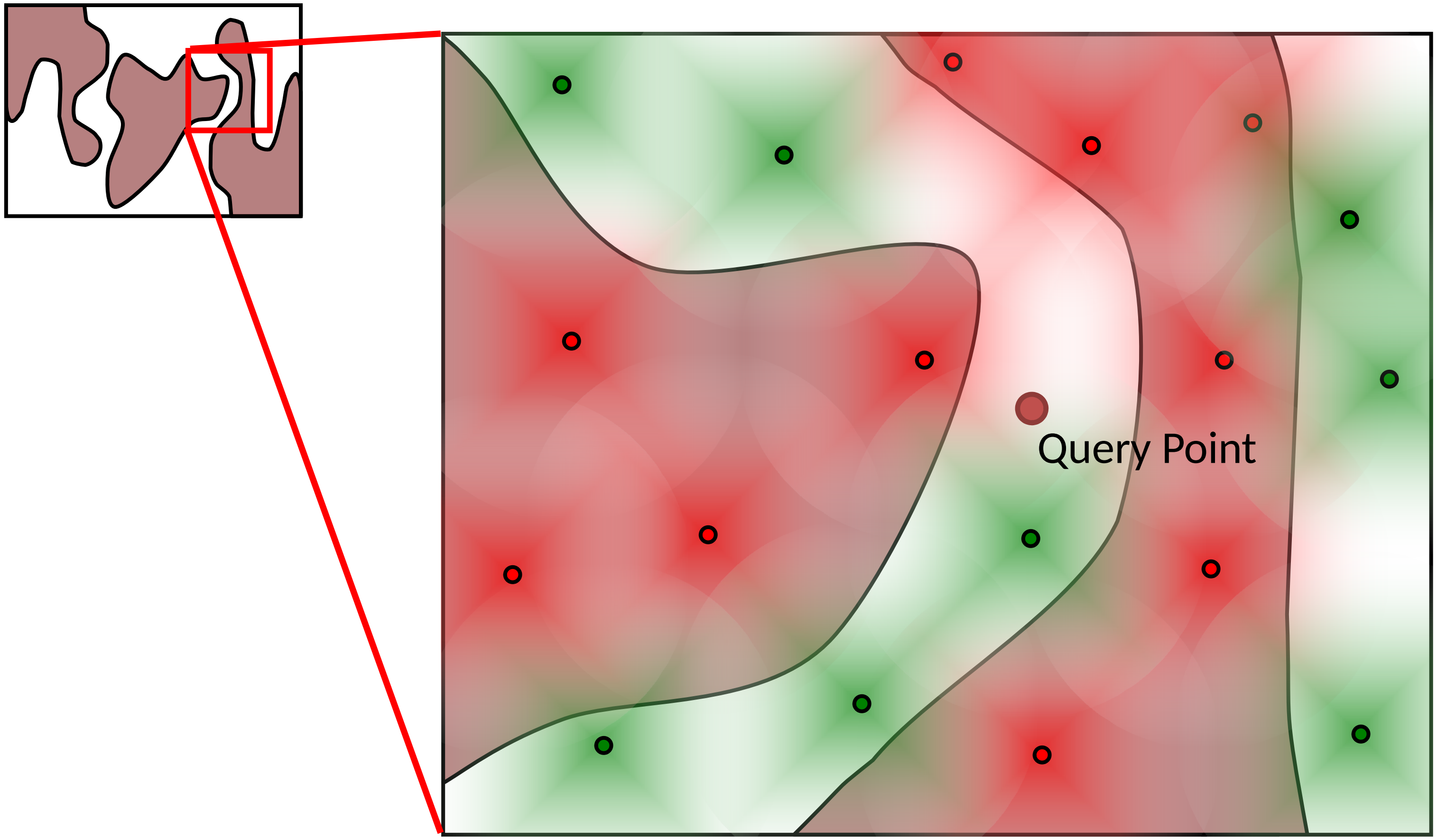
- *Identify* underlying structure
- *Represent* information about structure
- *Exploit* information
- *Structure* can come from
  - sampling
  - problem description

Learning Structure through Sampling

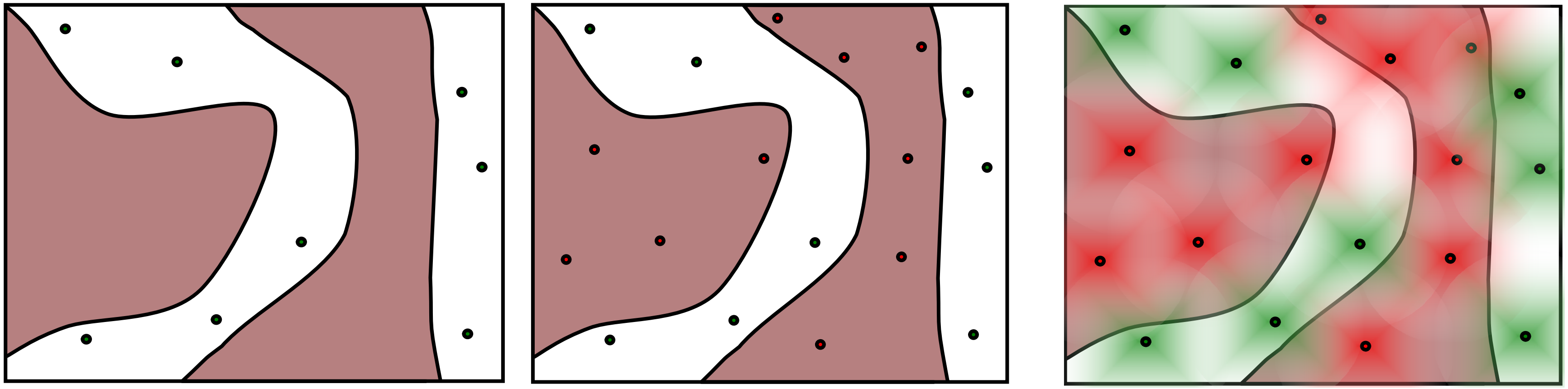
A non-parametric model of C-space



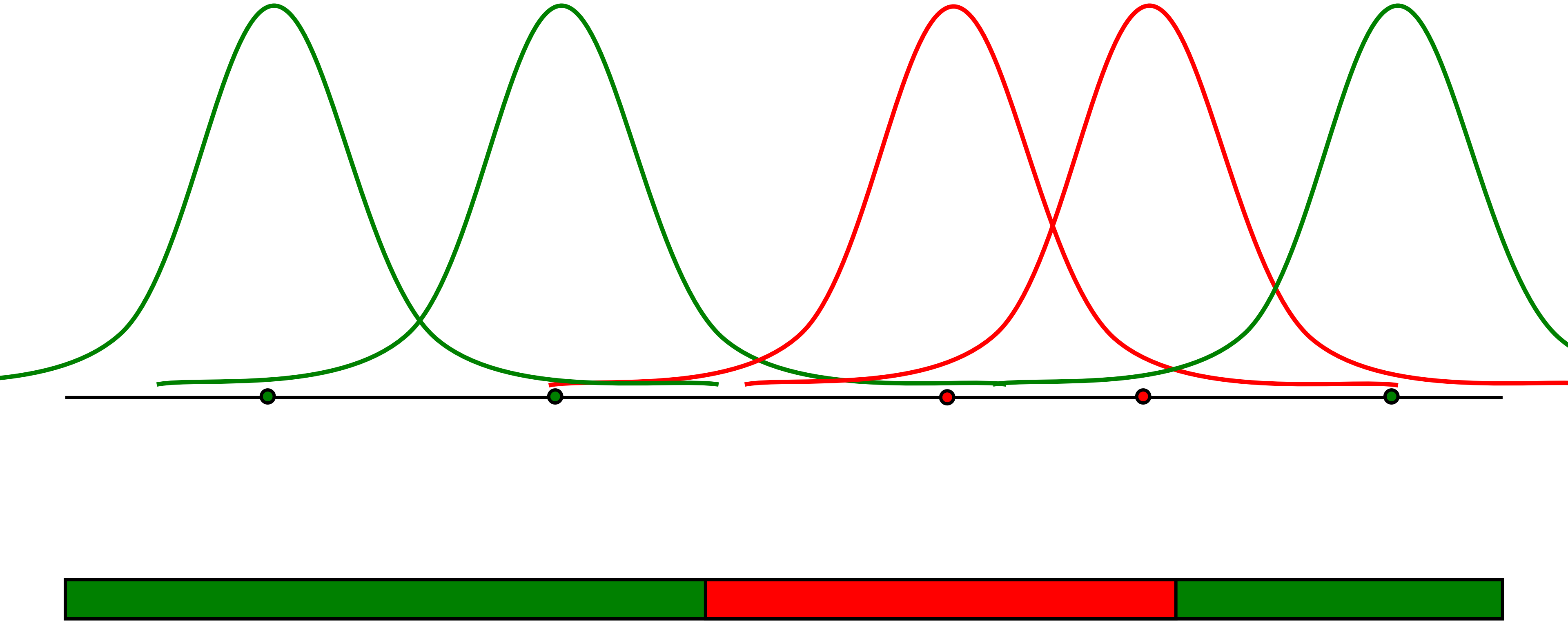
Building a Model of Configuration Space



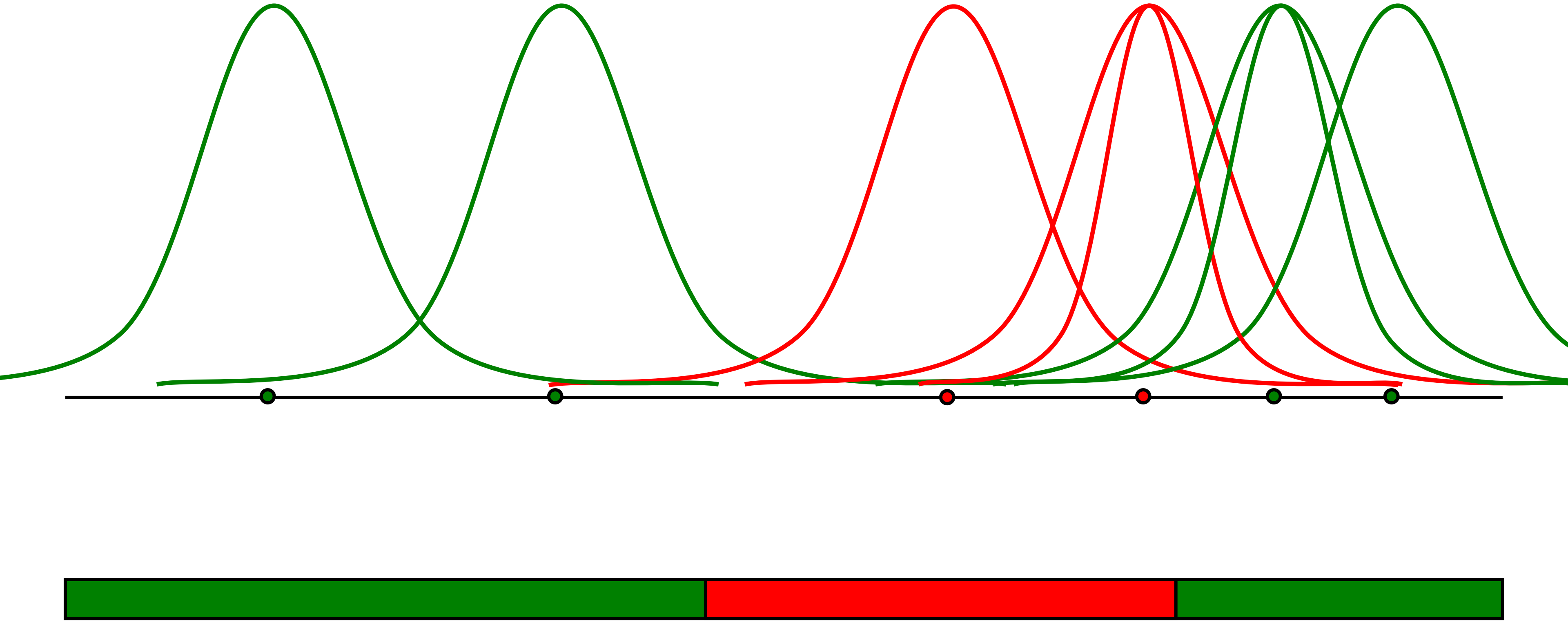
Comparison of Information Content



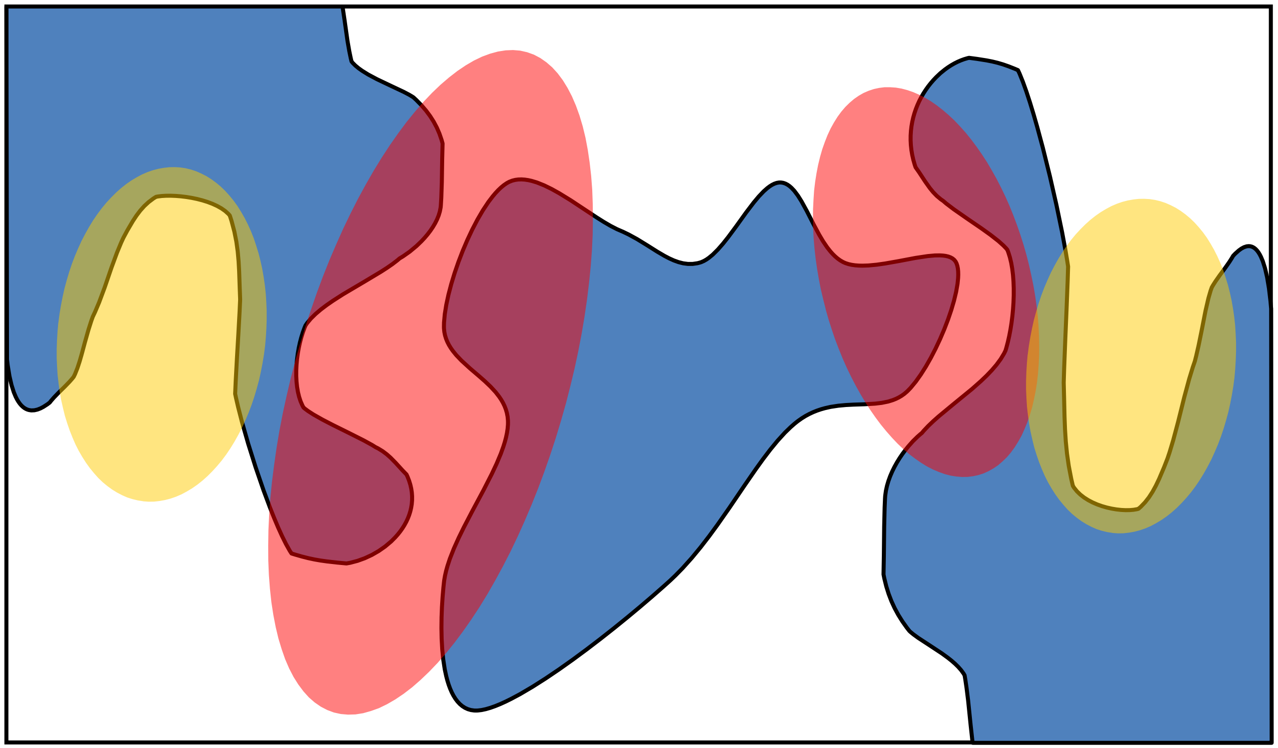
1D Non-Parametric Model



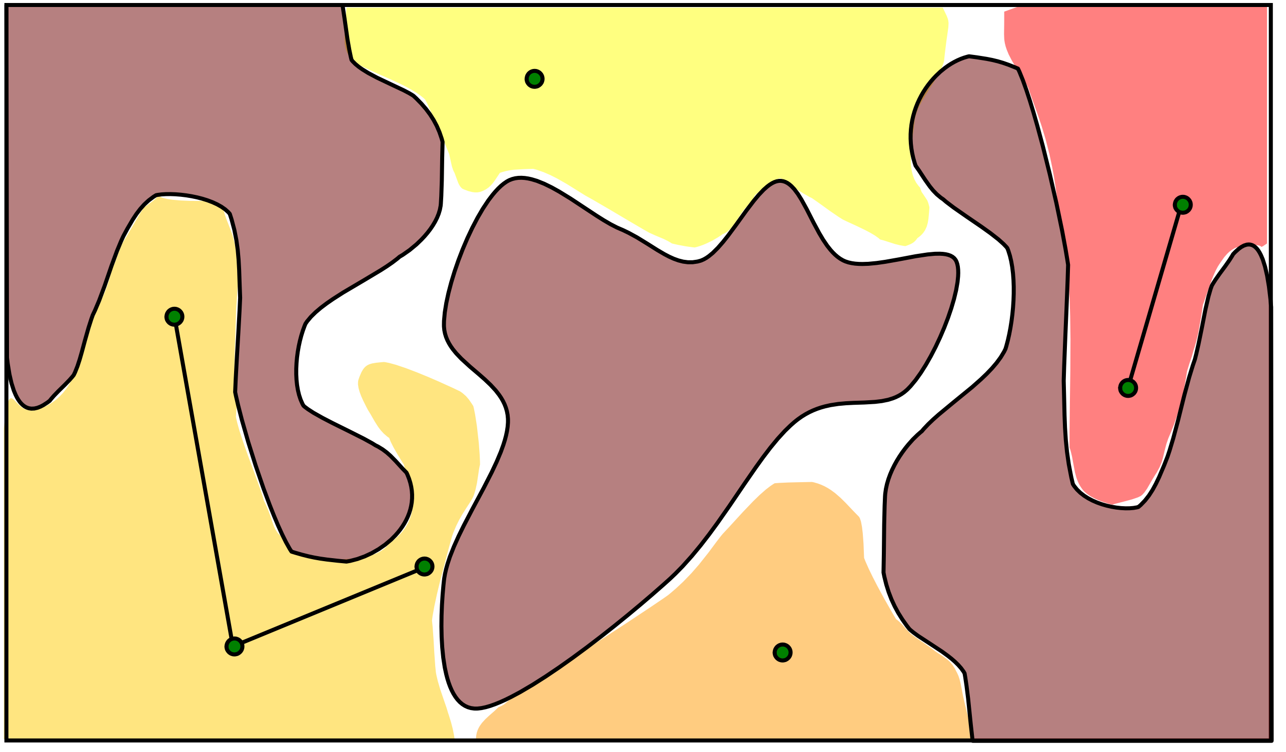
Sampling based on High Variance



High Model Variance



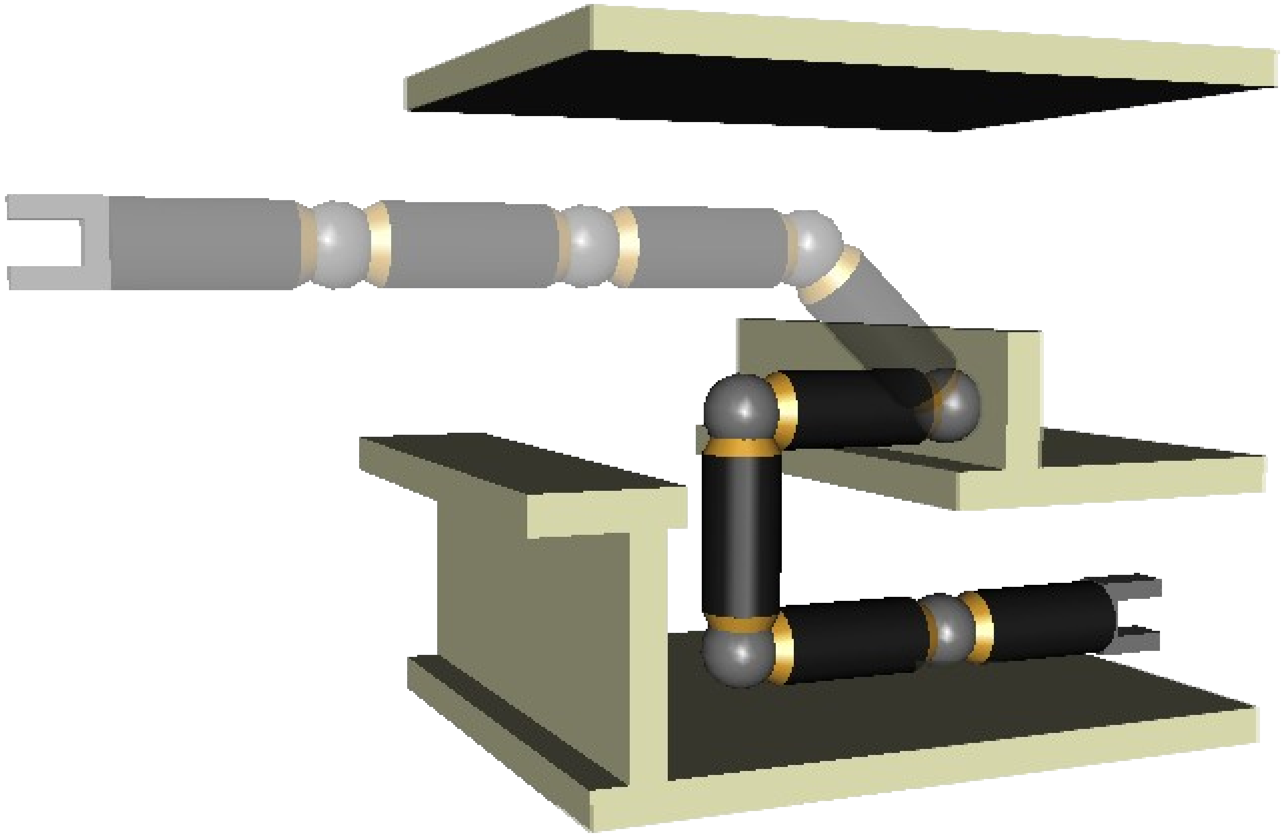
Estimating Utility

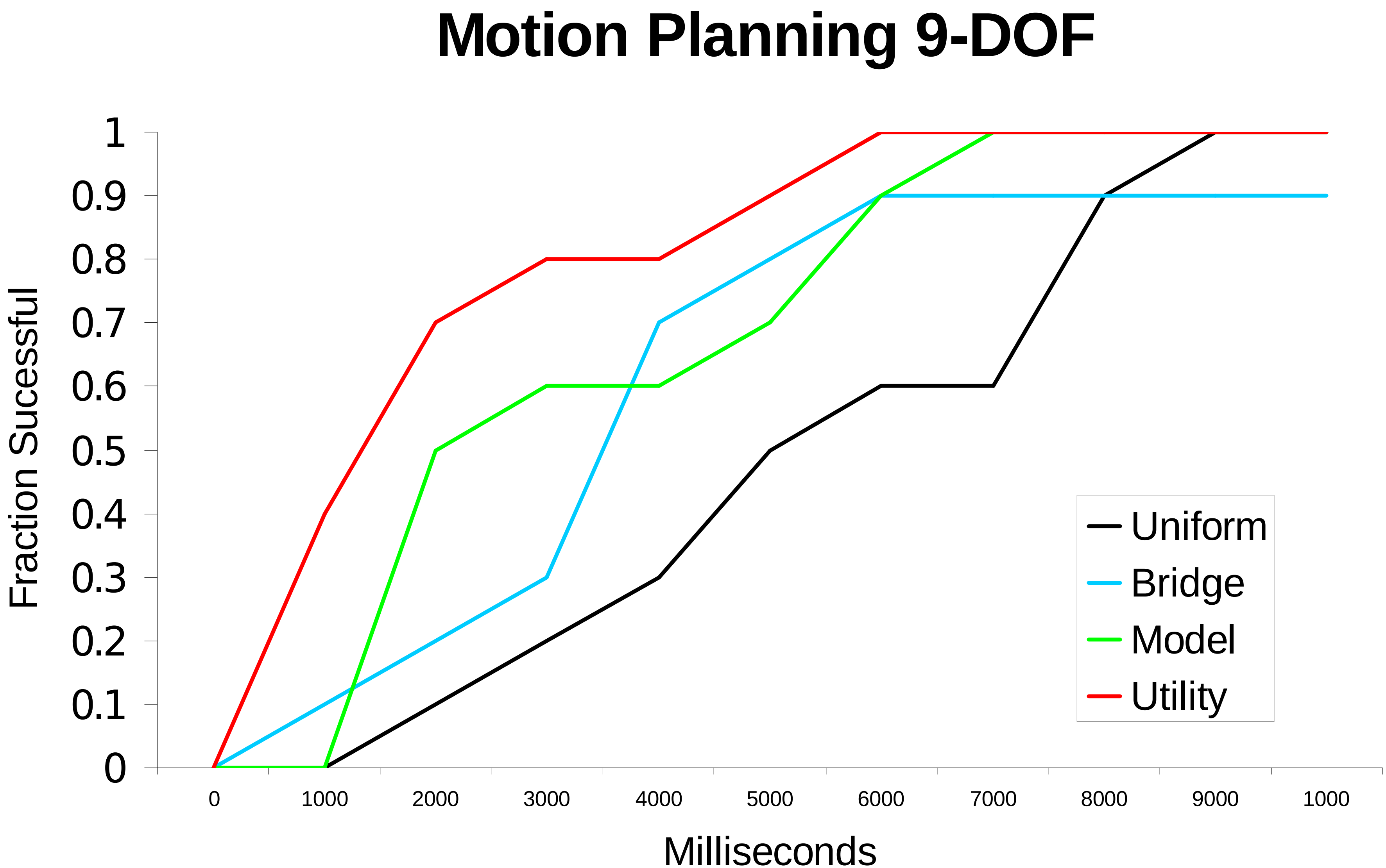


Expected Utility

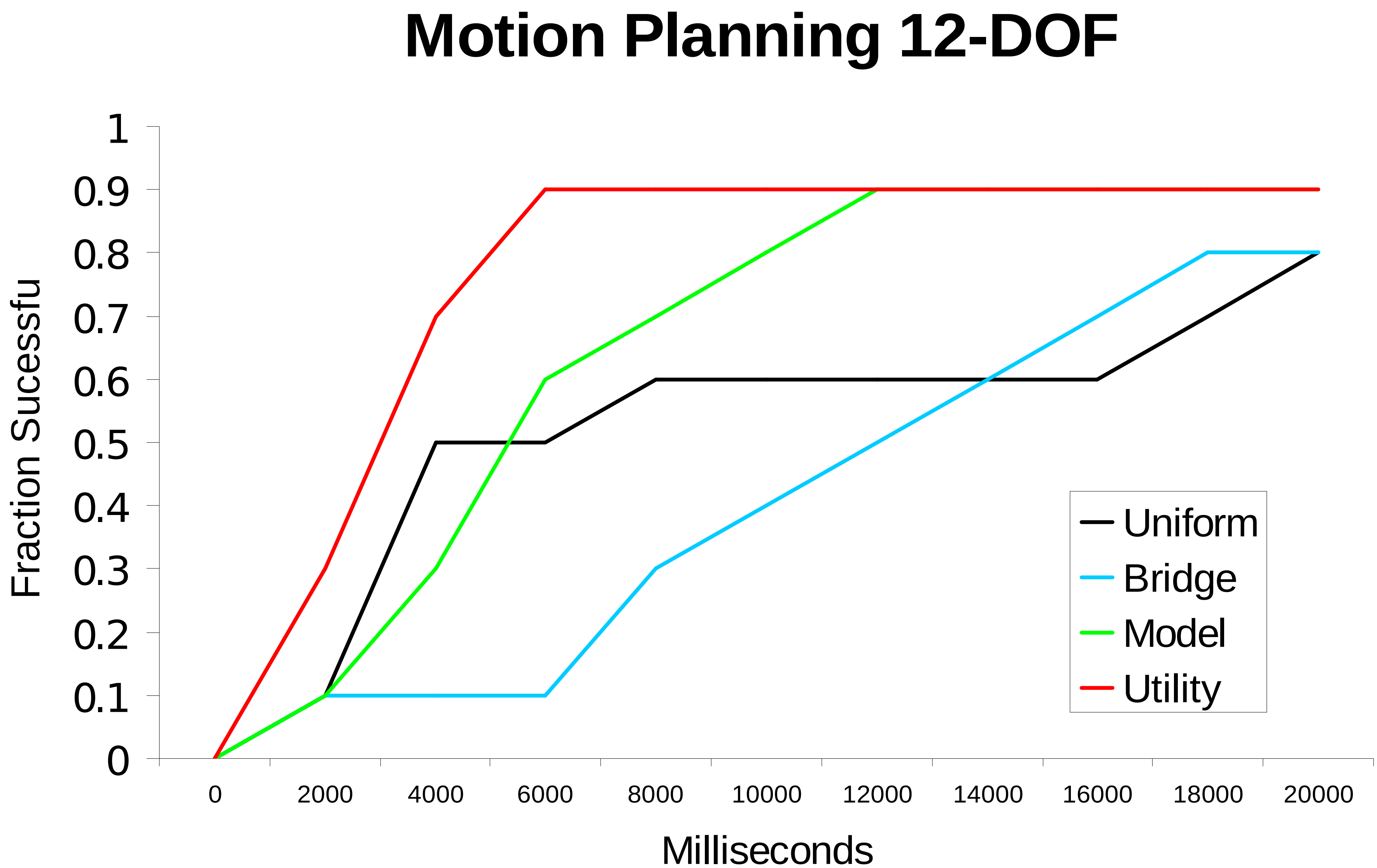
$$E[U(q \mid M)] = \sum_{i \in \{0,1\}} \overbrace{P(q = i \mid M)}^{\text{Instance}} \cdot \underbrace{U(q = i \mid M)}_{\text{Domain}}$$

Experimental Environment



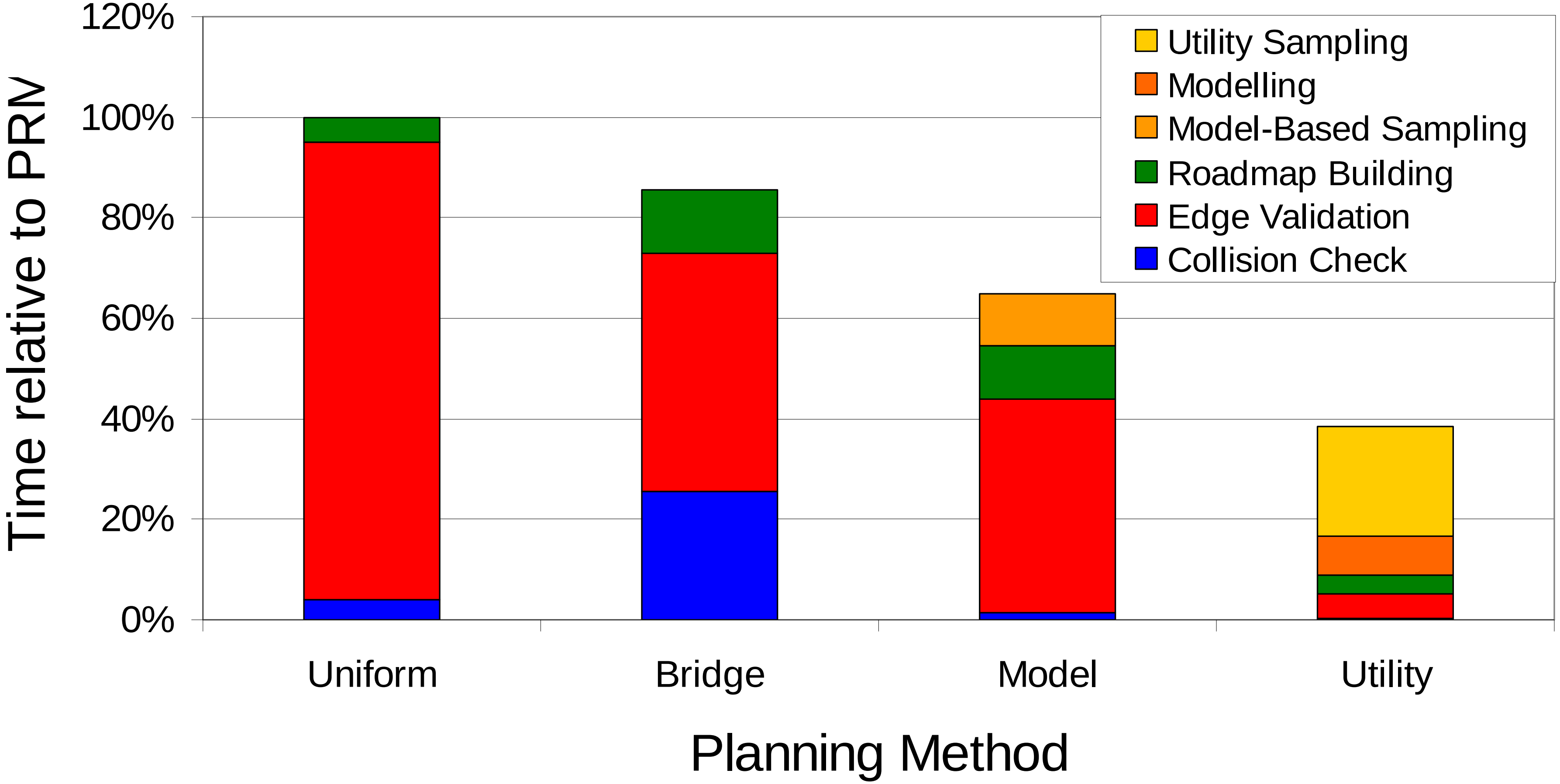


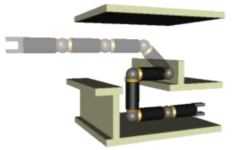
Motion Planning 9-DOF
Fraction Sucessful
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
1000
Milliseconds
Uniform
Bridge
Model
Utility



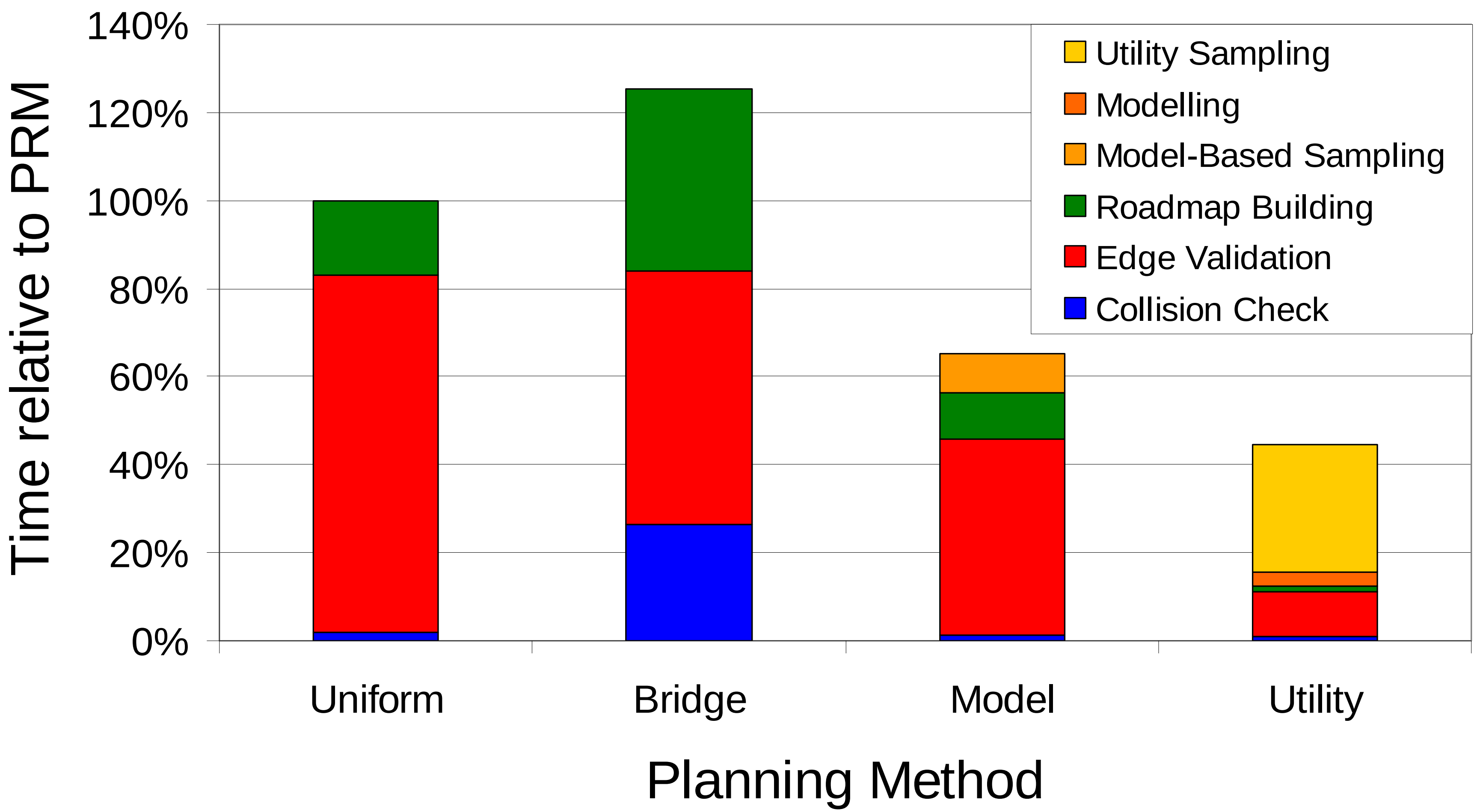
Motion Planning 12-DOF
Fraction Sucessfu
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
Milliseconds
Uniform
Bridge
Model
Utility

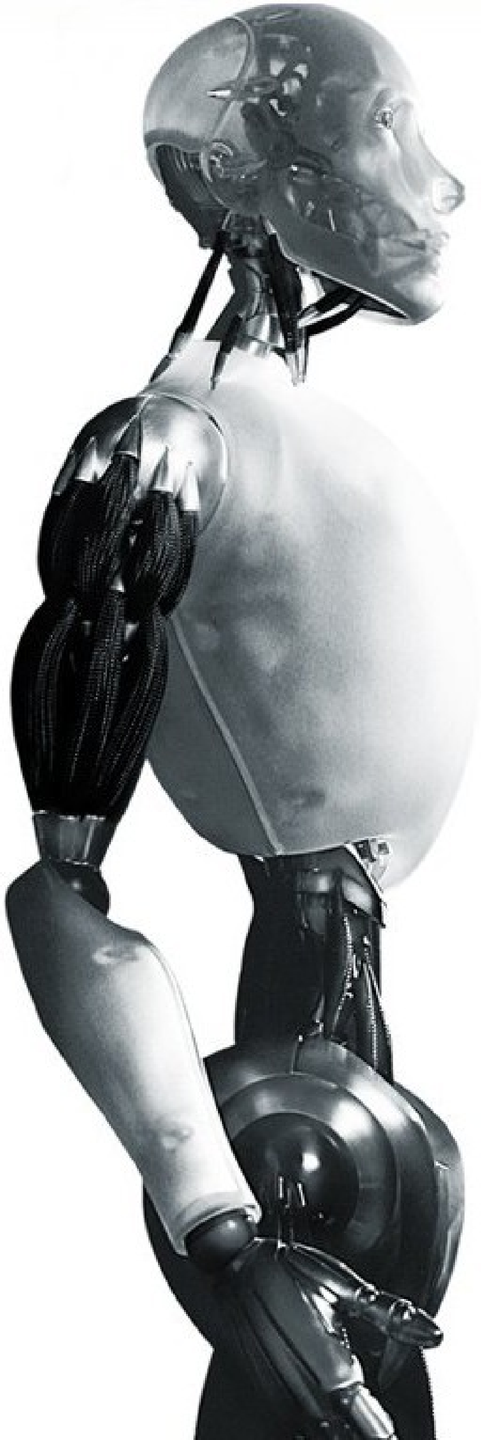
Motion Planning 9 DOF





Motion Planning 12 DOF





Robotics

Sampling-based Single Query Motion Planning

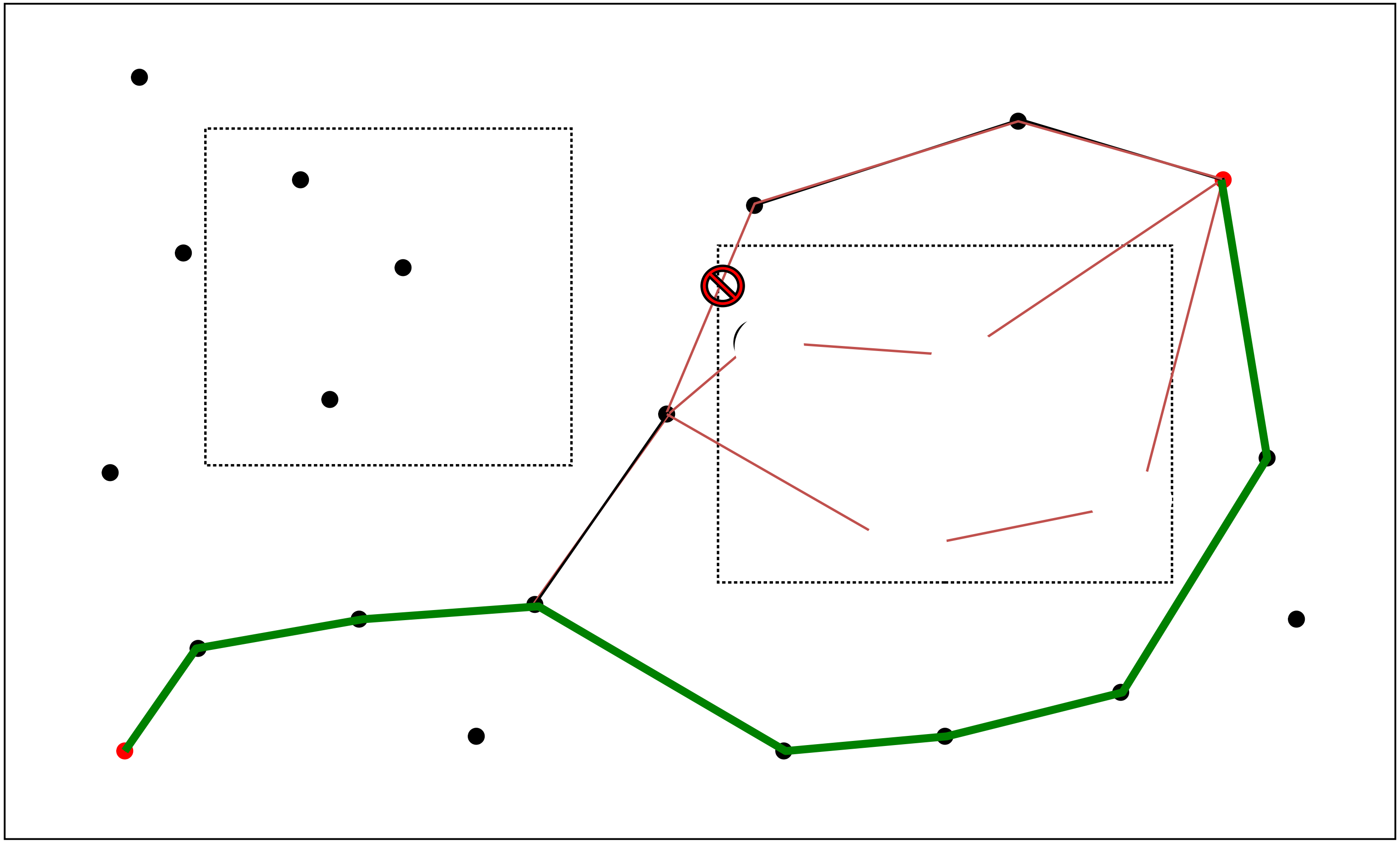
TU Berlin

Oliver Brock

Lazy PRM

- Observation: precomputation of roadmap takes a long time
- PRMs are good to answer **many** queries in an **unchanged** environment
- Precomputation only worth it if environment is "**permanently** static"
- How about
  - **single** queries?
  - **"temporarily** static"?

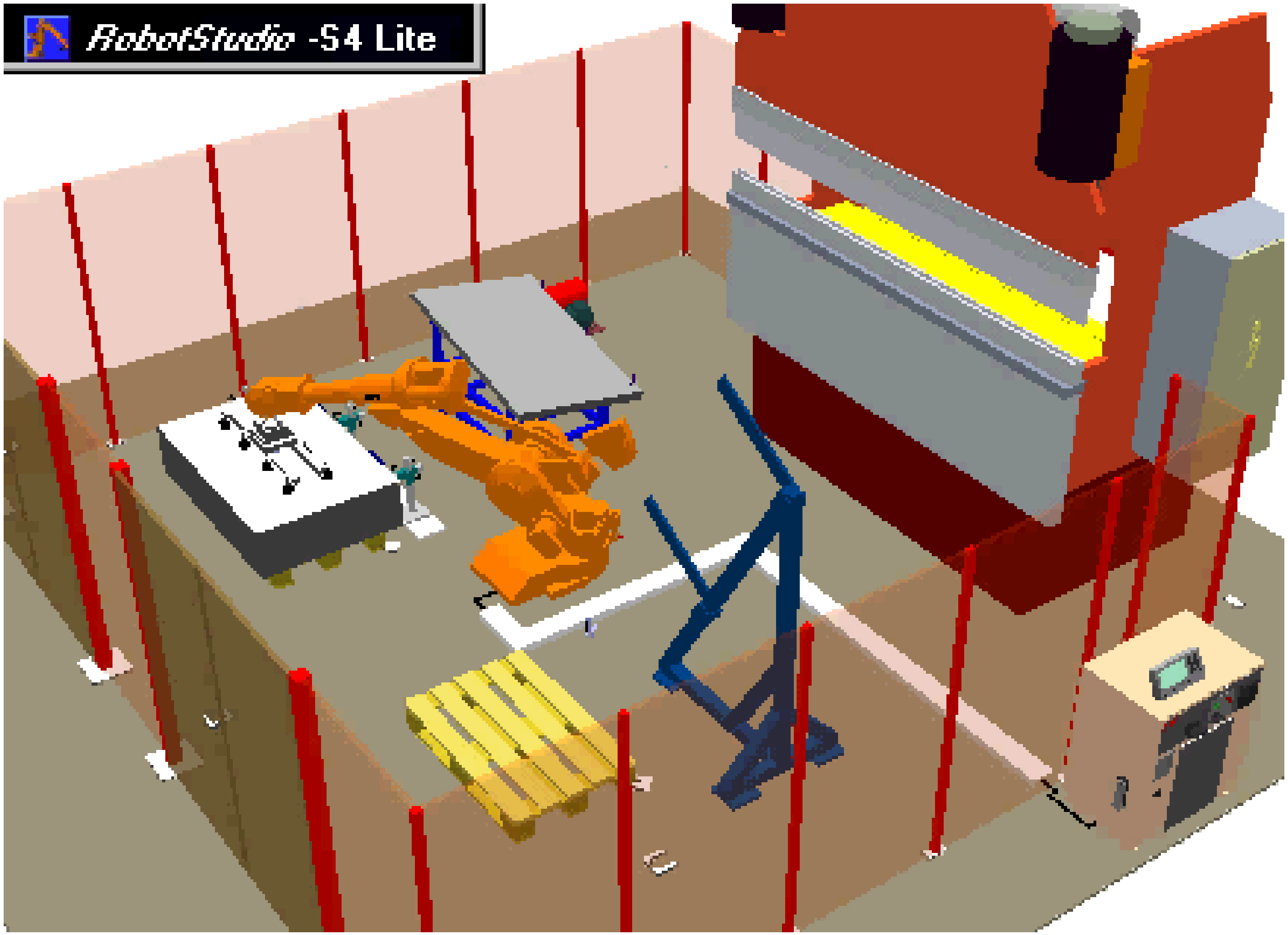
Lazy PRM



Lazy PRM

- Capable of single query planning
- Multiple queries:
  - Incremental roadmap computation
  - Individual query slower than query with PRM
  - Precomputation phase eliminated
- Attempts to minimize distance computations
- Demand-driven computational expense

Lazy PRM



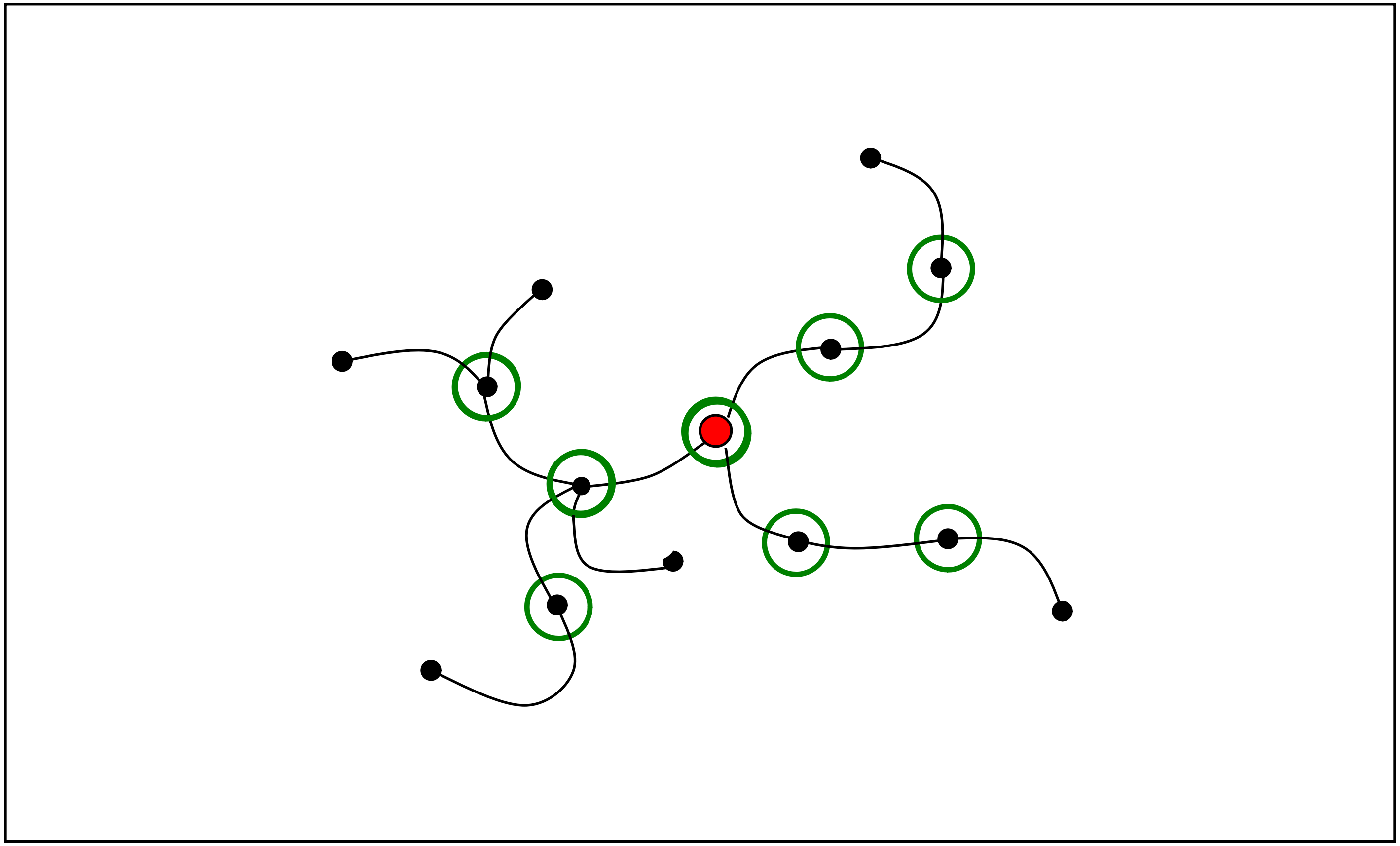
Robert Bohlin

Lazy PRM

- Average planning time of 289 seconds for previous example

- Expansion phase can be included in case of failure
- Resampling difficult areas

Rapidly-Exploring Random Trees (RRT)



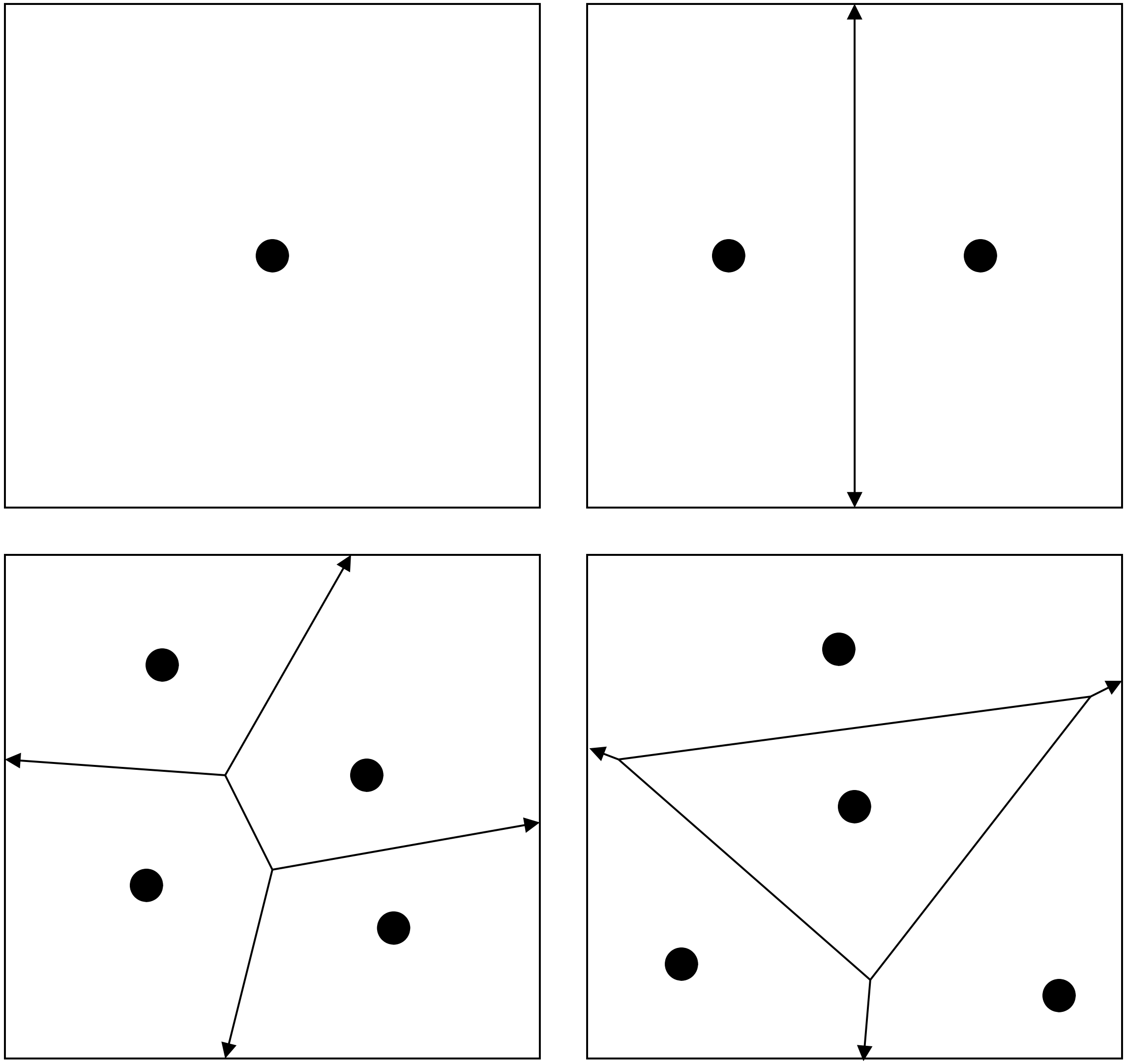
Rapidly-Exploring Random Trees

Sidebar: Voronoi Diagram

- The Post Office Problem: Which is the closest post office to every house? (Don Knuth)
- Given n **sites** in the plane
- Subdivision of plane based on proximity

Georgy Voronoi
1868-1908

Voronoi Diagram



Uses for Voronoi Diagram

- **Anthropology and Archeology** -- Identify the parts of a region under the influence of different neolithic clans, chiefdoms, ceremonial centers, or hill forts.
- **Astronomy** -- Identify clusters of stars and clusters of galaxies (Here we saw what may be the earliest picture of a Voronoi diagram, drawn by Descartes in 1644, where the regions described the regions of gravitational influence of the sun and other stars.)
- **Biology, Ecology, Forestry** -- Model and analyze plant competition ("Area potentially available to a tree", "Plant polygons")
- **Cartography** -- Piece together satellite photographs into large "mosaic" maps
- **Crystallography and Chemistry** -- Study chemical properties of metallic sodium ("Wigner-Seitz regions"); Modelling alloy structures as sphere packings ("Domain of an atom")
- **Finite Element Analysis** -- Generating finite element meshes which avoid small angles
- **Geography** -- Analyzing patterns of urban settlements
- **Geology** -- Estimation of ore reserves in a deposit using information obtained from bore holes; modelling crack patterns in basalt due to contraction by cooling
- **Geometric Modeling** -- Finding "good" triangulations of 3D surfaces
- **Marketing** -- Model market of US metropolitan areas; market area extending down to individual retail stores
- **Mathematics** -- Study of positive definite quadratic forms ("Dirichlet tesselation", "Voronoi diagram")
- **Metallurgy** -- Modelling "grain growth" in metal films
- **Meteorology** -- Estimate regional rainfall averages, given data at discrete rain gauges ("Thiessen polygons")
- **Pattern Recognition** -- Find simple descriptors for shapes that extract 1D characterizations from 2D shapes ("Medial axis" or "skeleton" of a contour)
- **Physiology** -- Analysis of capillary distribution in cross-sections of muscle tissue to compute oxygen transport ("Capillary domains")
- **Robotics** -- Path planning in the presence of obstacles
- **Statistics and Data Analysis** -- Analyze statistical clustering ("Natural neighbors" interpolation)
- **Zoology** -- Model and analyze the territories of animals

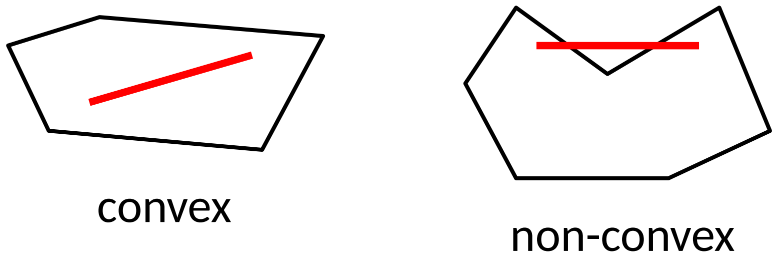
Facts about Voronoi

- A site has an unbounded region if and only if it lies on the convex hull of all sites.
- All Voronoi regions are convex.

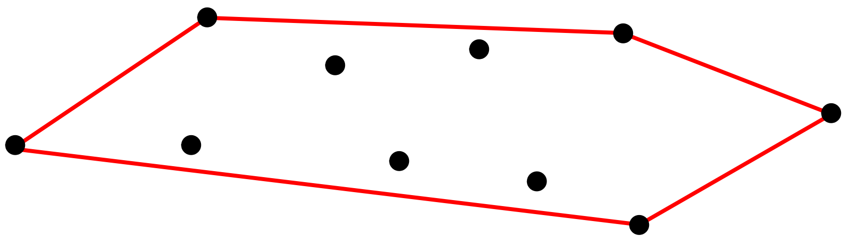
- What is convex?

Sidebar: Convexity

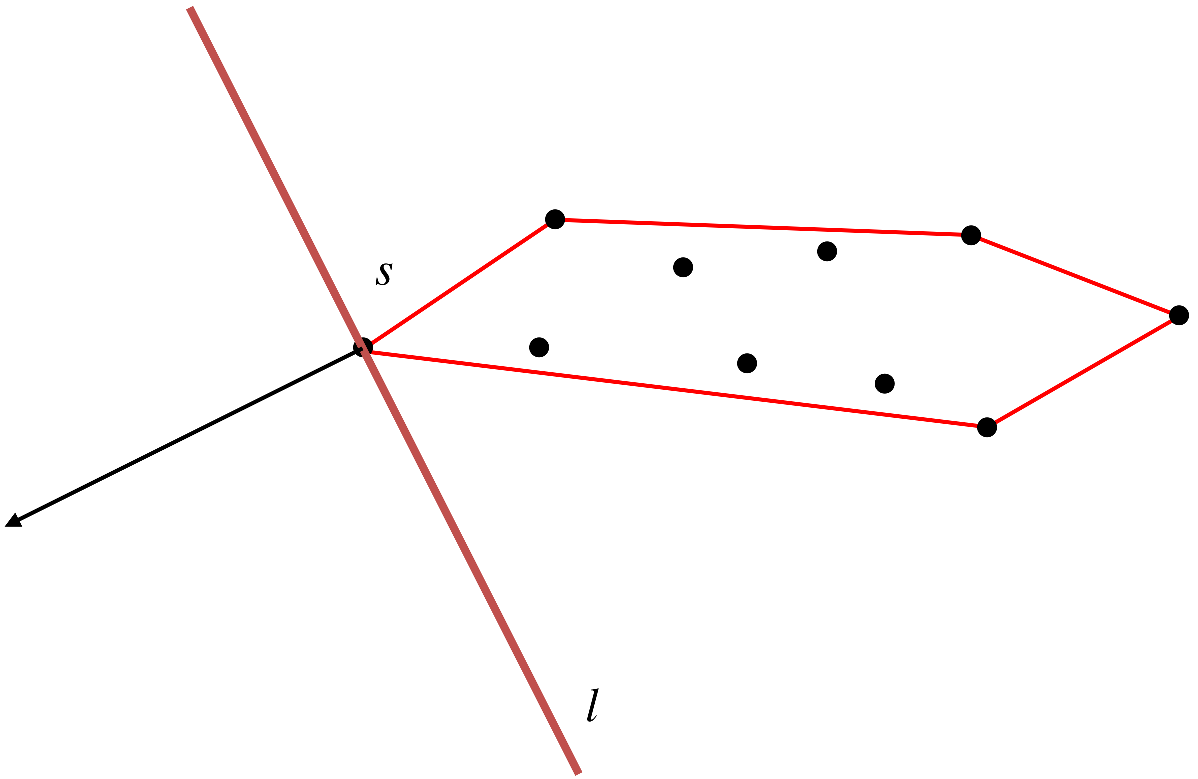
- A polygon P is convex if a line connecting any two points $p_1, p_2 \in P$ is entirely contained in P.



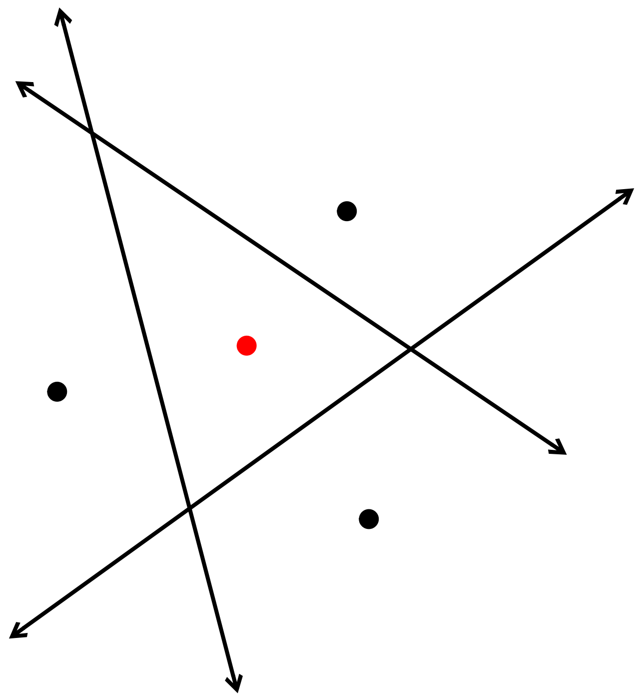
- The convex hull $\mathrm{hull}(S)$ of a set of points S is the smallest polygon P for which all points in S are either on the boundary or on the inside of P.



A site s has an unbounded region if and only if it lies on the convex hull of all sites.

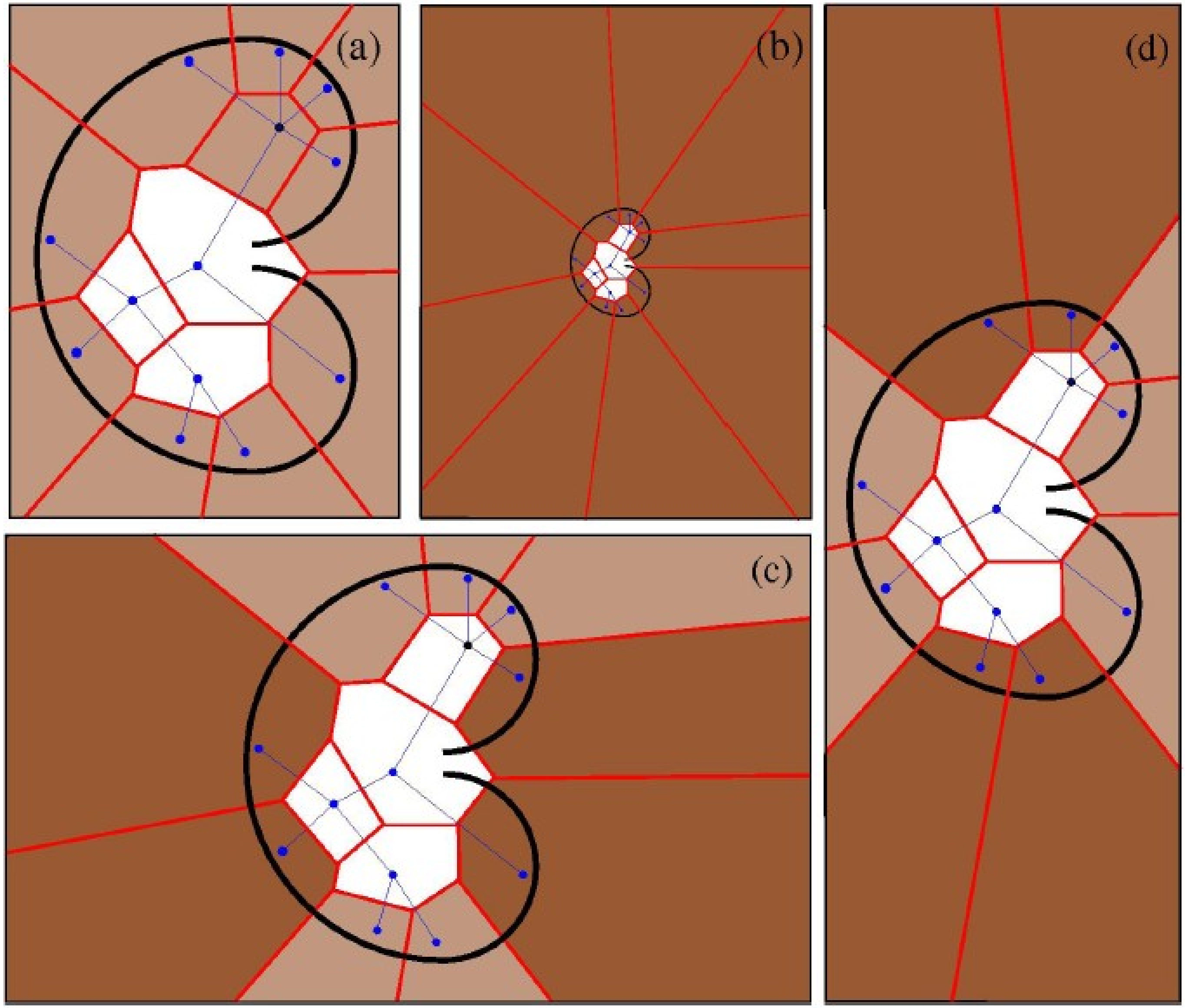


All Voronoi regions are convex.

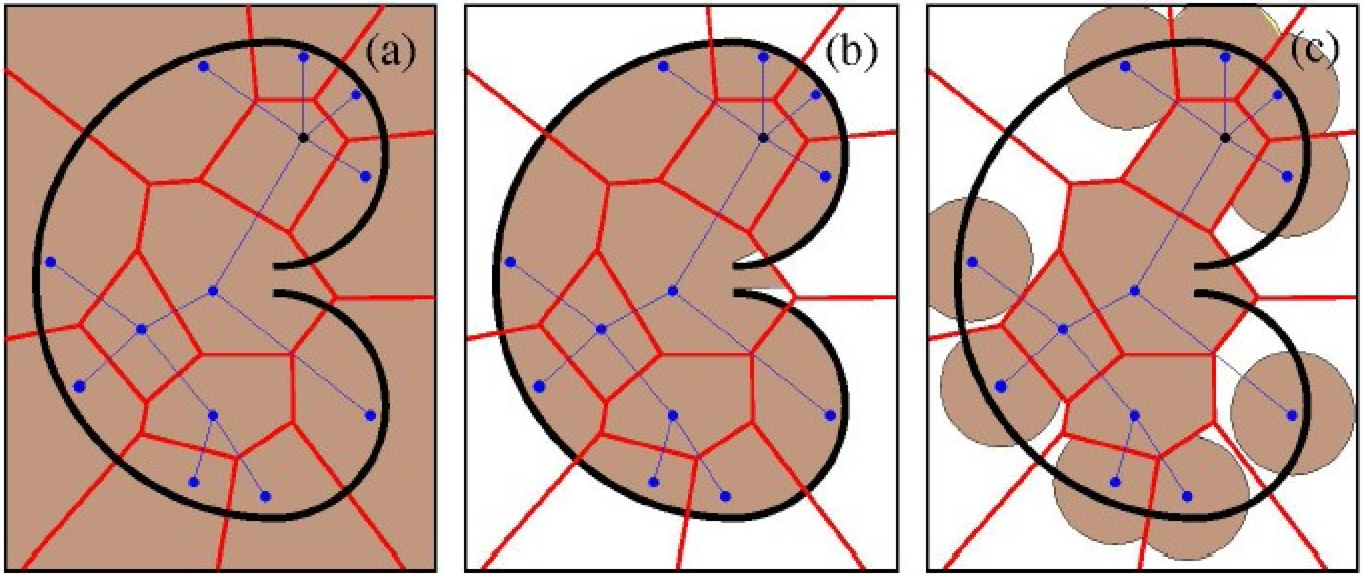


Properties of RRT

- $P_{\text{point } i}$ (extended)?
- Difference Lazy PRM / RRT?
  – Explored area (goal-directed vs. uniform)
  – Integration of constraints
  – Step-size versus sampling density

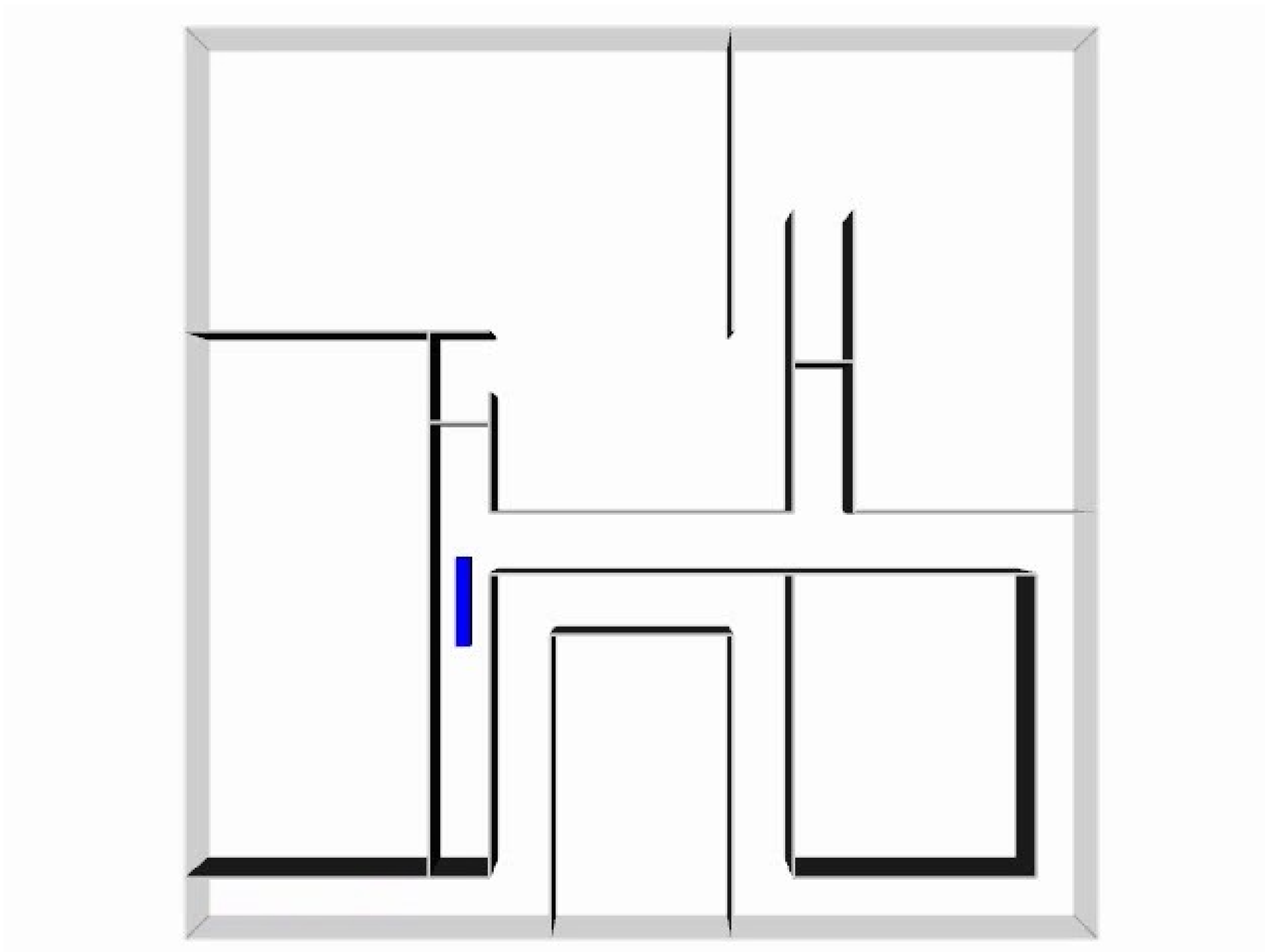

(a)
(b)
(c)
(d)

Adaptive Domain RRT

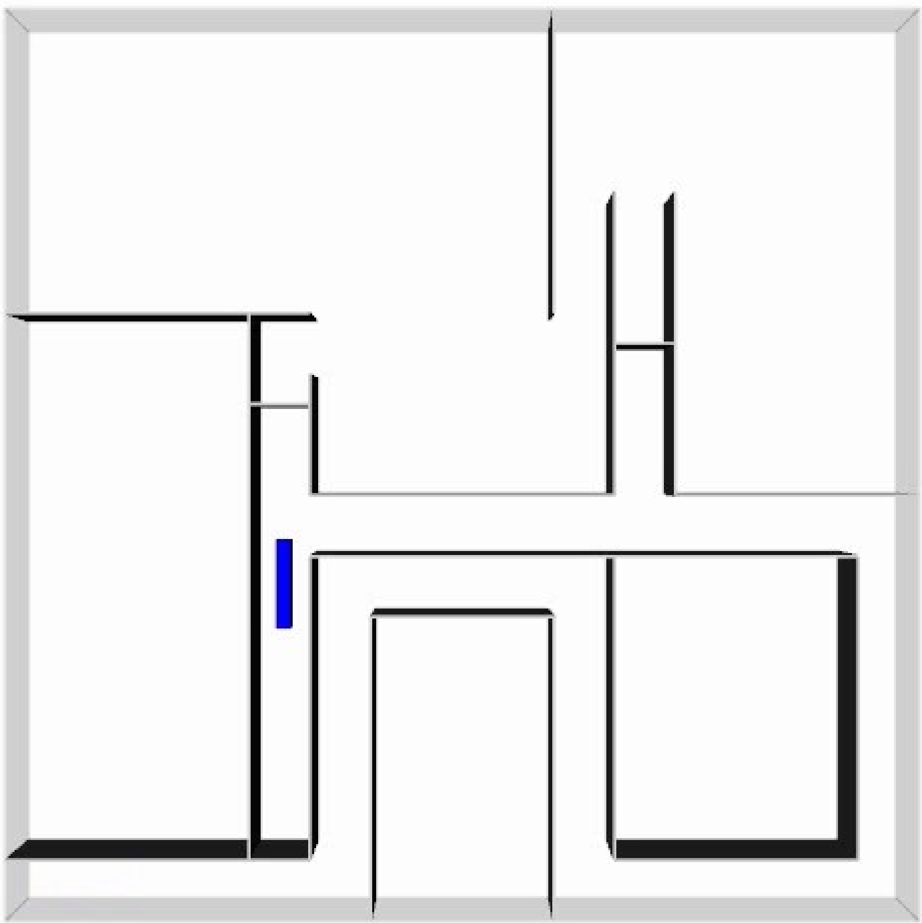


The Ideal “Planner” or Search

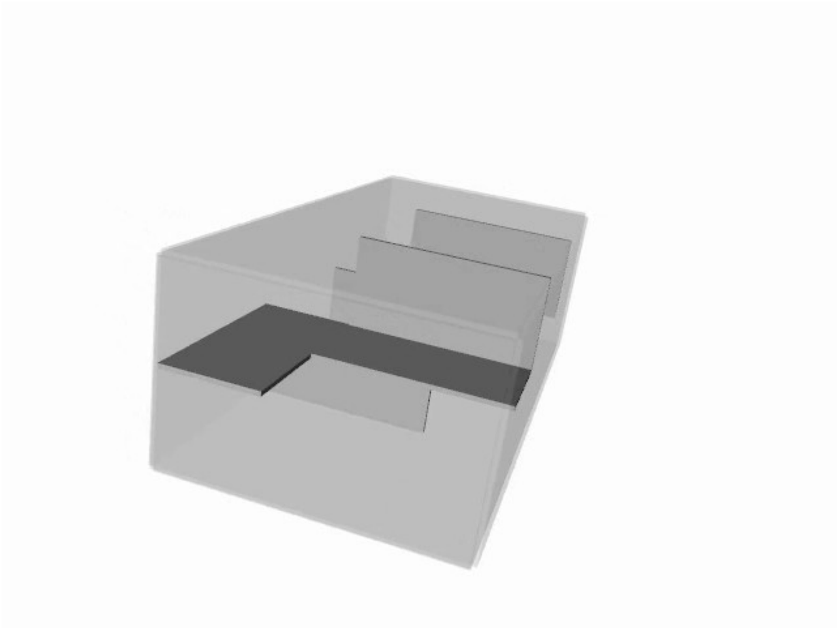
EXPLORATION  EXPLOITATION



ADD-RRT: Jaillet, Yershova, LaValle, Simeon 2005

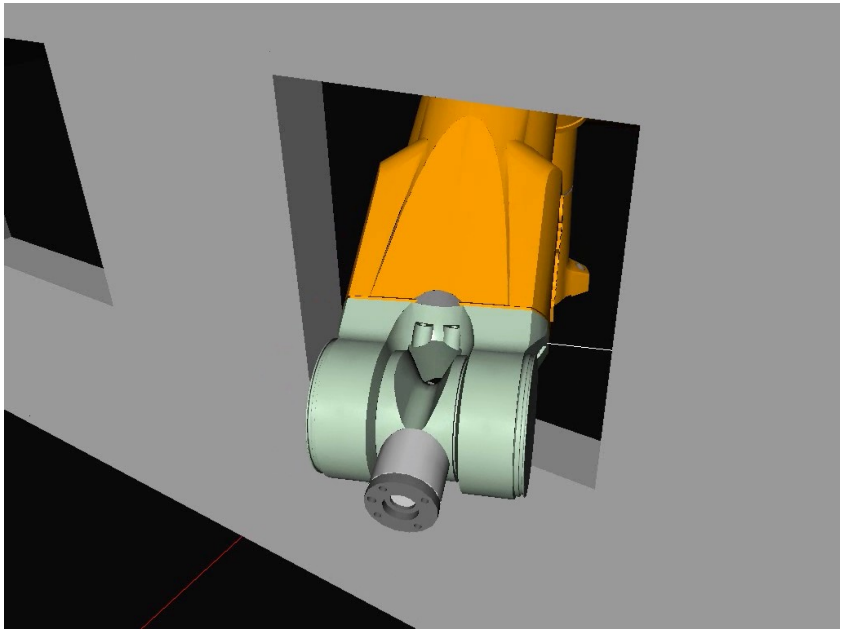


Exploiting workspace information



Workspace connectivity

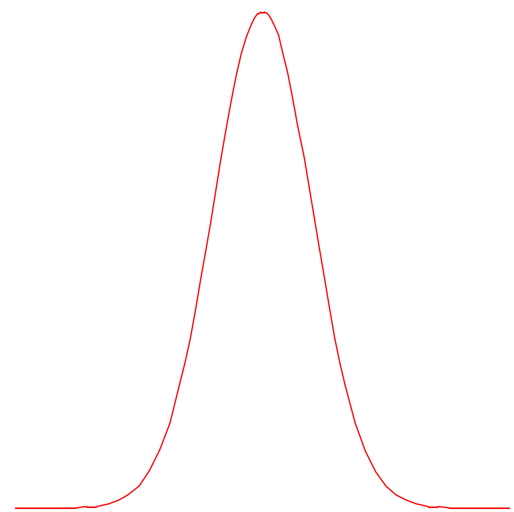
$$\tau = J^T F$$

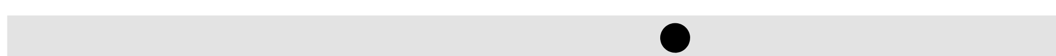


Proximity to obstacles

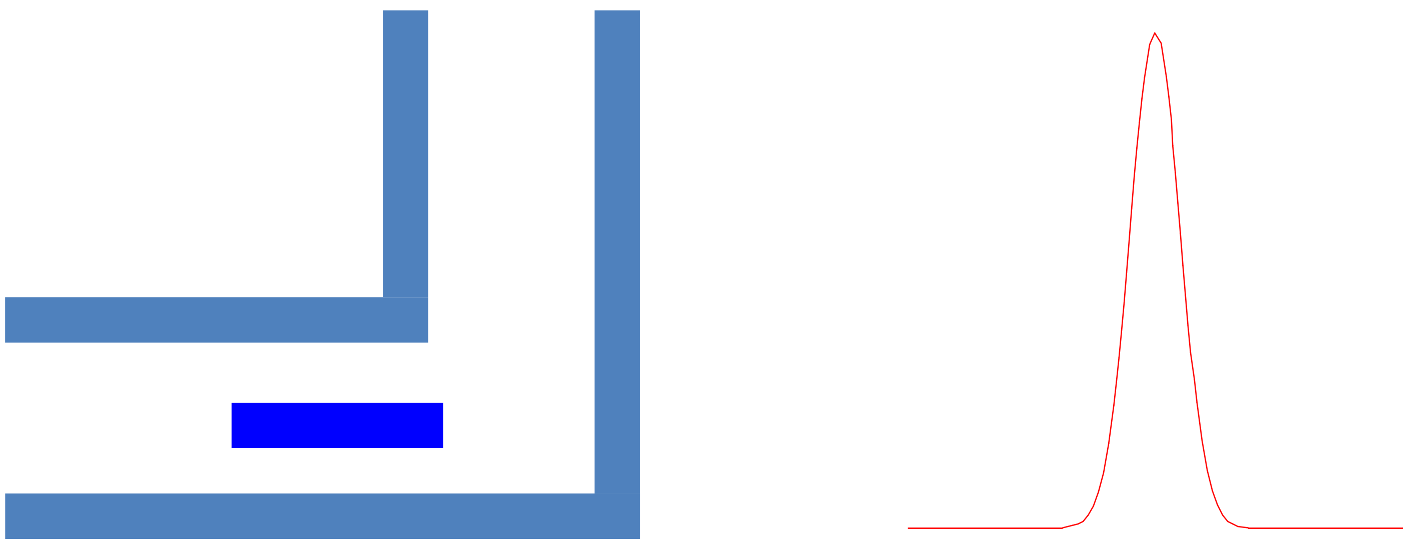
$$\tau = \sum_i J_i^T F_i$$

Balancing exploration and exploitation

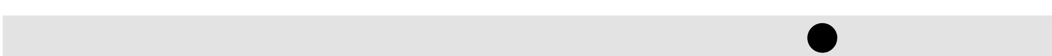


EXPLORATION  EXPLOITATION

Balancing exploration and exploitation

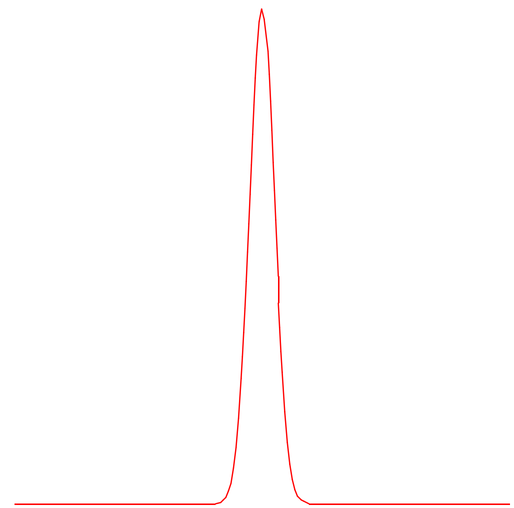


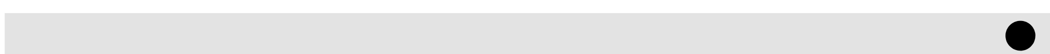
EXPLORATION



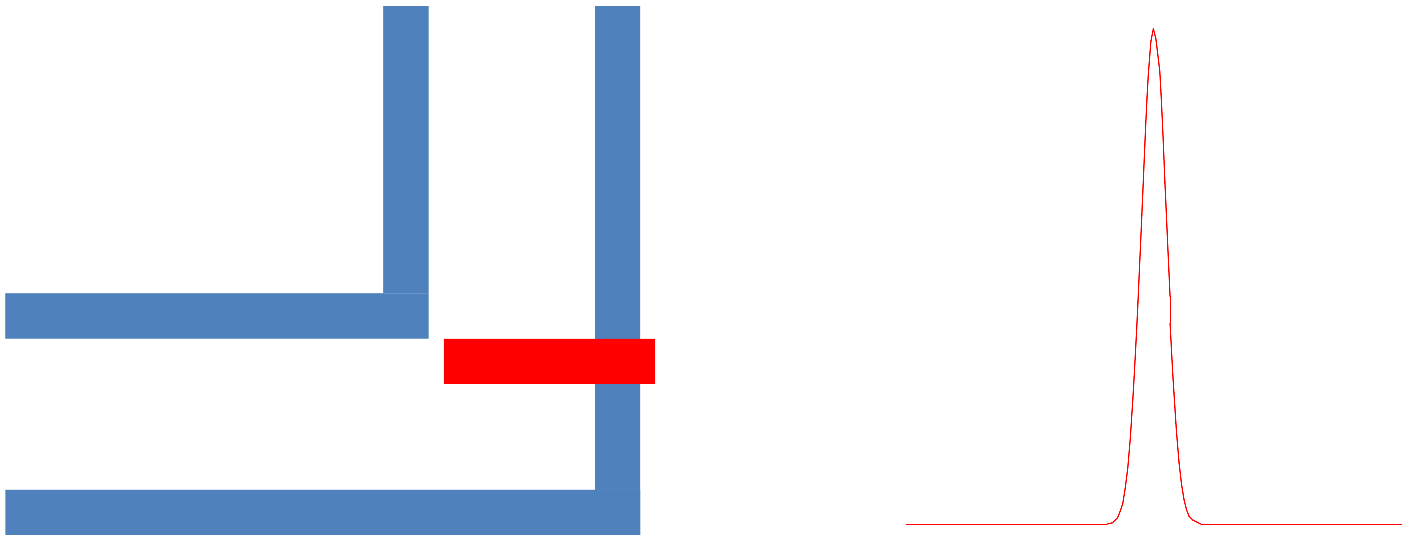
EXPLOITATION

Balancing exploration and exploitation



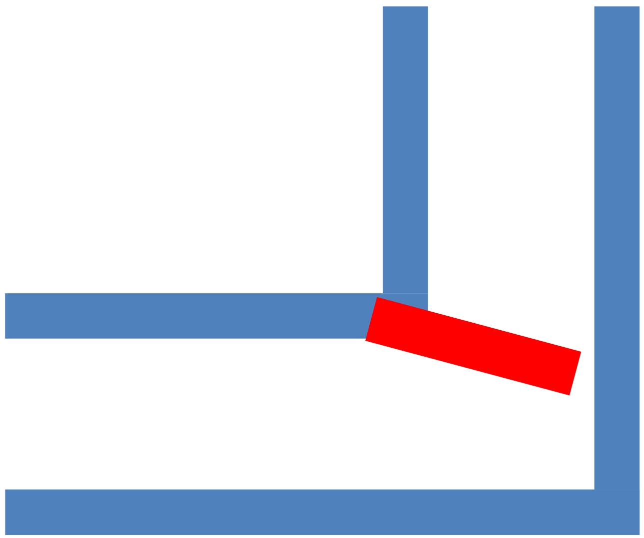
EXPLORATION  EXPLOITATION

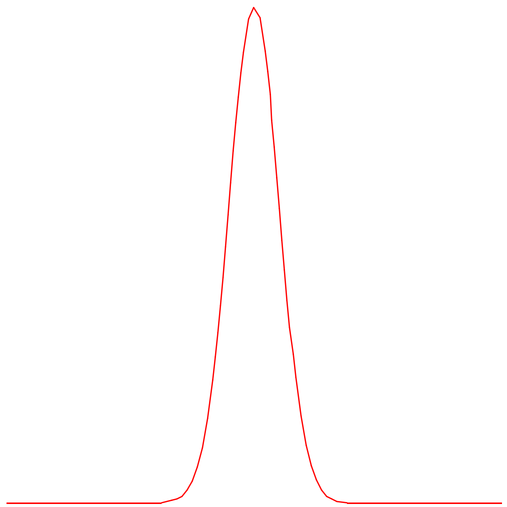
Balancing exploration and exploitation



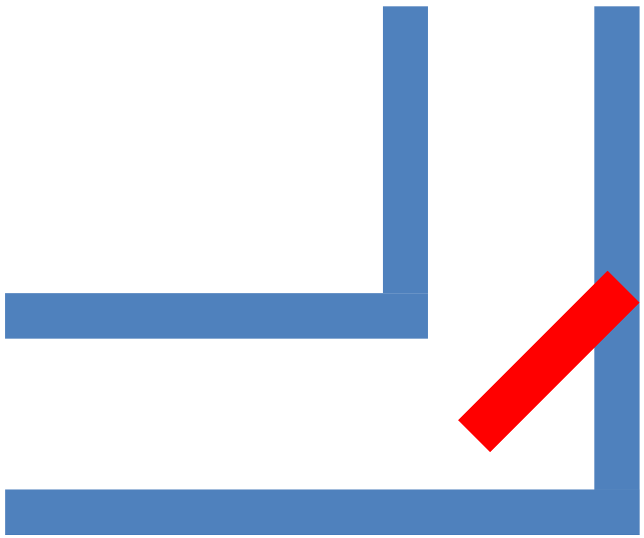
EXPLORATION  EXPLOITATION

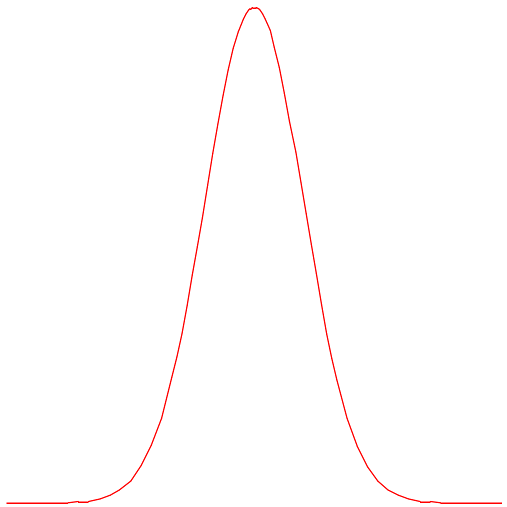
Balancing exploration and exploitation

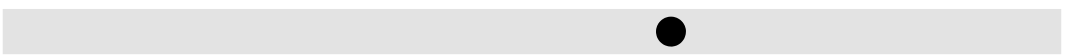




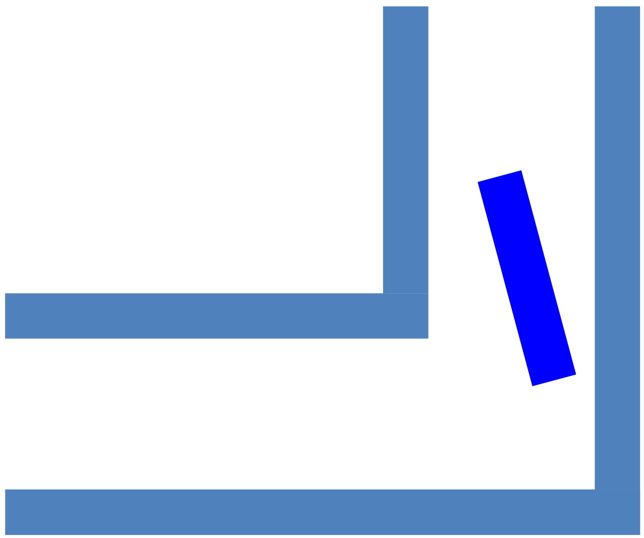
Balancing exploration and exploitation

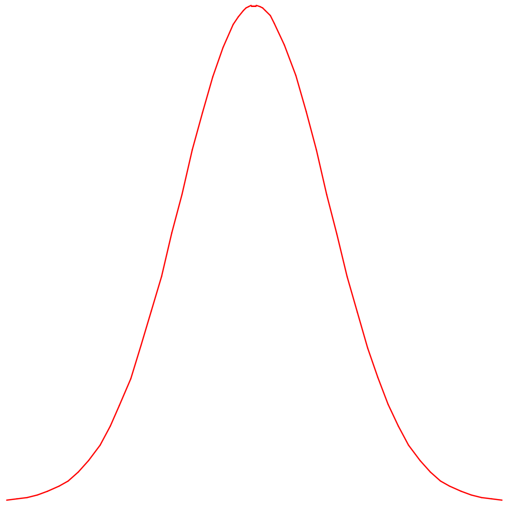




EXPLORATION  EXPLOITATION

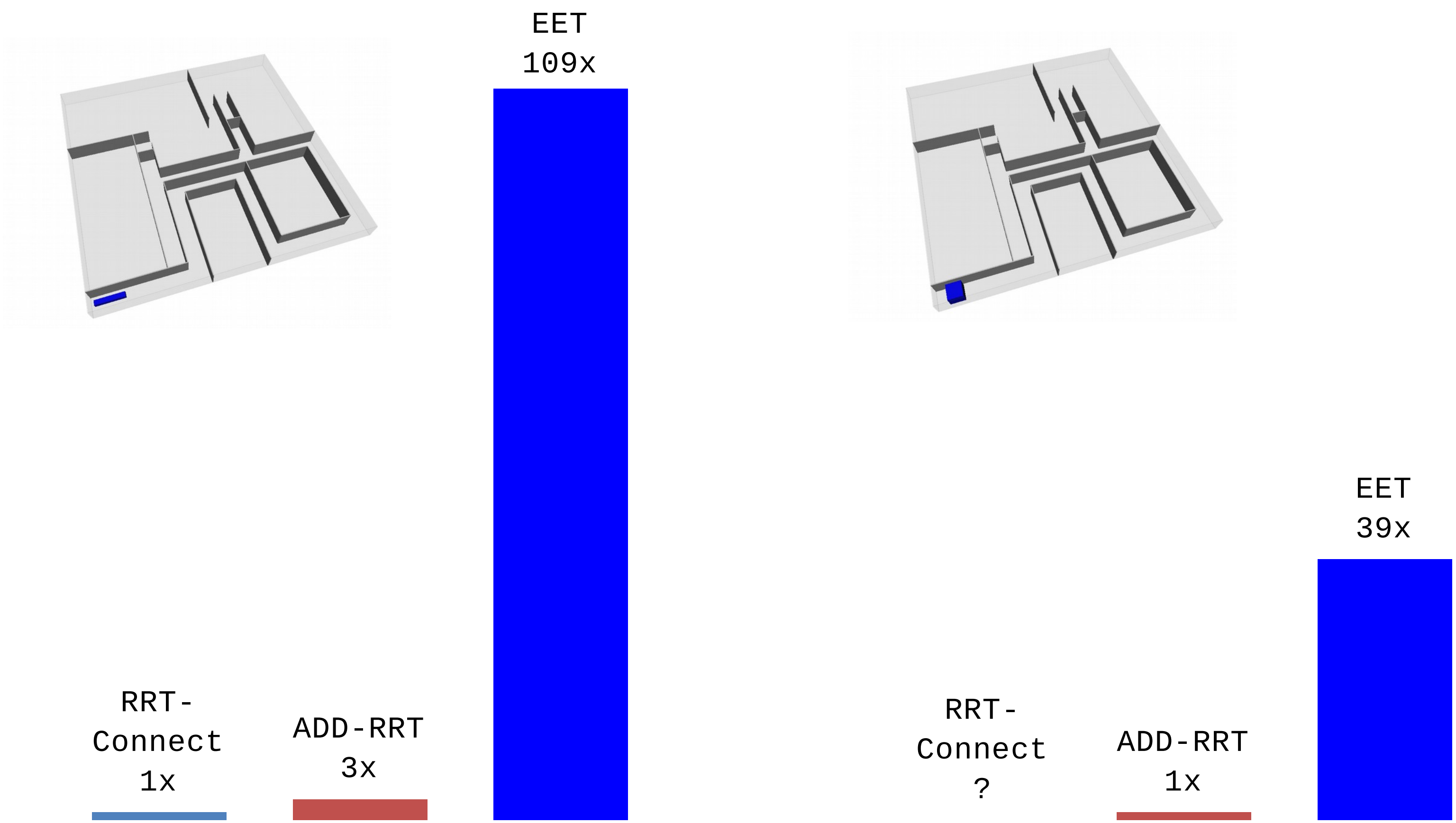
Balancing exploration and exploitation





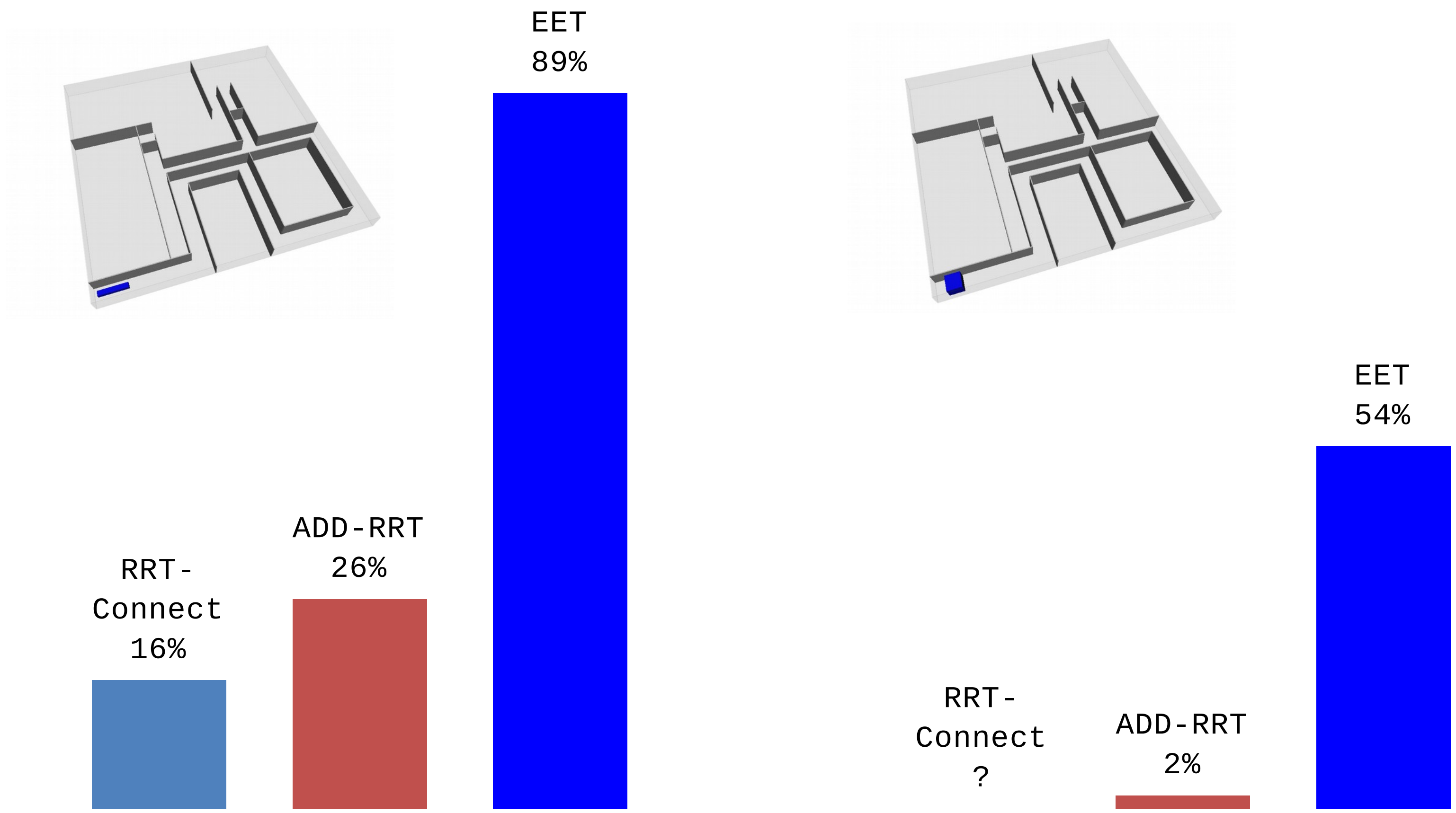
EXPLORATION  EXPLOITATION

Speedup



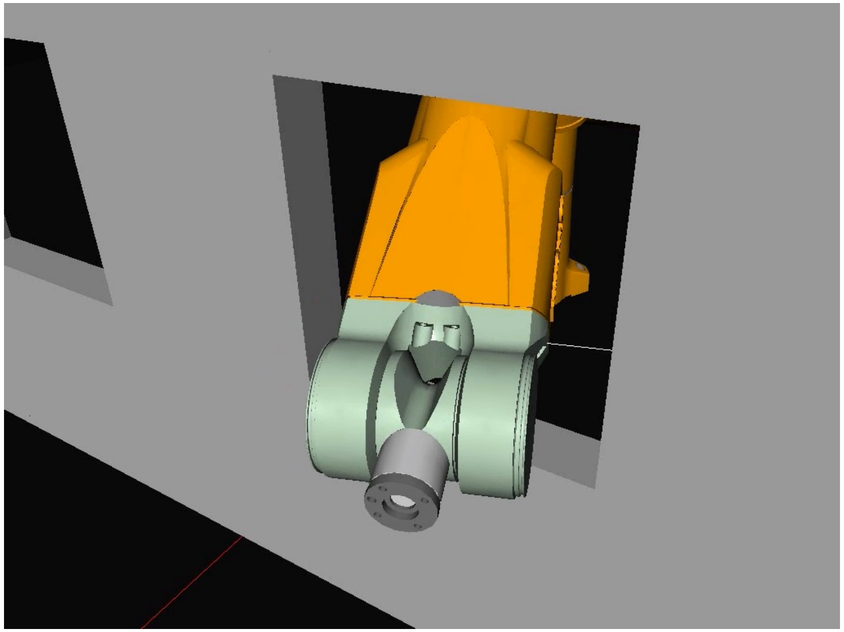
RRT-Connect: Kuffner and LaValle 2000
ADD-RRT: Jaillet, Yershova, LaValle, Simeon 2005

Effectiveness of exploitation
(% of non-colliding samples)



RRT-Connect: Kuffner and LaValle 2000
ADD-RRT: Jaillet, Yershova, LaValle, Simeon 2005

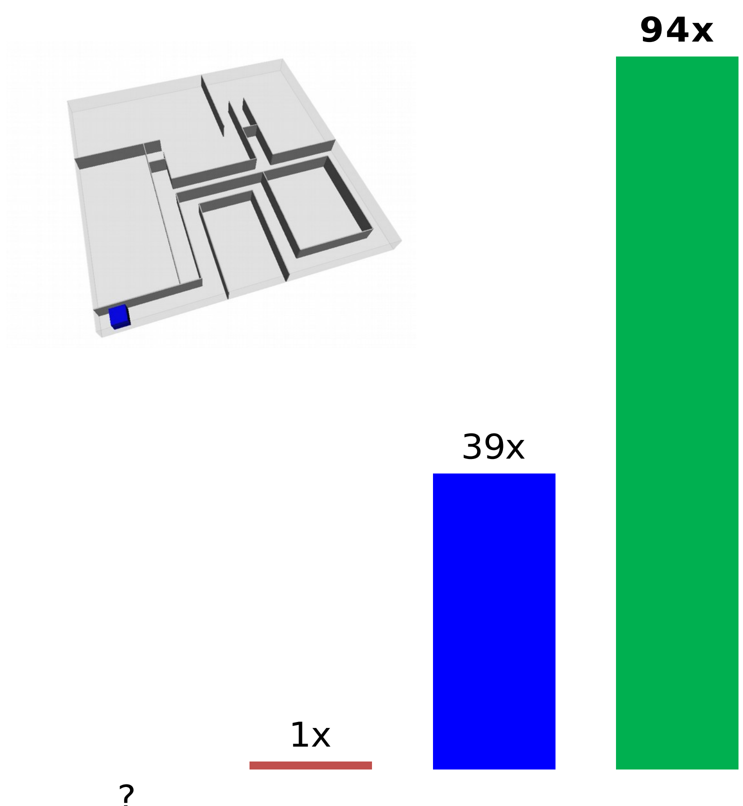
More information for better exploitation

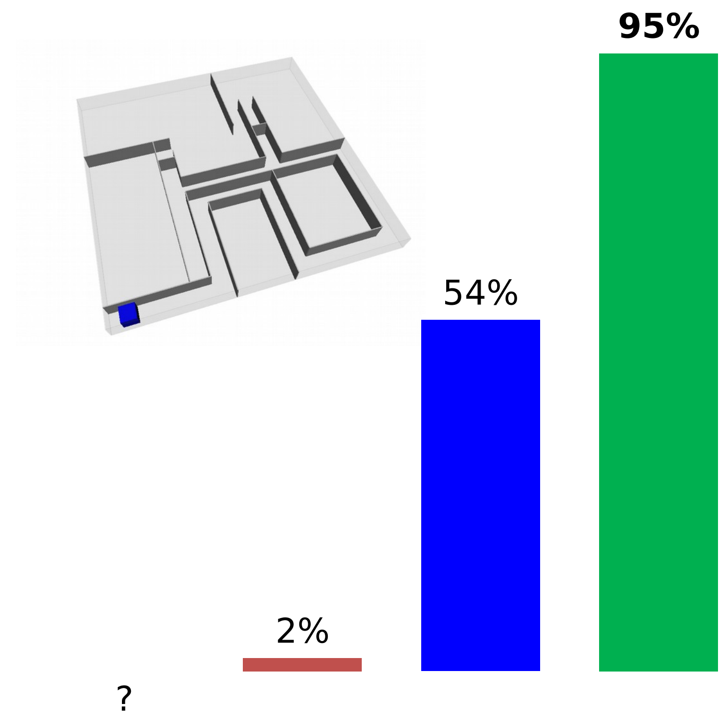


Repulsive forces

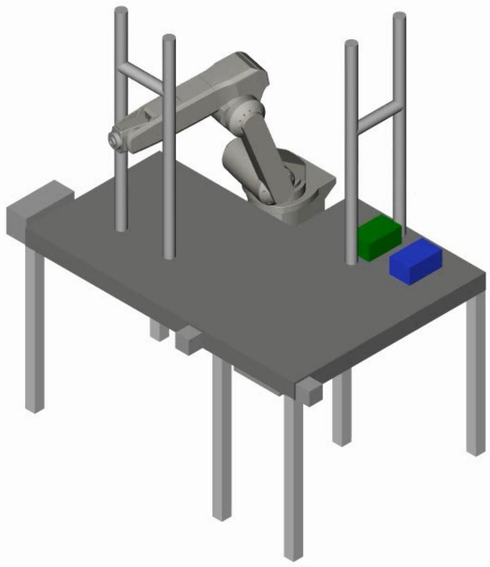
$$\tau = \sum_i J_i^T F_i$$

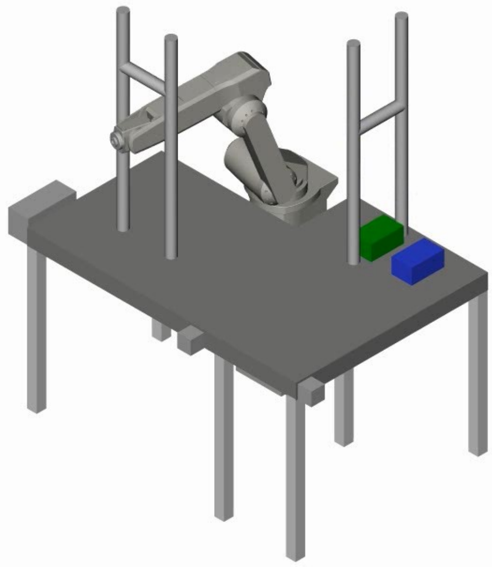
Speedup / Effectiveness of exploitation

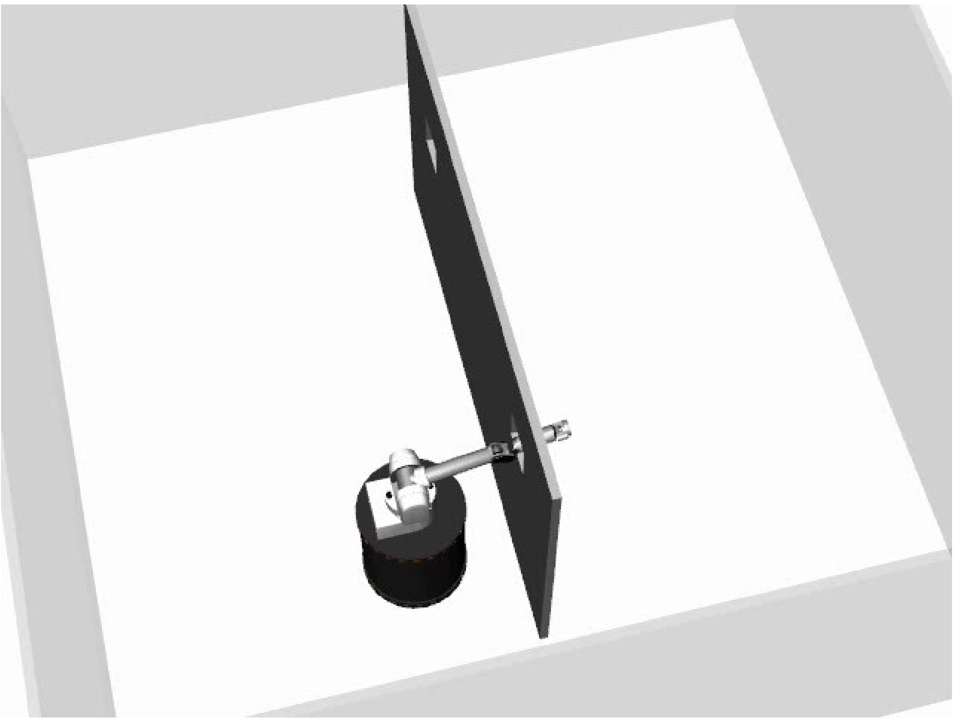
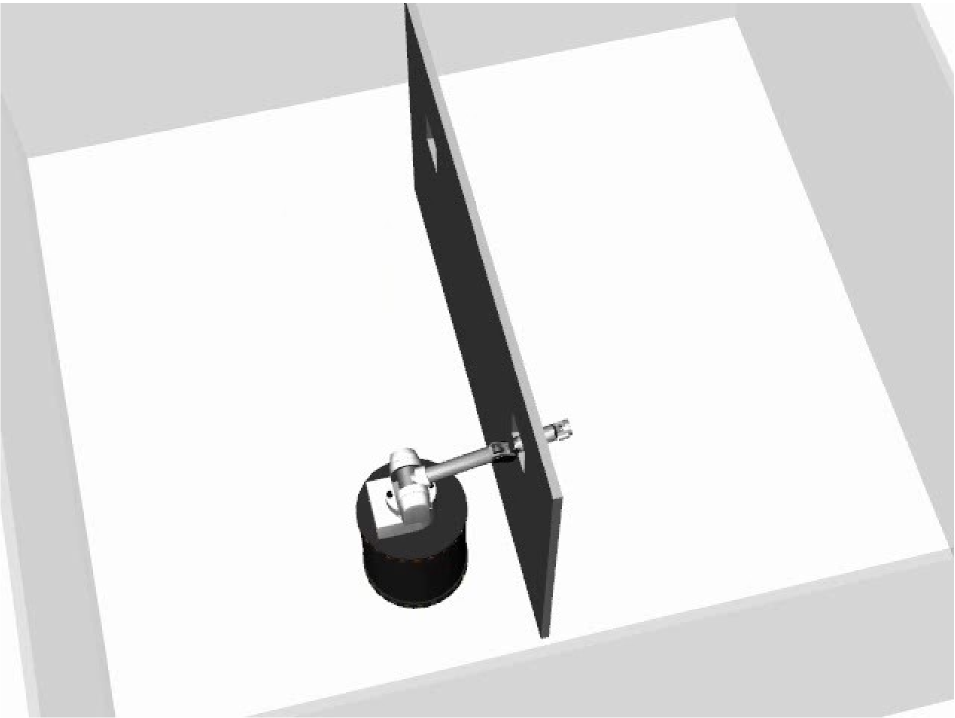




RRT-Connect: Kuffner and LaValle 2000
ADD-RRT: Jaillet, Yershova, LaValle, Simeon 2005

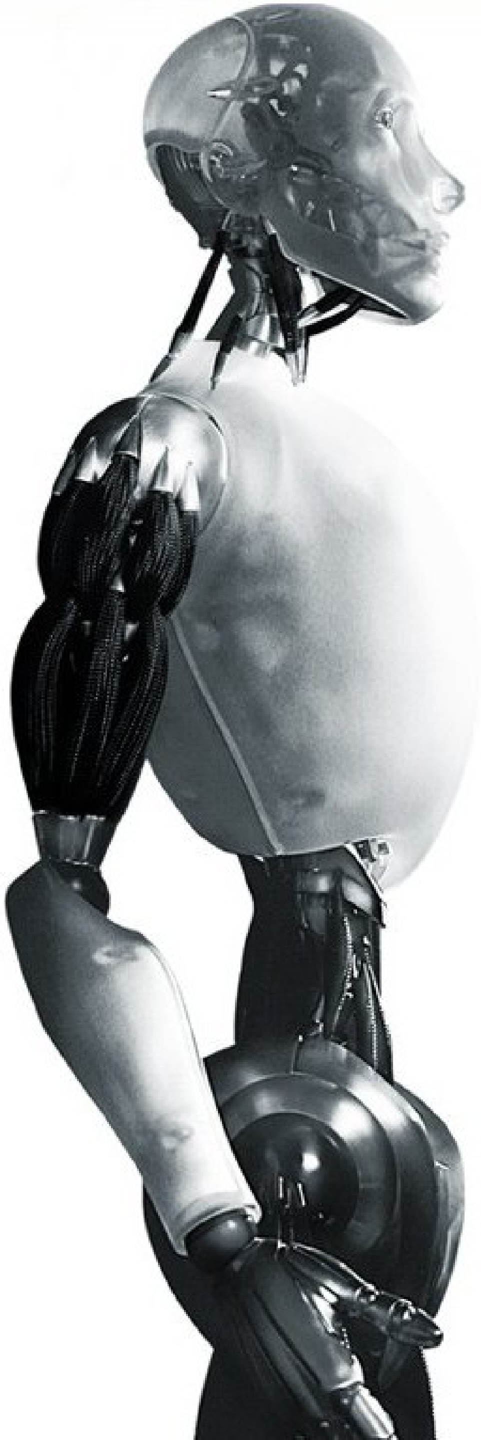




Kinodynamic Planning with RRT

- Easy to integrate local planners for
  - Kinematic constraints
  - Dynamic constraints
- Expand C-space to state space for velocity representation (2 *d* dimensions)
- Requires known dynamic model (grasping!)
- Planning times for 7 dof ~ 10-20 seconds (holonomic, no dynamics)



Robotics

Motion Planning – Some History

TU Berlin
Oliver Brock

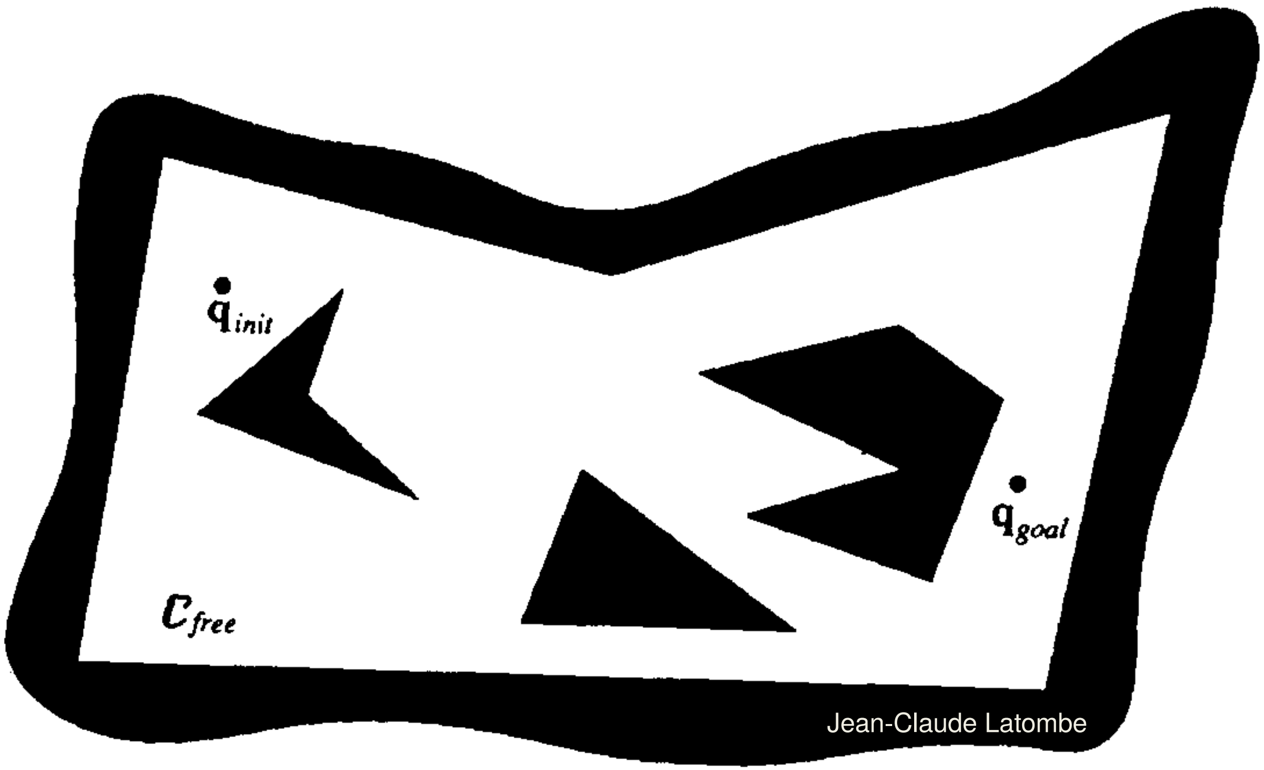
Origins of Motion Planning

- T. Lozano-Pérez and M.A. Wesley:
  “An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles”, 1979.
- Introduced the notion of C-space to robotics
- Many approaches have been devised since then in C-space

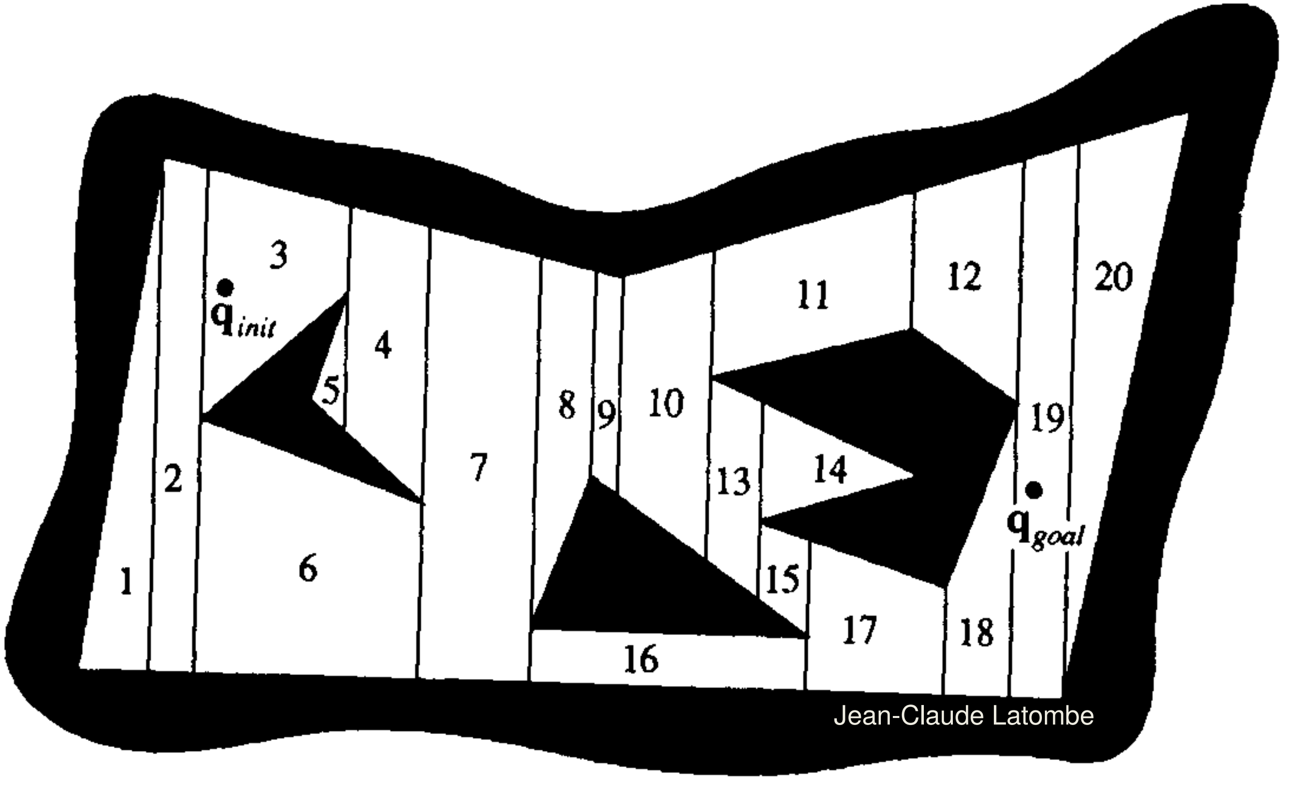
4 Basic Motion Planning Methods

- Exact Cell Decomposition
- Approximate Cell Decomposition
- Roadmap Methods
- Potential Field Methods

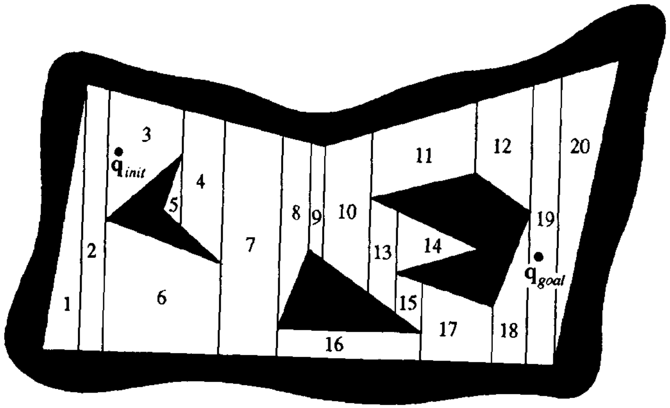
Exact Cell Decomposition

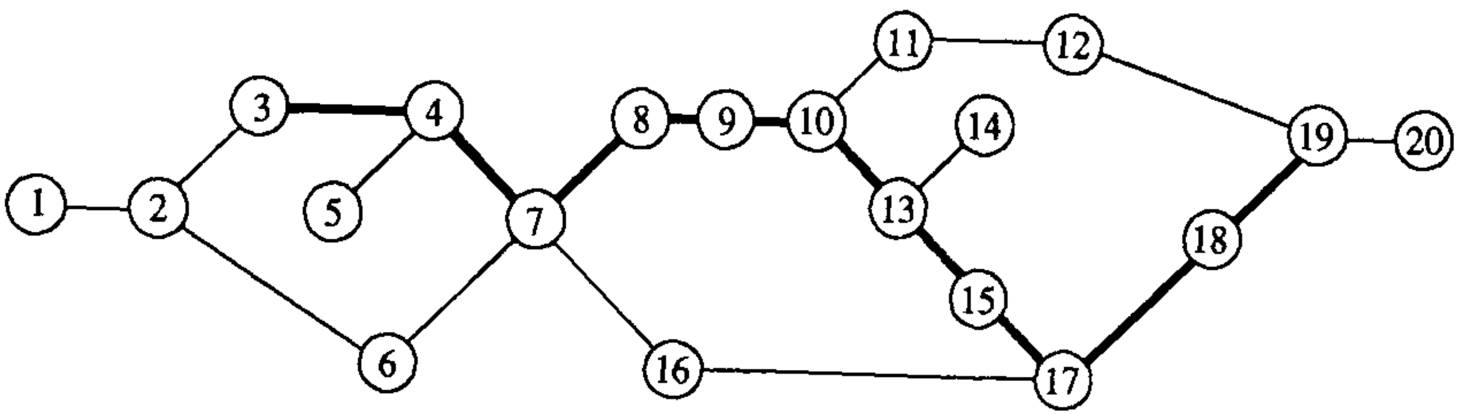


Exact Cell Decomposition cont.

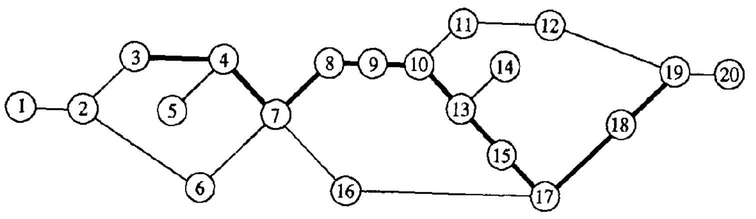


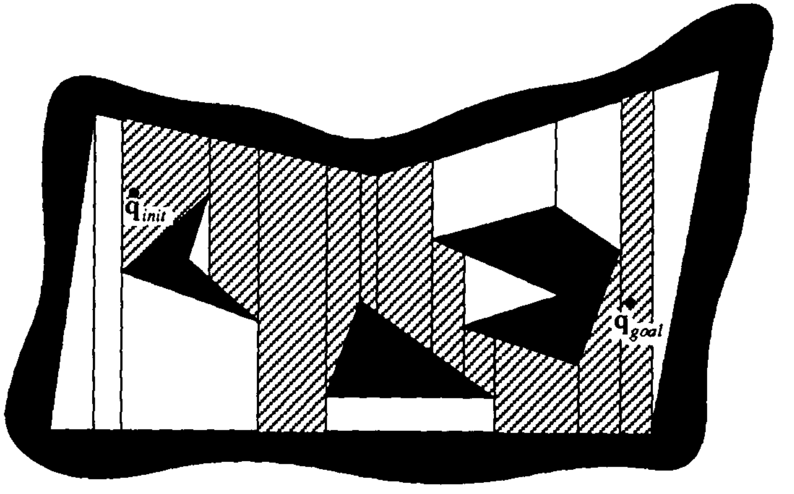
Exact Cell Decomposition cont.

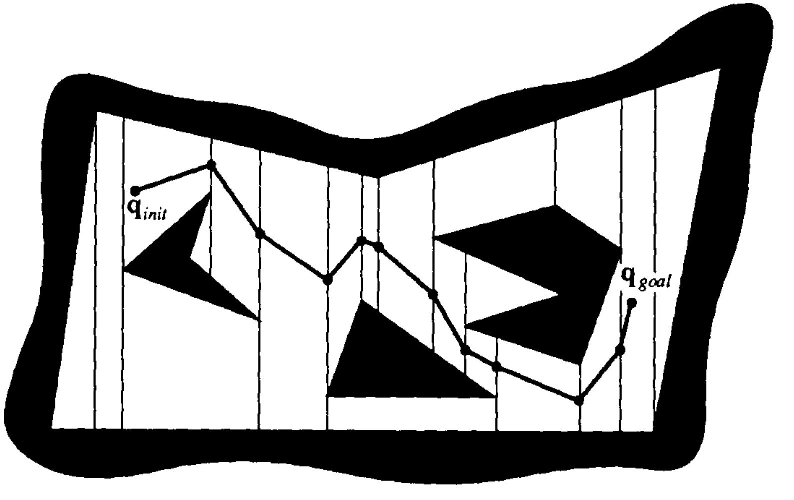




Exact Cell Decomposition cont.







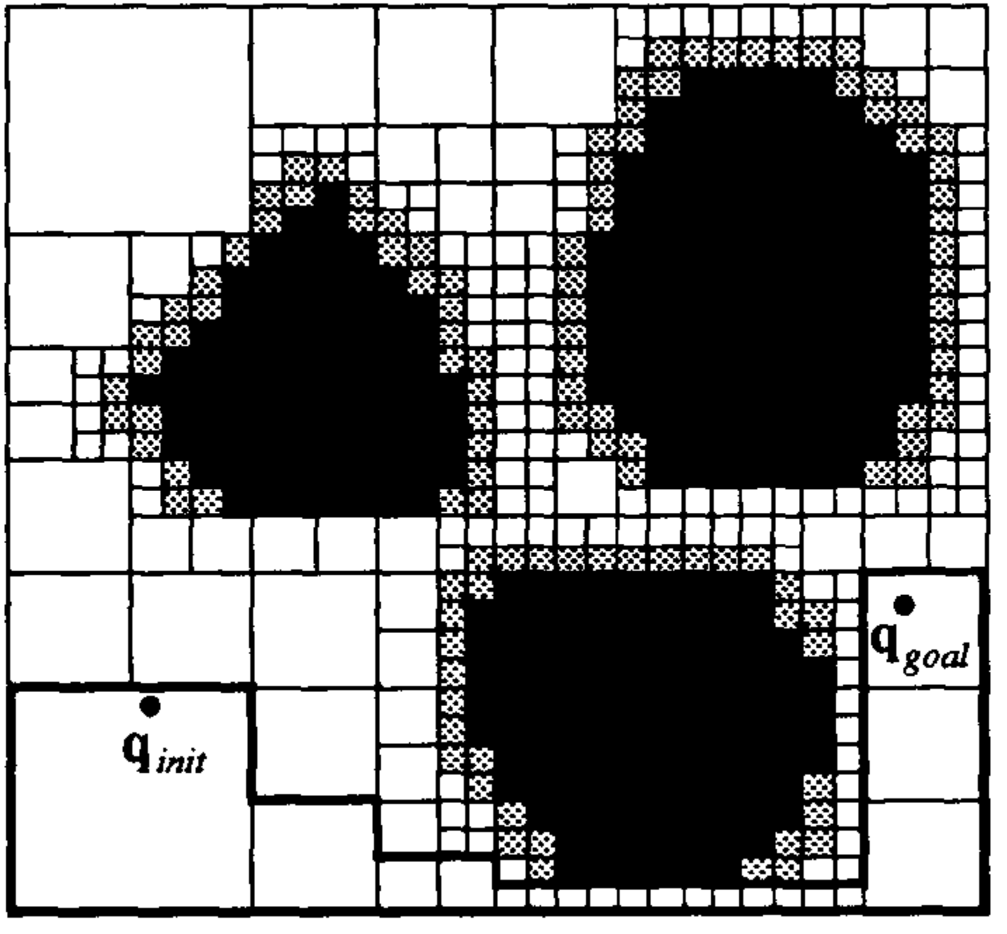
Additional Exact Cell Decomp. Methods

- Stick robot (line segment) in the plane
- General case
  - "…makes use of a well-known result established by Collins ('75) for deciding the satisfiability of Tarski sentences."
- We'll pass…
  - Complicated
  - Impractical
  - Computationally complex
  - But theoretically very appealing!

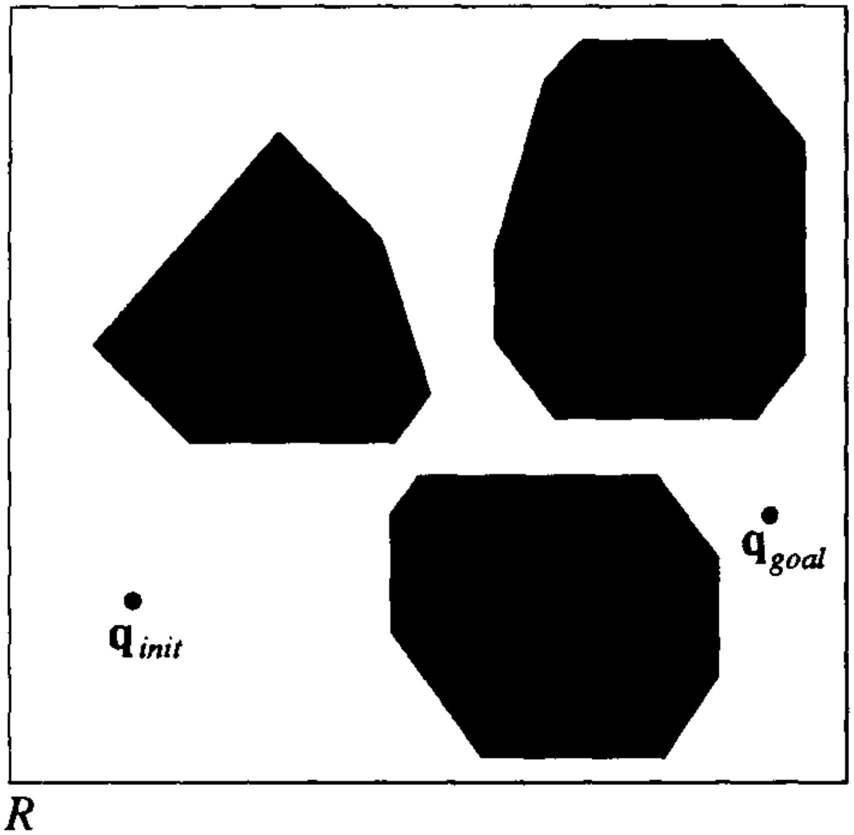
From Exact to Approximate

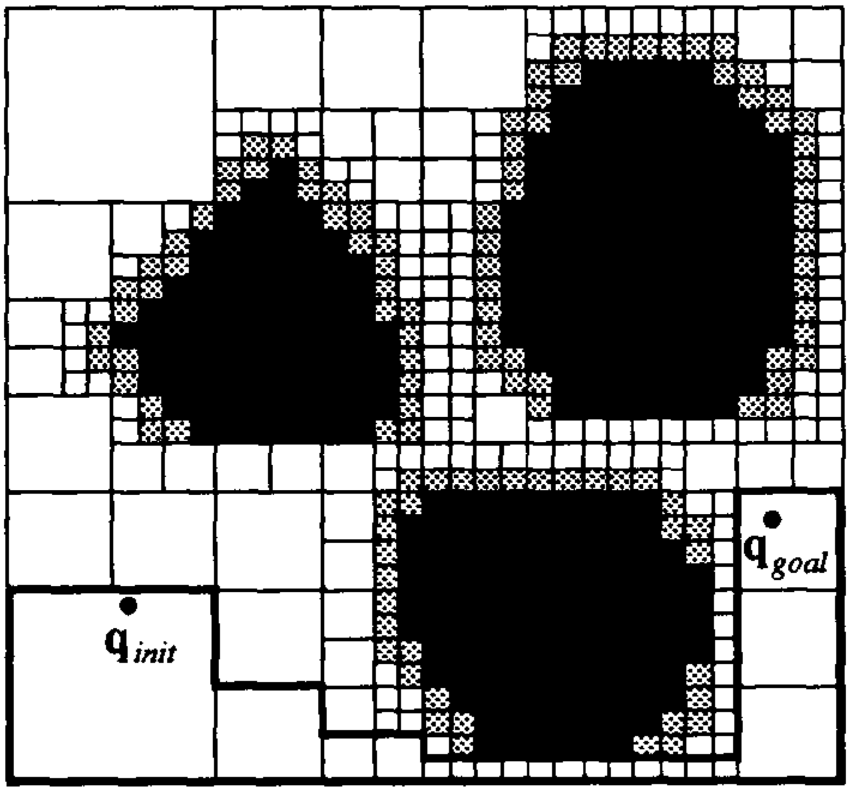
- Maybe we don’t even need to know **everything** there is to know…
- Incremental refinement of our understanding until we find a solution

2^n-Tree



Approximate Cell Decomposition





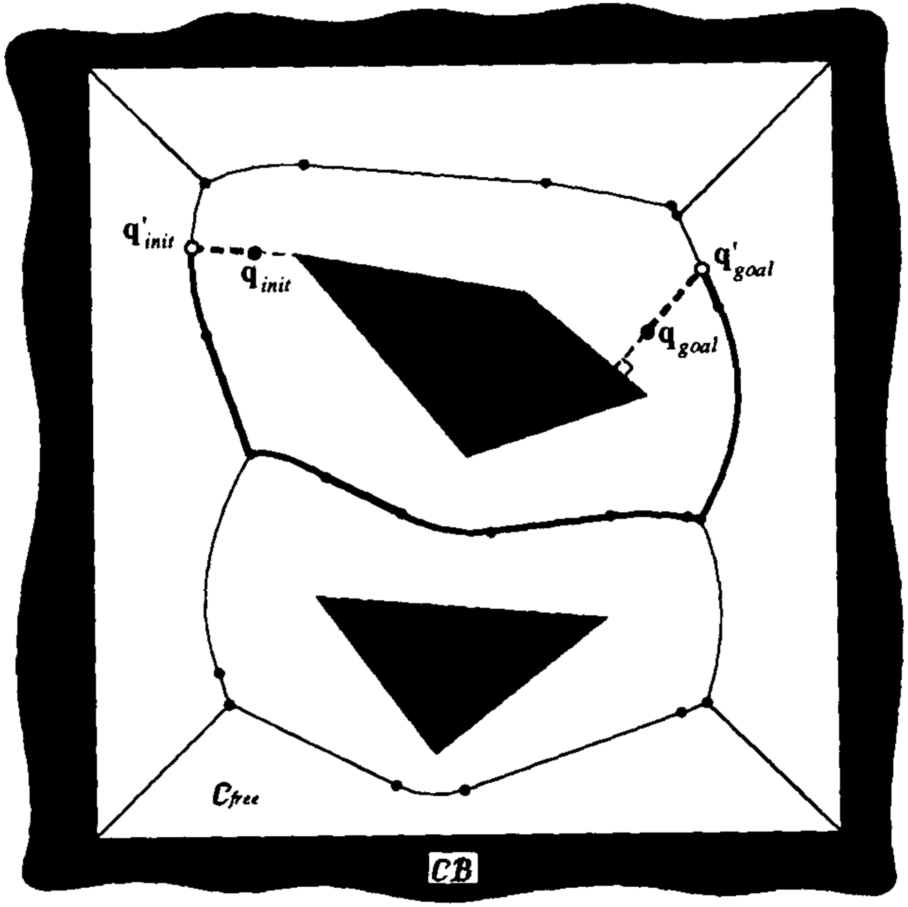
Again: we build a graph and search it to find a path!

Jean-Claude Latombe

So far:

- Cell decomposition methods:
  - exact
  - approximate
- General approach:
  - decompose C-space into simple cells
  - represent adjacency of cells as a graph
  - search graph to obtain path

Generalized Voronoi Diagram



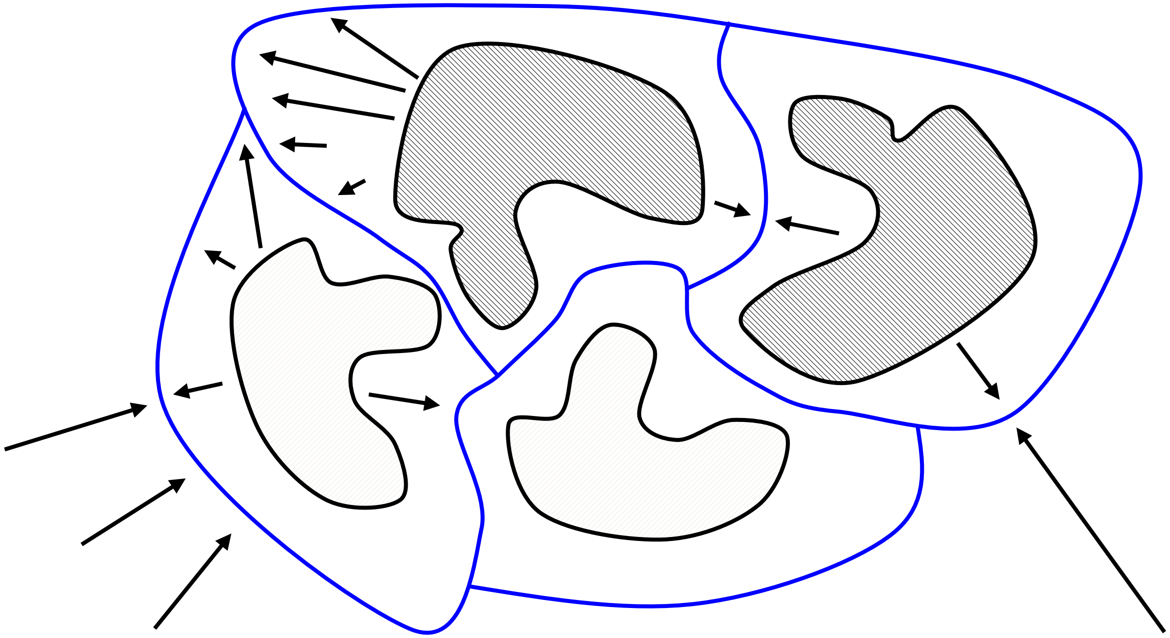
Jean-Claude Latombe

Completeness of Planning Algorithms

- A planner is **complete**, if it will find a path whenever one exists
- Resolution complete
  - Complete up to a given resolution
- Probabilistic Completeness
  - Complete with a certain probability
- Most global methods exhibit some form of completeness
- Local methods are incomplete

Retraction

A retraction is a continuous mapping of C_{free} onto a one-dimensional network of curves $R \subset C_{free}$.



Voronoi = Retraction

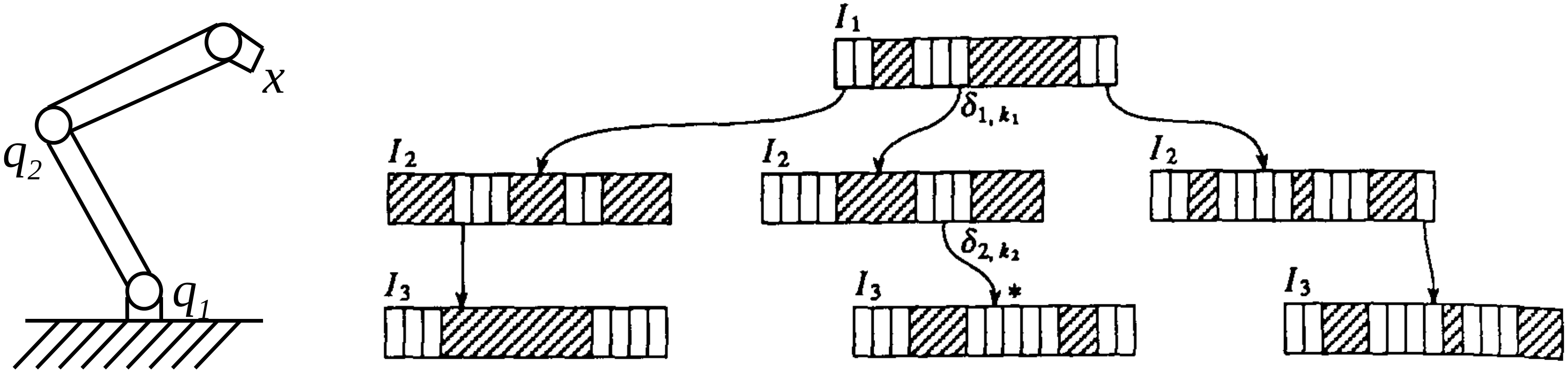
Four Categories

- Exact Cell Decomposition
- Approximate Cell Decomposition
- Roadmap Methods; examples:
  - Visibility
  - Voronoi
  - PRM
- Potential Field Methods

Global Motion Planning is Hard

- Most efficient *complete* and *general* algorithm so far:
  - Silhouette Method by Canny in 1988
  - Roadmap method
  - C-space with d dimensions
  - Exponential in d
- *PSPACE*-hardness has been shown for many simple problems:
  - Planar arm among polygonal obstacles
  - Rectangles in the plane

Approximate Cell Decomposition



Fortunately there are PRM Methods!

Jean-Claude Latombe

Summary: Basic Motion Planning

- Methods
  - Exact cell decomposition
  - Approximate cell decomposition
  - Roadmap
  - Potential Field
  - **PRM**
- Global / Complete / Expensive versus Local / Incomplete / Efficient

Comparison of Planning Methods