Probabilistic Roadmaps

Sung-Eui Yoon
(윤성의)

Course URL:
http://sglab.kaist.ac.kr/~sungeui/MPA

KAIST

Class Objectives

- **Understand probabilistic roadmap (PRM) approaches**
  - **Multi-query PRMs**
  - **Single-query PRMs**

Difficulty with Classic Approaches

- **Running time increases exponentially with the dimension of the configuration space**
  - **For a d-dimension grid with 10 grid points on each dimension, how many grid cells are there?**

$$10^d$$

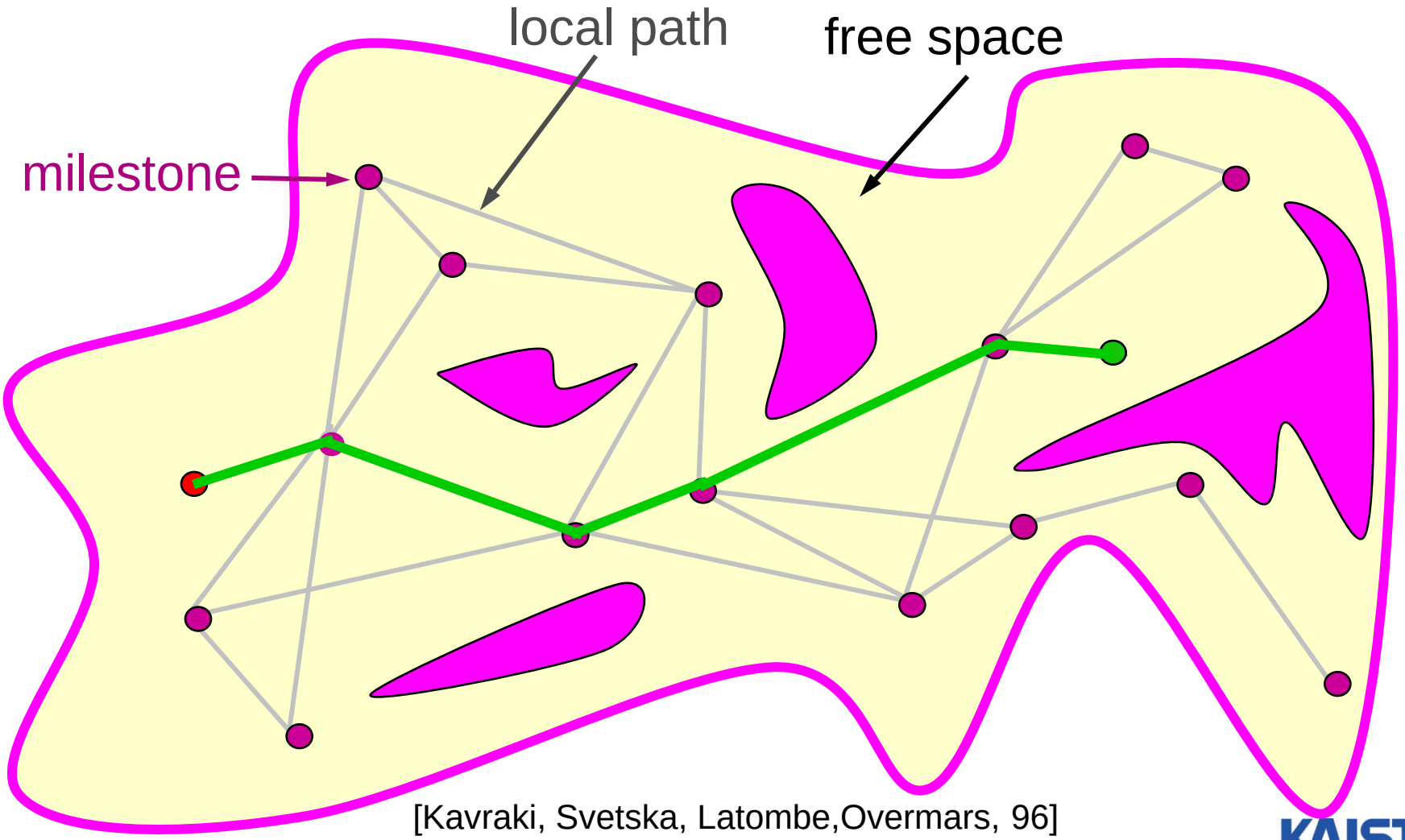
- **Several variants of the path planning problem have been proven to be PSPACE-hard**

Completeness

- **Complete algorithm → Slow**
  - **A complete algorithm finds a path if one exists and reports no otherwise**
  - **Example: Canny's roadmap method**
- **Heuristic algorithm → Unreliable**
  - **Example: potential field**

- **Probabilistic completeness**
  - **Intuition: If there is a solution path, the algorithm will find it with high probability**

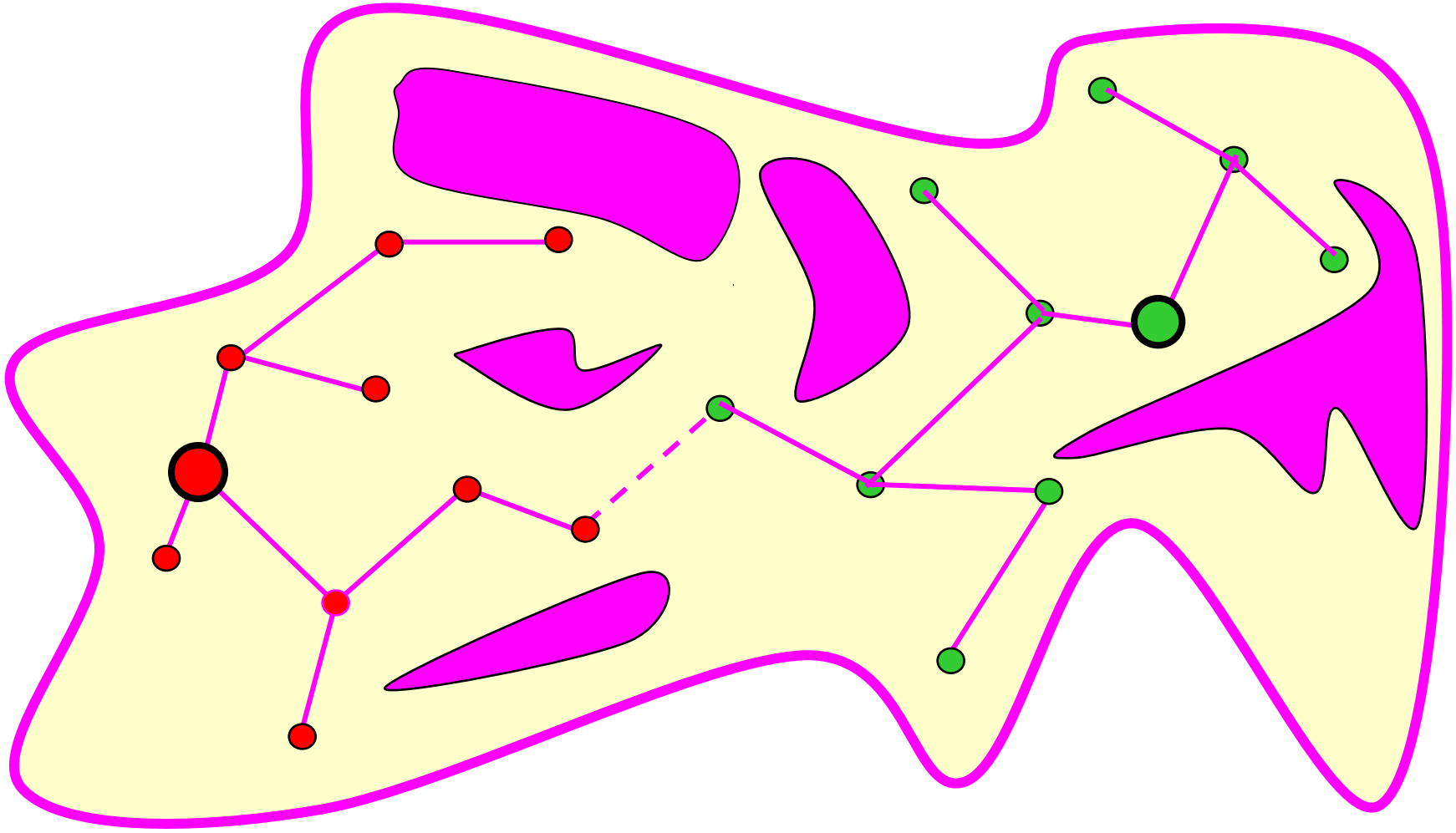
Probabilistic Roadmap (PRM): multiple queries



[Kavraki, Svetska, Latombe,Overmars, 96]

KAIST

Probabilistic Roadmap (PRM): single query



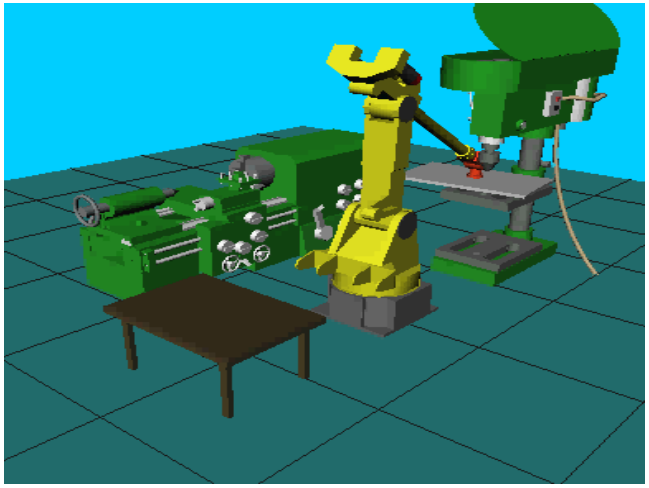
KAIST

Classic multiple-query PRM

- *Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,* **L. Kavraki** ***et al*****., 1996.**

Assumptions

- Static obstacles
- Many queries to be processed in the same environment
- Examples
  - Navigation in static virtual environments
  - Robot manipulator arm in a workcell



Overview

- **Precomputation: roadmap construction**
  - **Uniform sampling**
  - **Resampling (expansion)**
- **Query processing**

Uniform sampling

Input: geometry of the moving object & obstacles
Output: roadmap G = (V, E)

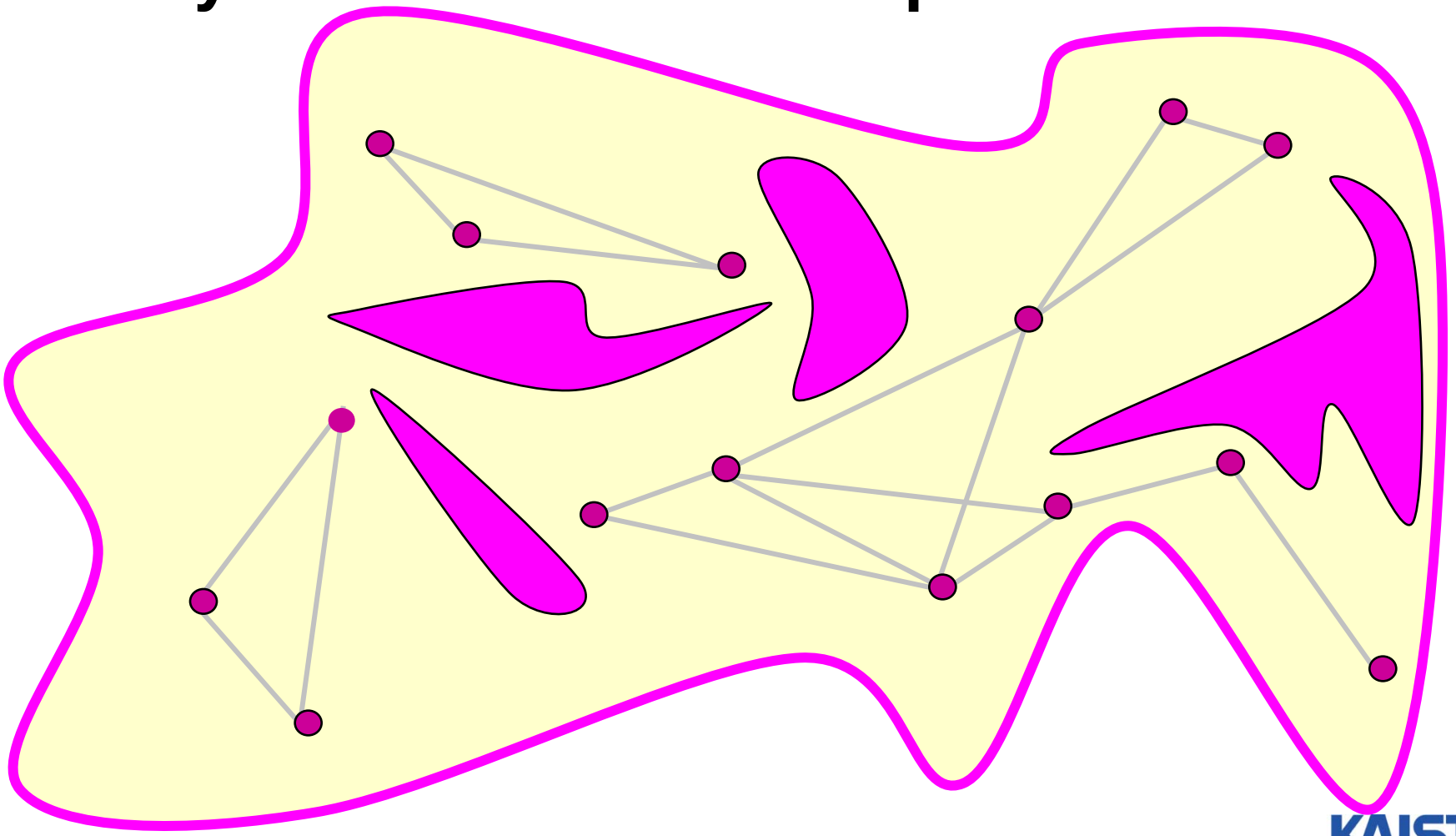
```
1: V ← ∅ and E ← ∅.
2:  repeat
3:   q ← a configuration sampled uniformly at random from C
4:    if CLEAR(q)then
5:      Add q to V.
6:      N_q ← a set of nodes in V that are close to q.
6:      for each q'∈ N_q, in order of increasing d(q,q')
7:        if LINK(q',q)then
8:          Add an edge between q and q' to E.
```

Some terminology

- The graph G is called a **probabilistic roadmap**.
- The nodes in G are called **milestones**.

Difficulty

- **Many small connected components**



Resampling (expansion)

- **Failure rate**

$$r(q) = \frac{\#.\,\text{failed LINK}}{\#.\,\text{LINK}}$$

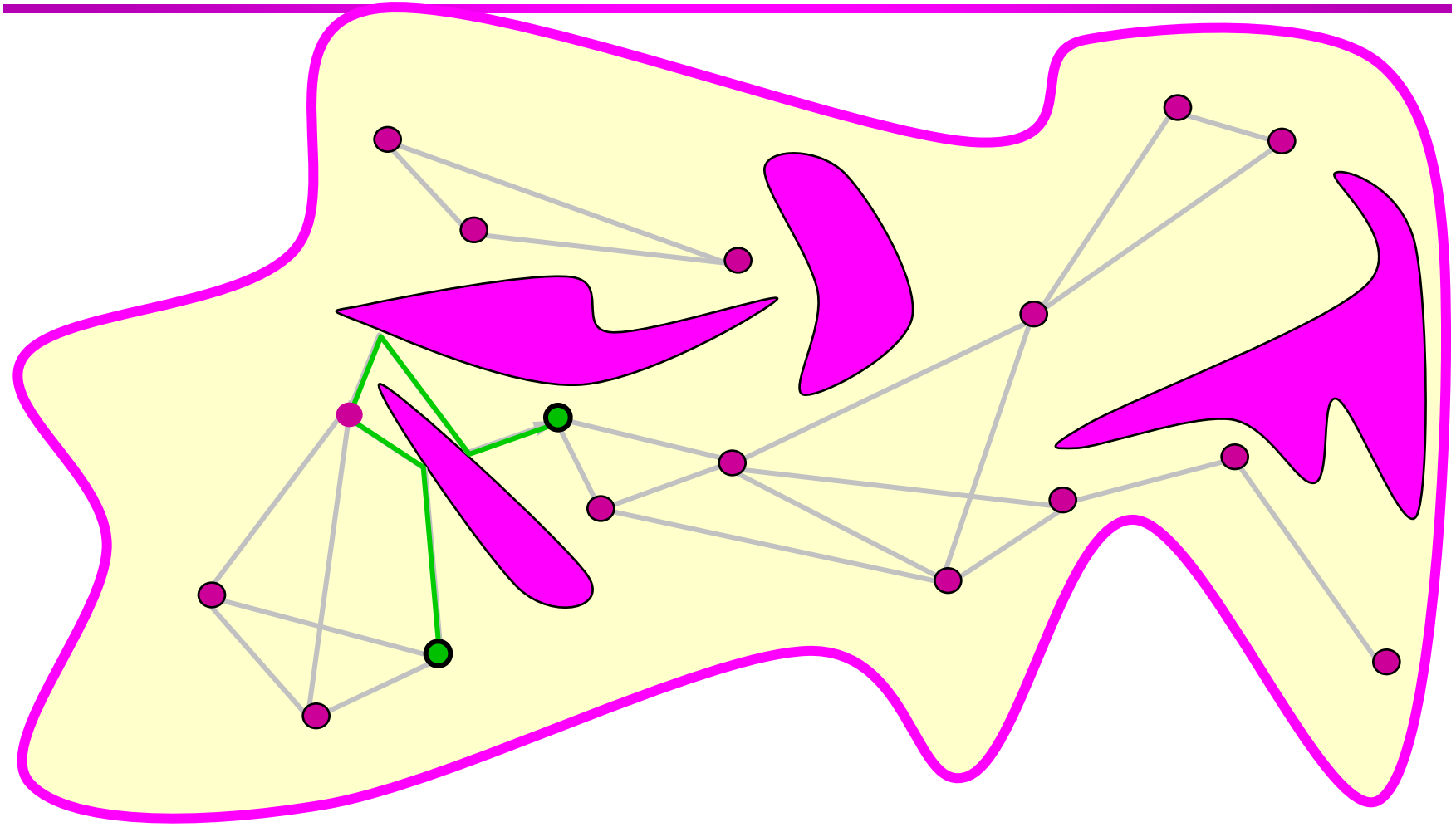
- **Normalized weight**

$$w(q) = \frac{r(q)}{\sum_p r(p)}$$

- **Resampling probability**

$$\Pr(q) = w(q)$$

Resampling (expansion)



KAIST

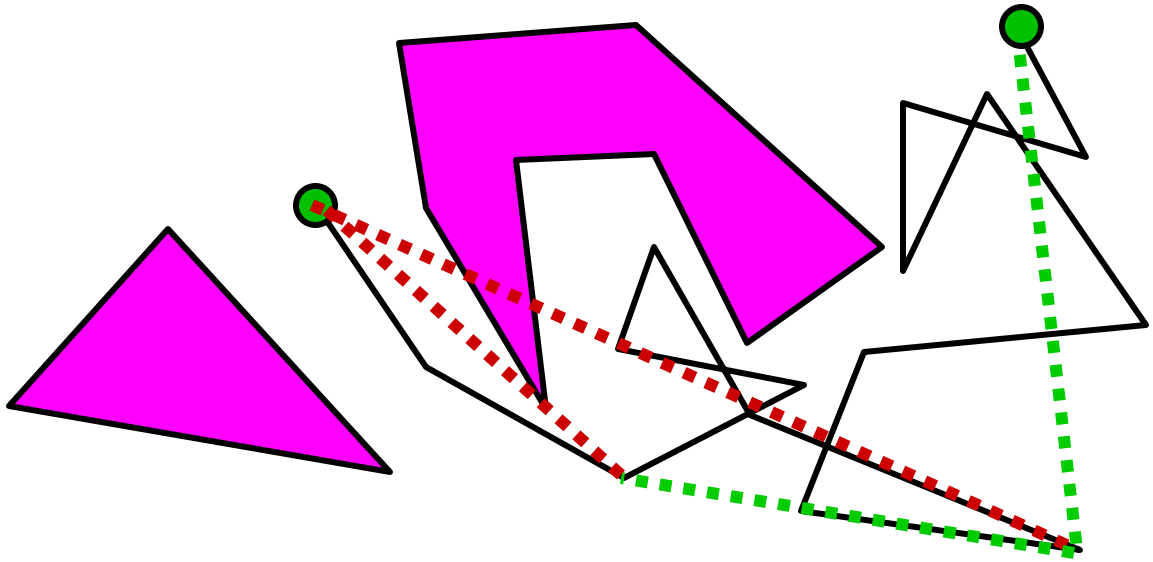
Query processing

- Connect q_{init} and q_{goal} to the roadmap
- Start at q_{init} and q_{goal}, perform a random walk, and try to connect with one of the milestones nearby
- Try multiple times

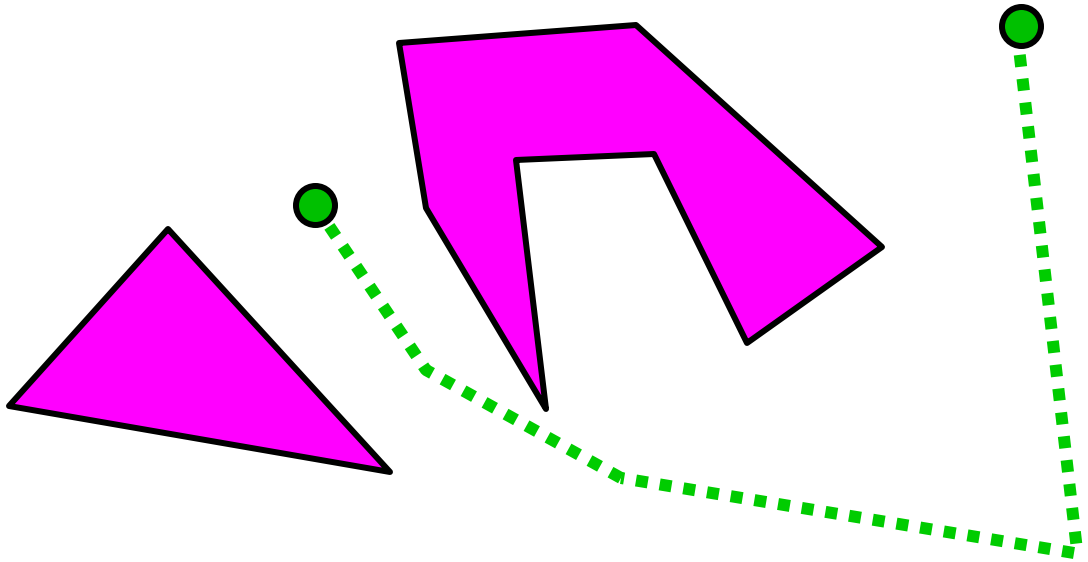
Error

- **If a path is returned, the answer is always correct.**
- **If no path is found, the answer may or may not be correct. We hope it is correct with high probability.**

Smoothing the path



Smoothing the path



Summary

- **What probability distribution should be used for sampling milestones?**
- **How should milestones be connected?**
- **A path generated by a randomized algorithm is usually jerky. How can a path be smoothed?**

Sing-Query PRM

- *Path Planning Using Lazy PRM*, **R. Bohlin & L. Kavraki, 2000.**

Precomputation: roadmap construction

- **Nodes**
  - **Randomly chosen configurations, which may or may not be collision-free**
  - **No call to CLEAR**
- **Edges**
  - **an edge between two nodes if the corresponding configurations are close according to a suitable metric**
  - **no call to LINK**

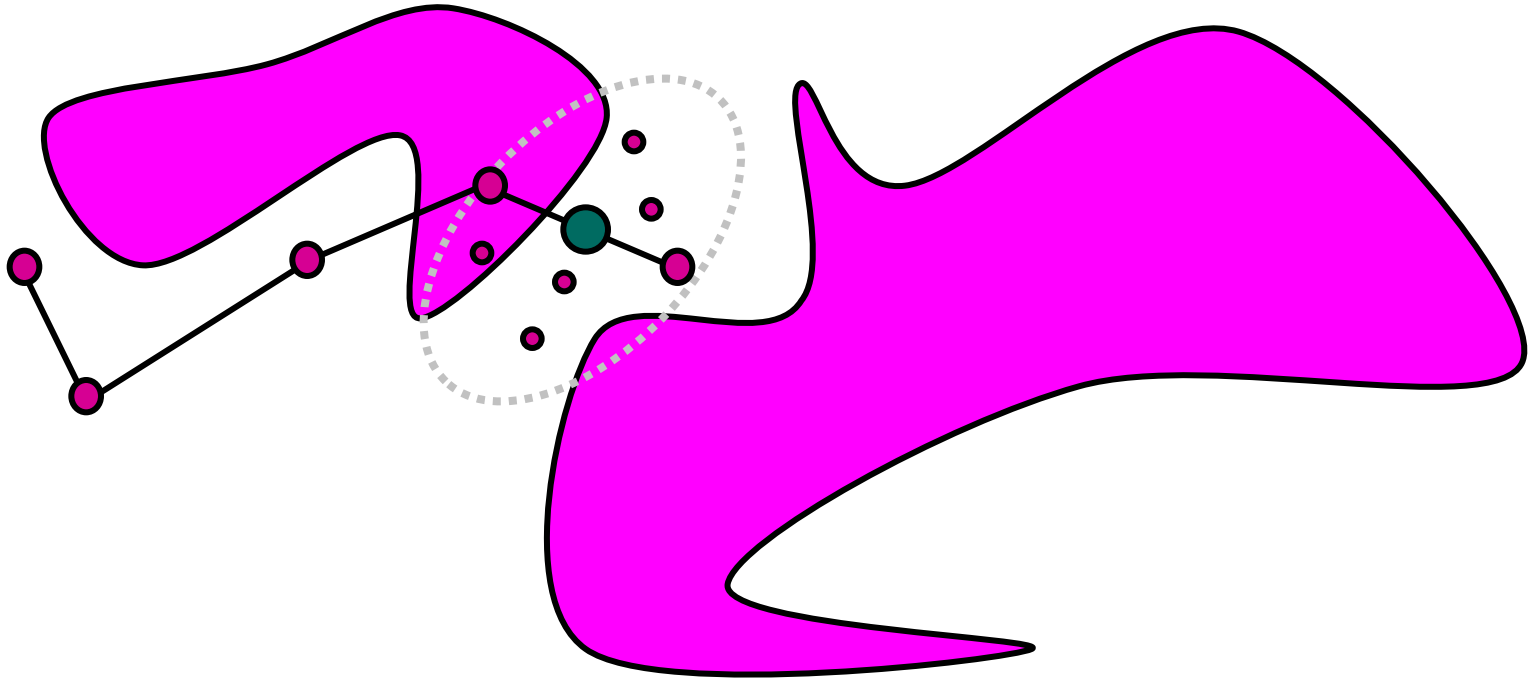
Query processing: overview

1. **Find a shortest path in the roadmap**
2. **Check whether the nodes and edges in the path are in collision-free regions.**
3. **If yes, then done. Otherwise, remove the nodes or edges in violation. Go to (1).**

We either find a collision-free path, or exhaust all paths in the roadmap and declare failure.

Node enhancement

- **Select nodes that close the boundary of F**



Class Objectives were:

- **Understand probabilistic roadmap (PRM) approaches**
  - **Multi-query PRMs**
  - **Single-query PRMs**