CS 460/560
Introduction to Computational Robotics
Fall 2019, Rutgers University

Lecture 13
Intro To Sampling-Based Planning Methods

Instructor: Jingjin Yu

Outline (for Next 3 Lectures)

Drawbacks of combinatorial motion planning methods

Probabilistic roadmap (PRM): an introduction

Components of sampling-based motion planning methods

- ⇨ Sampling
- ⇨ k-d tree and nearest neighbor search
- ⇨ Distance metric
- ⇨ Collision detection

PRM in more detail

A new notion of completeness

Rapidly-exploring random trees (RRT)

When would sampling-based method work well?

Optimality issues

Drawbacks of Combinatorial Methods

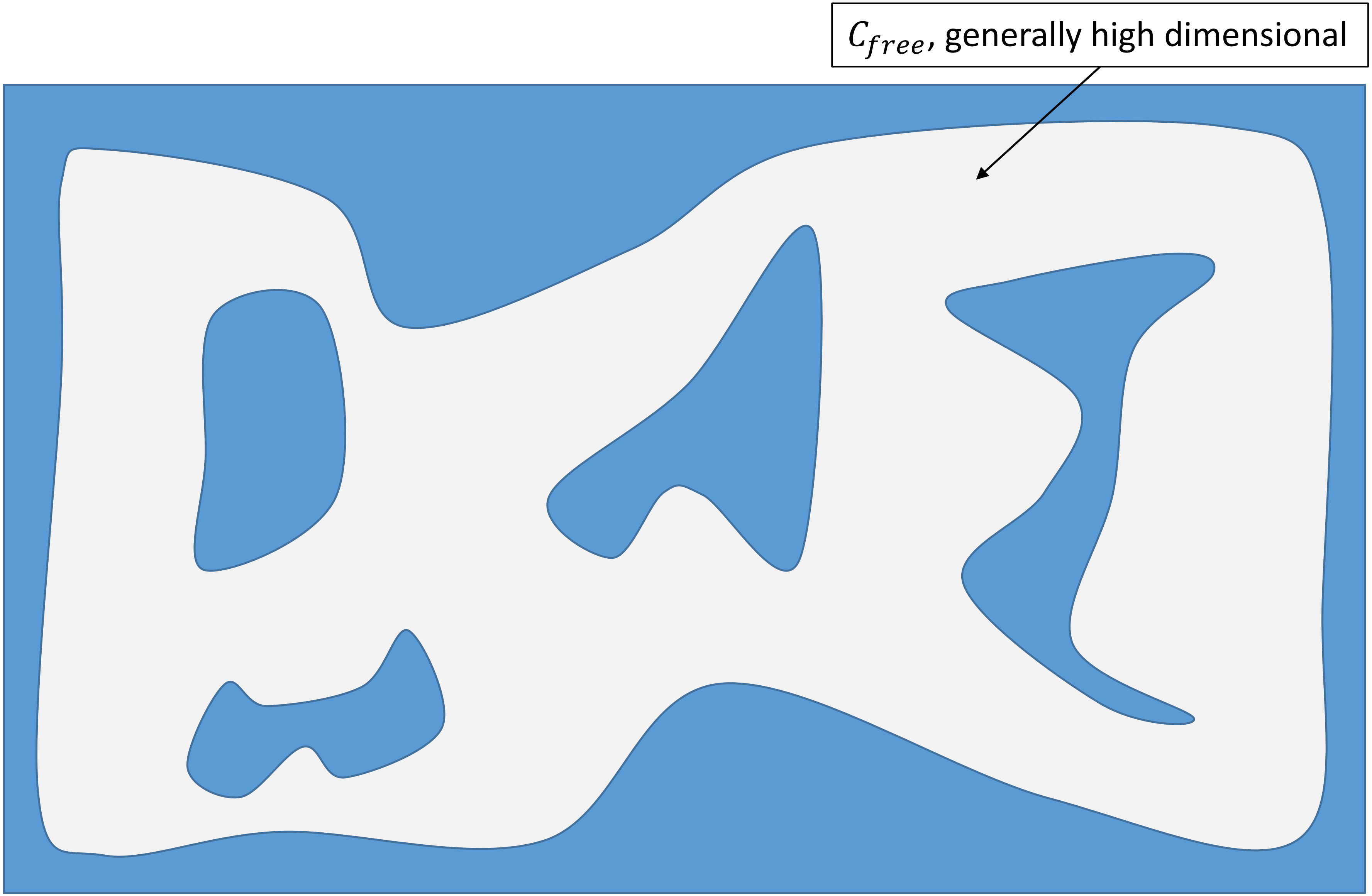
Recall the illustration of the Piano Mover's Problem

- ⇨Modeling of the free configuration space C_{free} can be a daunting task – they have to be represented as semi-algebraic sets
- ⇨The associated computation is also prohibitively expensive for even just a few degrees of freedom

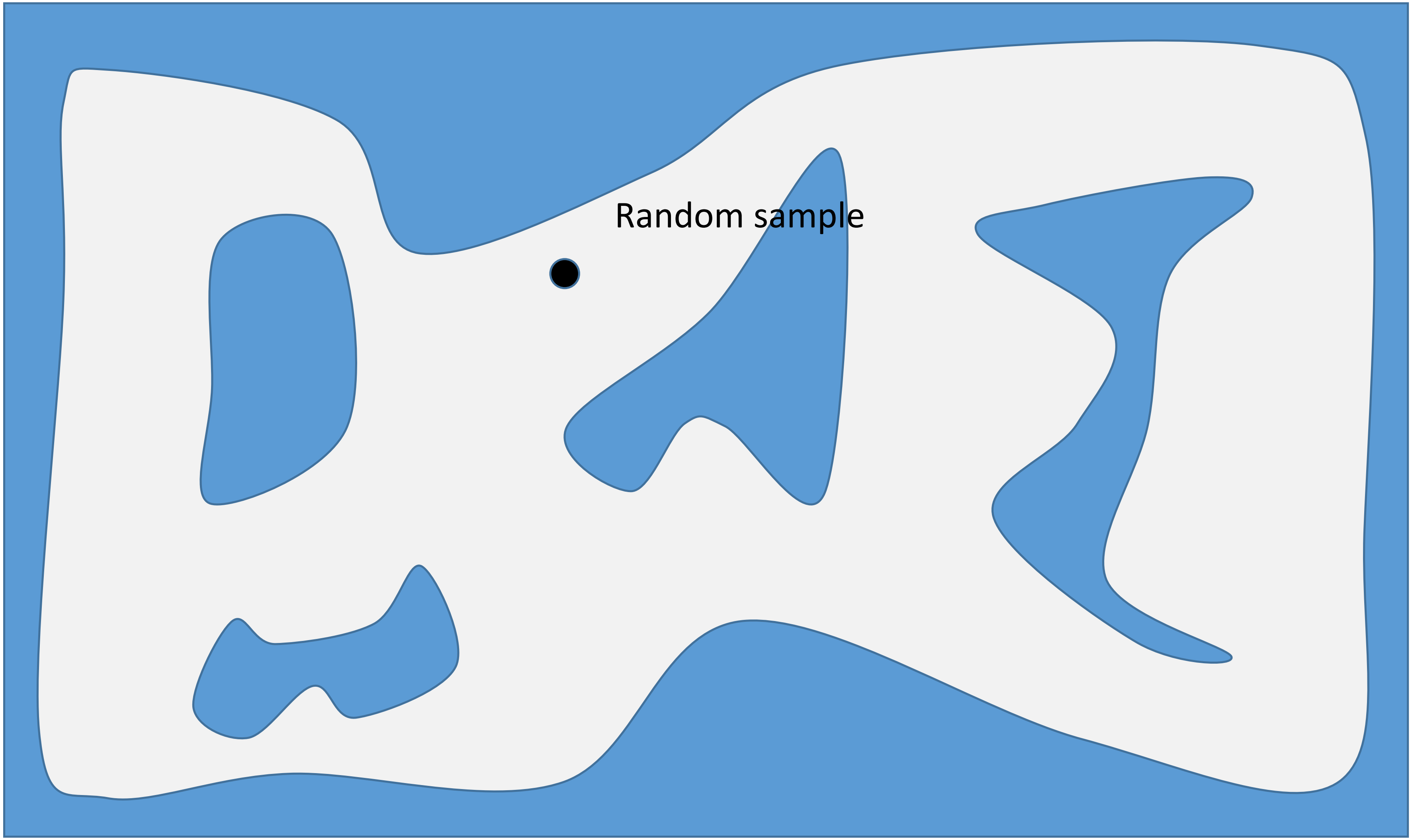
To the rescue: **sampling-based methods** – instead of representing C_{free} explicitly and globally, we instead "probe" the space locally, as necessary

Image source: Planning Algorithms by LaValle

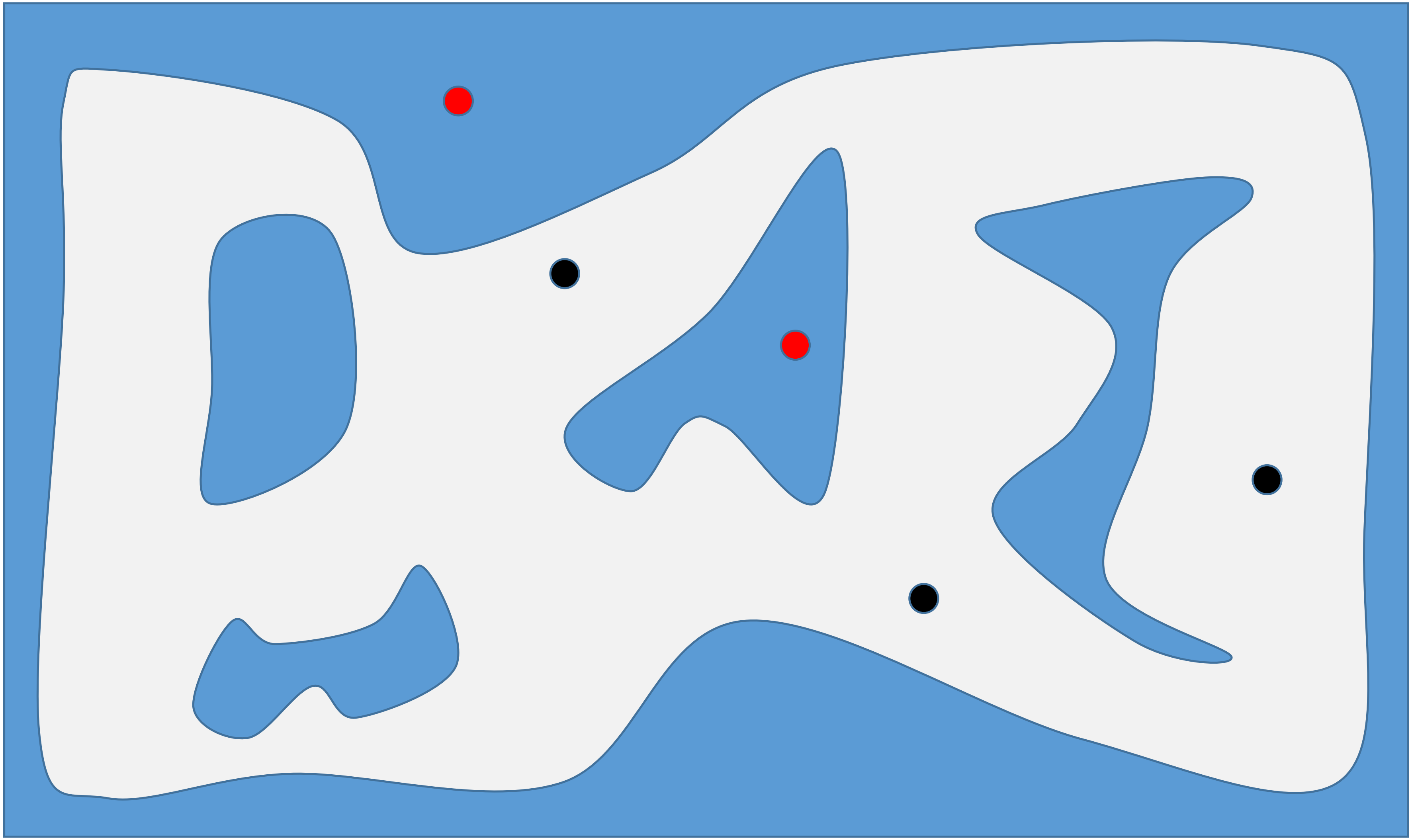
The Workings of Probabilistic Roadmap



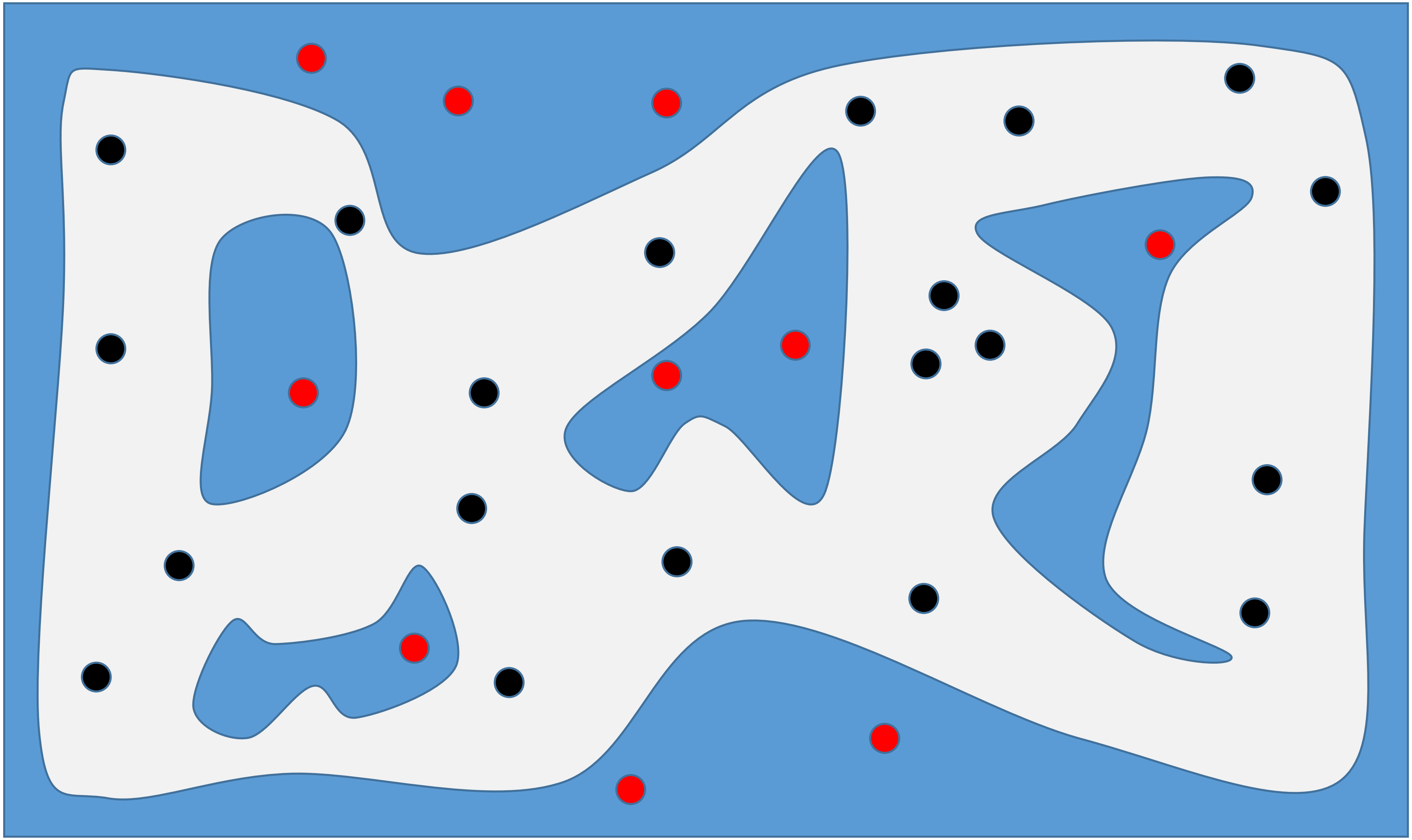
Generating Random Samples



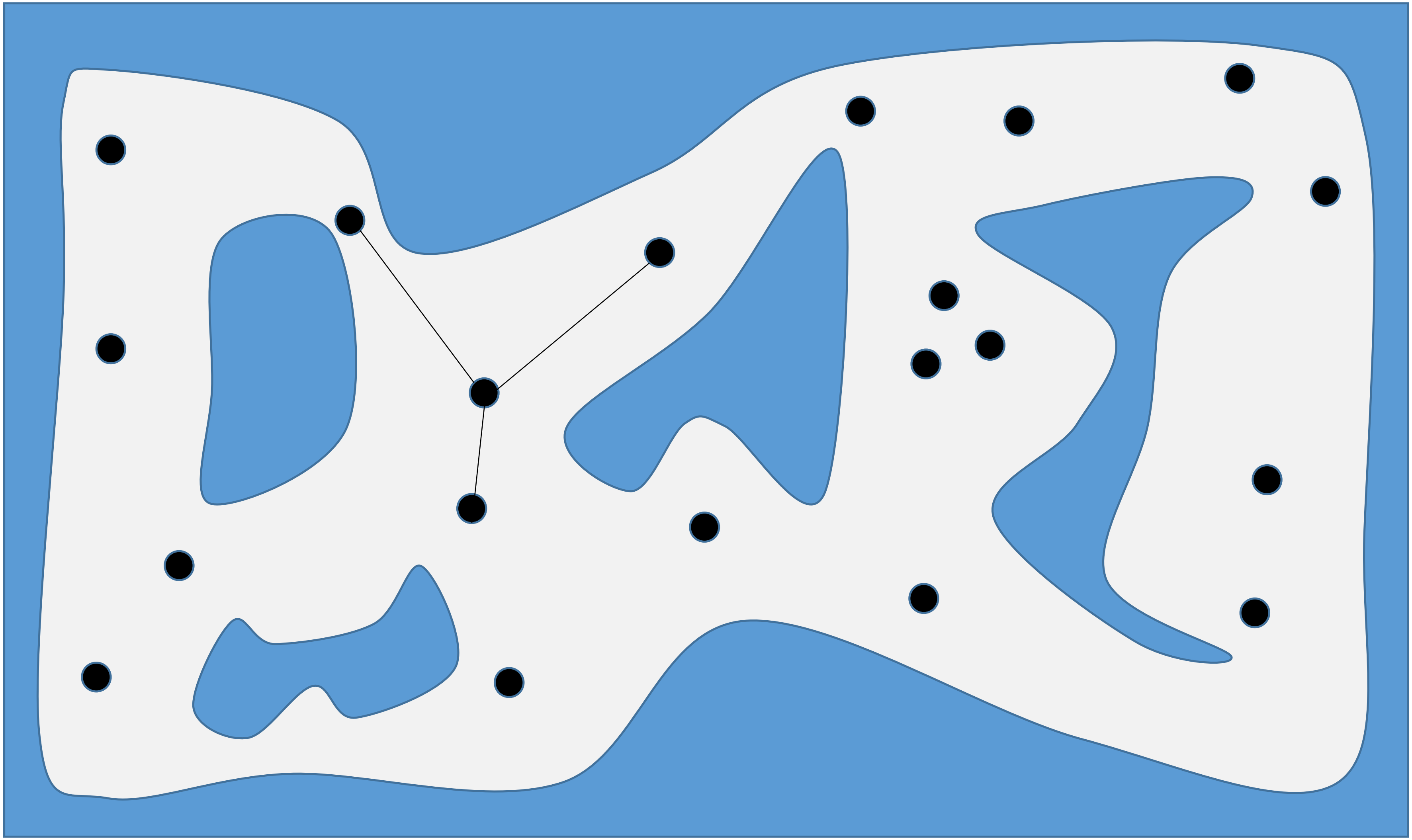
Rejecting Samples Outside C_{free}



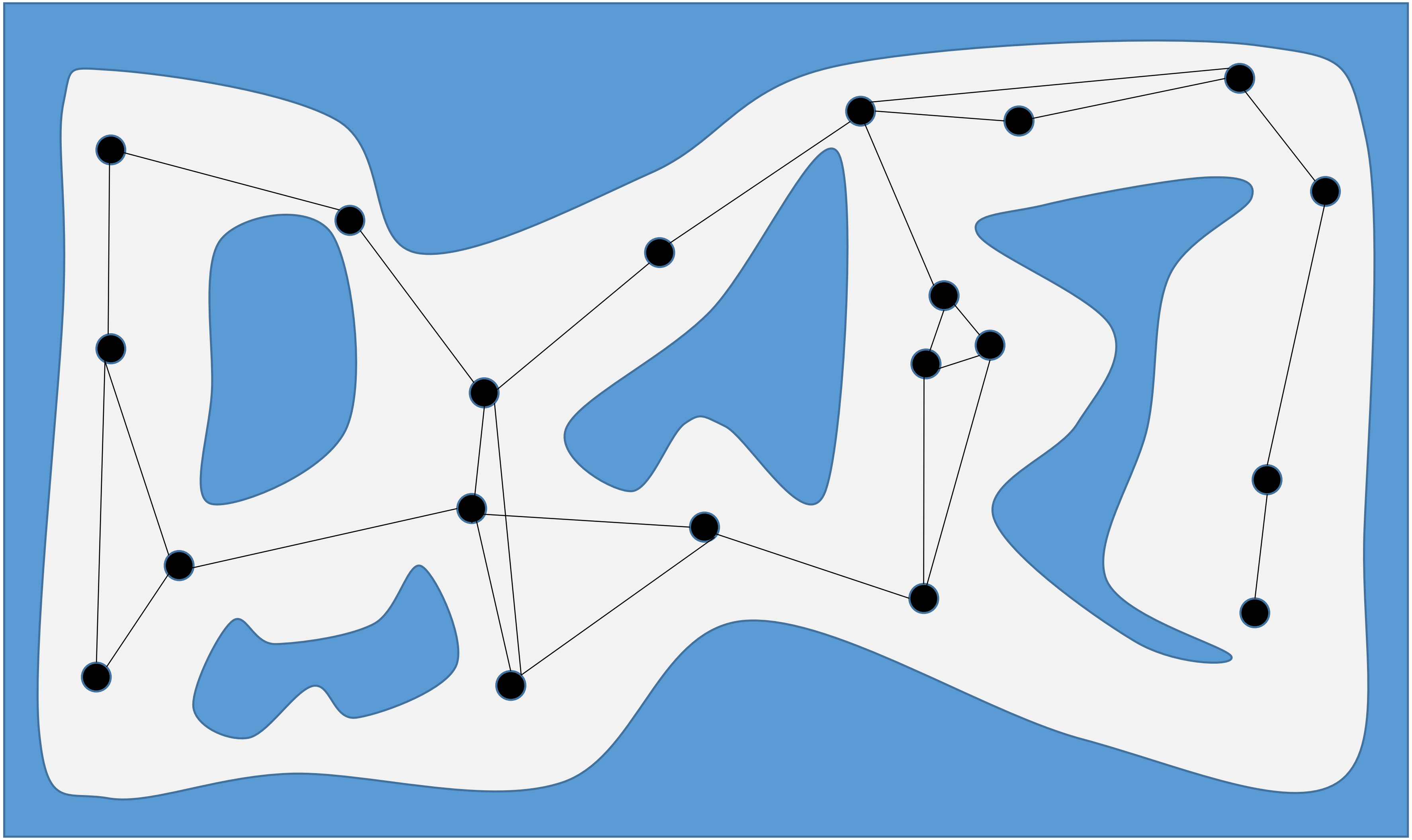
Collecting Enough Samples in C_{free}



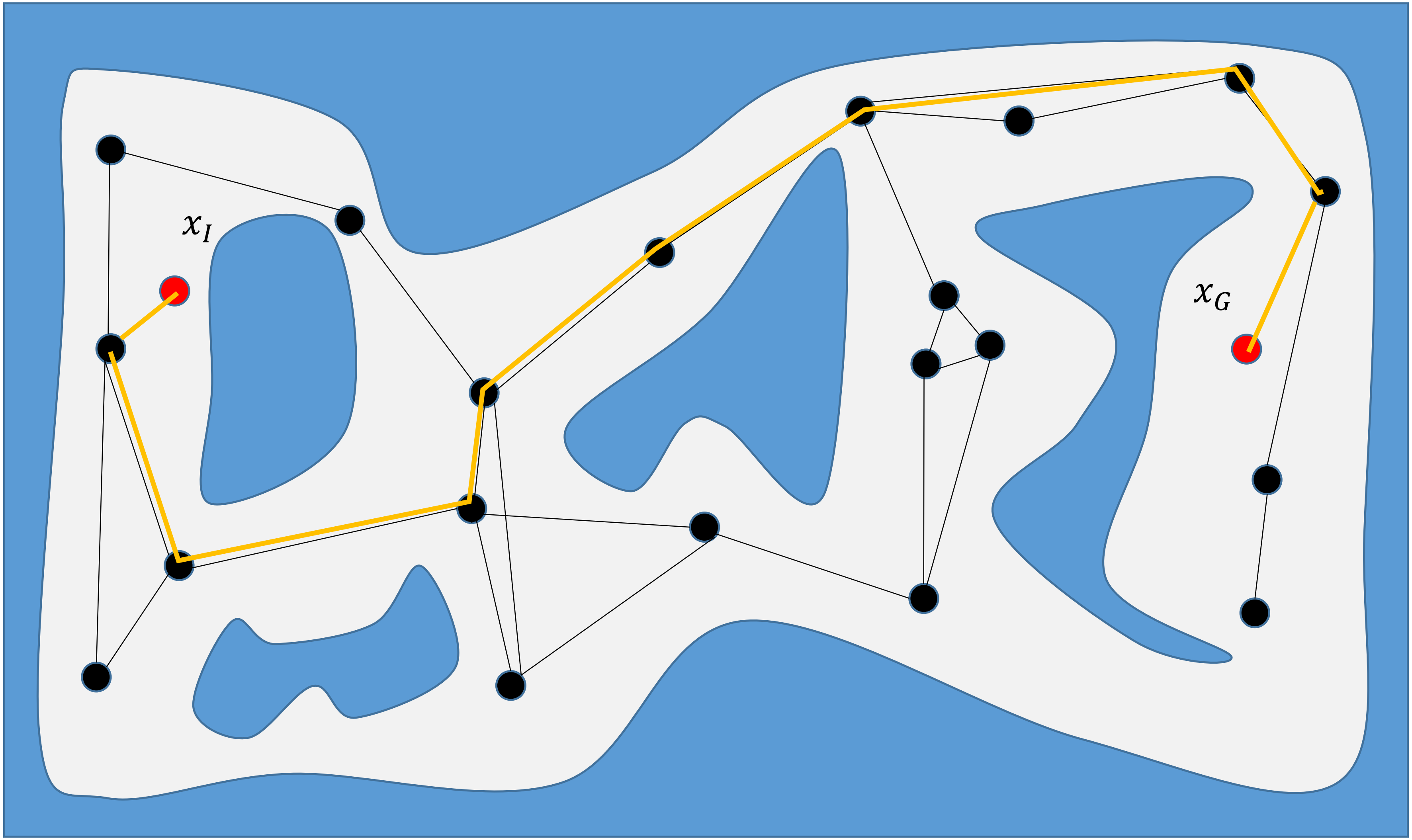
Connect to Nearest Neighbors (If Possible)



Connect to Nearest Neighbors (If Possible)



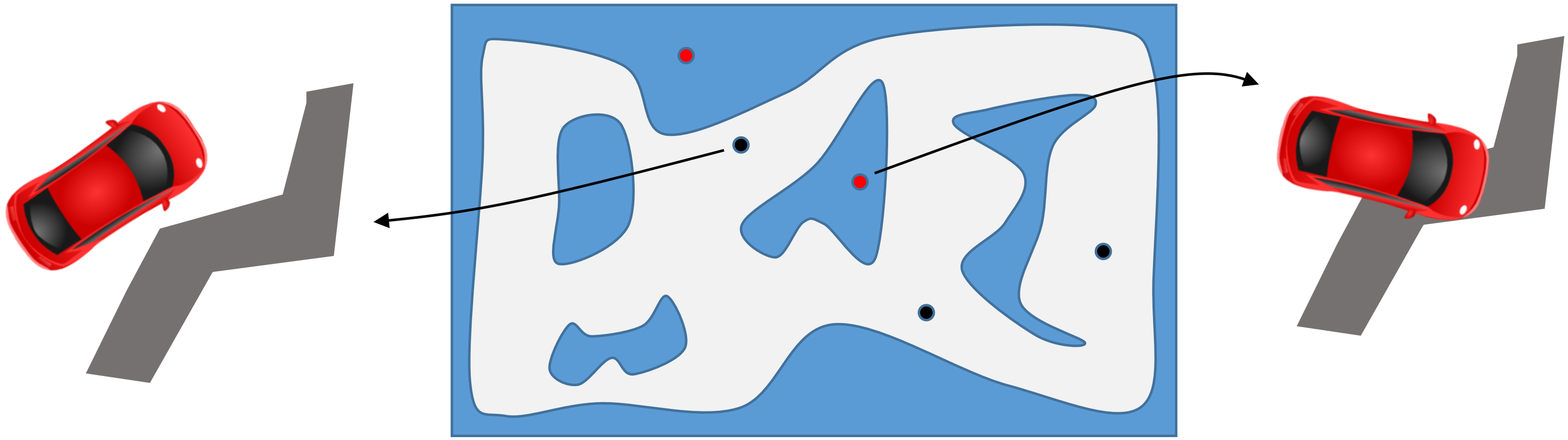
Query Phase



Key Components of Sampling-Based Planning

Sampling-based planning requires several important subroutines

- ⇨An **efficient sampling routine** is needed to generate the samples. These samples should **cover** C_{free} well in order to be effective
- ⇨**Efficient nearest neighbor search** is necessary for quickly building the roadmap: for each sample in C_{free} we must find its k-nearest neighbors
- ⇨The neighbor search also requires a **distance metric** to be properly defined so we know the distance between two samples
  - ⇨ This can be tricky for certain spaces, e.g., $SE(3)$
- ⇨**Collision checking** - Note that C_{free} is not computed explicitly so we actually are checking collisions between a complex robot and a complex environment



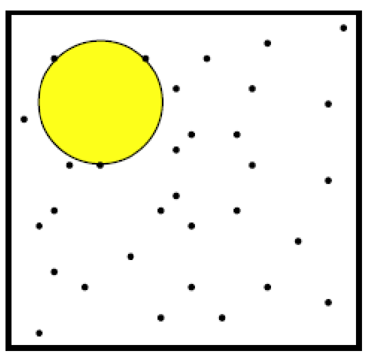
“Goodness” of Samples

The sampling process aims at “covering” C_{free}. How to measure the “goodness” of a set of samples?

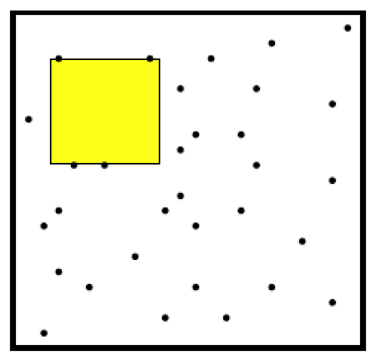
Dispersion: the dispersion of a finite set P of samples in a metric space (X, ρ) is

$$\delta(P) = \sup_{x \in X}\{\min_{p \in P}\{\rho(x, p)\}\}$$

Roughly, this means the largest ball that can be fit in the samples without including any sample inside the ball



(a) L_2 dispersion



(b) L_∞ dispersion

Generally speaking, given $|P|$ samples, a sample set with smaller dispersion $\delta(P)$ is better.

Image source: Planning Algorithms by LaValle

Computing Dispersion in 2D

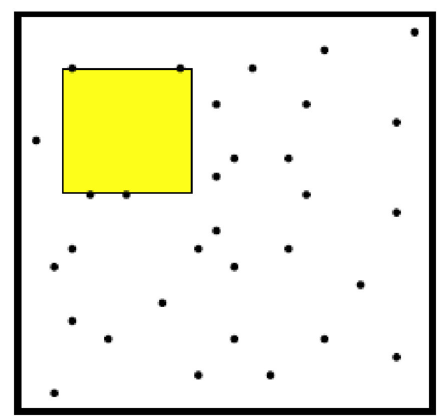
Dispersion of a set of points can be difficult to compute in general

2D cases are somewhat simpler, e.g., L_∞ dispersion

- ⇨ This is to find the largest square that does not contain any point inside
- ⇨ How to compute?

A simple (not very efficient) method (for your homework)

- ⇨ For each point, assume its on the top/left side of a rectangle
- ⇨ Then iteratively shrink the rectangle: for every other point
  - ⇨ If the point is outside the current candidate rectangle, do nothing
  - ⇨ If the point is inside, shrink the rectangle
    - ⇨ A few cases here to handle
- ⇨ Yields a "final" rectangle
- ⇨ Take the largest square that can fit in
- ⇨ Do this for all points and pick the largest
- ⇨ Complexity: $O(n^2)$



(b) L_∞ dispersion

Image source: Planning Algorithms by LaValle

Uniformly Random Sampling Strategy

The simplest way of achieving this is through (probabilistic) **uniformly random samples**

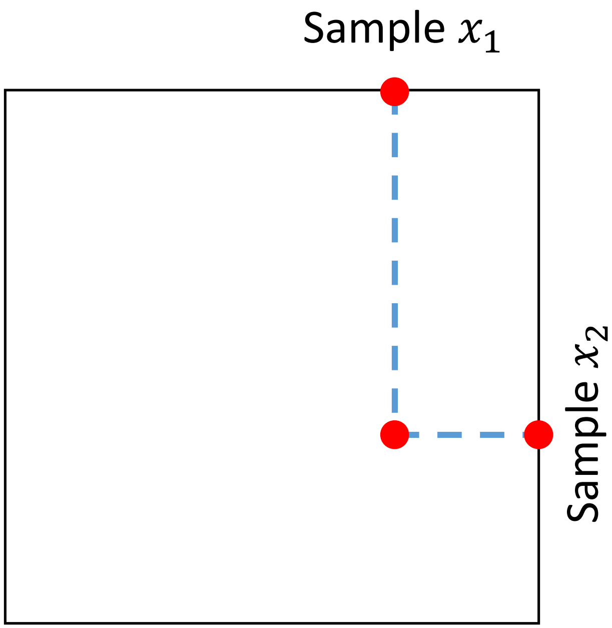
Examples in $\mathbb{R}^n$

⇨1D uniform sampling

⇨Through your favorite random function

⇨Uniform sampling in the unit square

⇨Readily generalizes to high dimensions



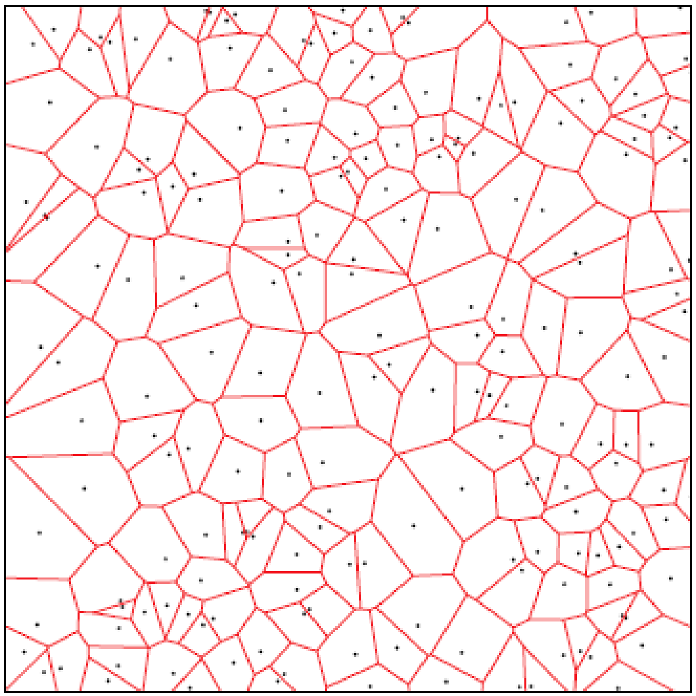
Similar for other spaces, e.g.,$\mathbb{R}^2 \times S^1$

⇨Simply sample each dimension individually

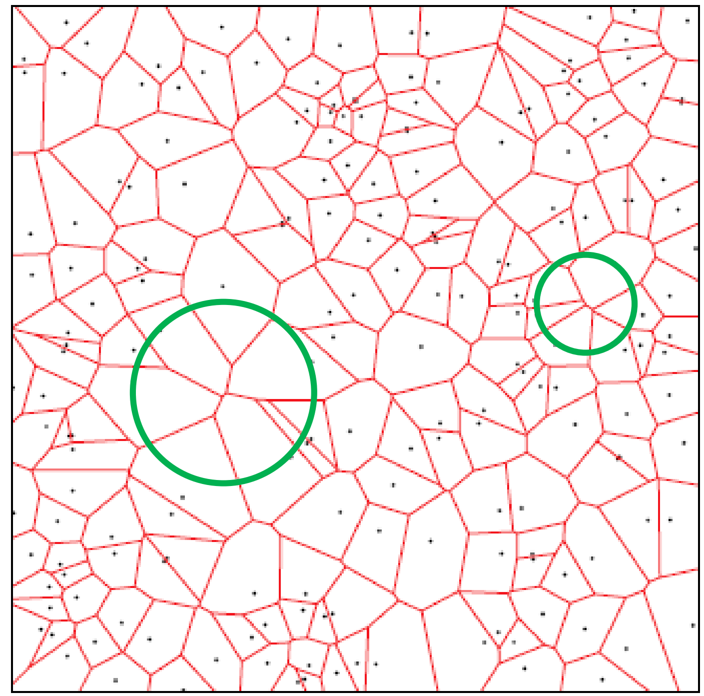
⇨Then put together the individual dimensions

Uniformly Random Sampling Strategy

Generally, uniform sampling works quite well



(a) 196 pseudorandom samples



(b) 196 pseudorandom samples

But it is not the best for dispersion: uniformly random samples require an additional $O(\log n)$ factor to achieve the same $\delta(P)$

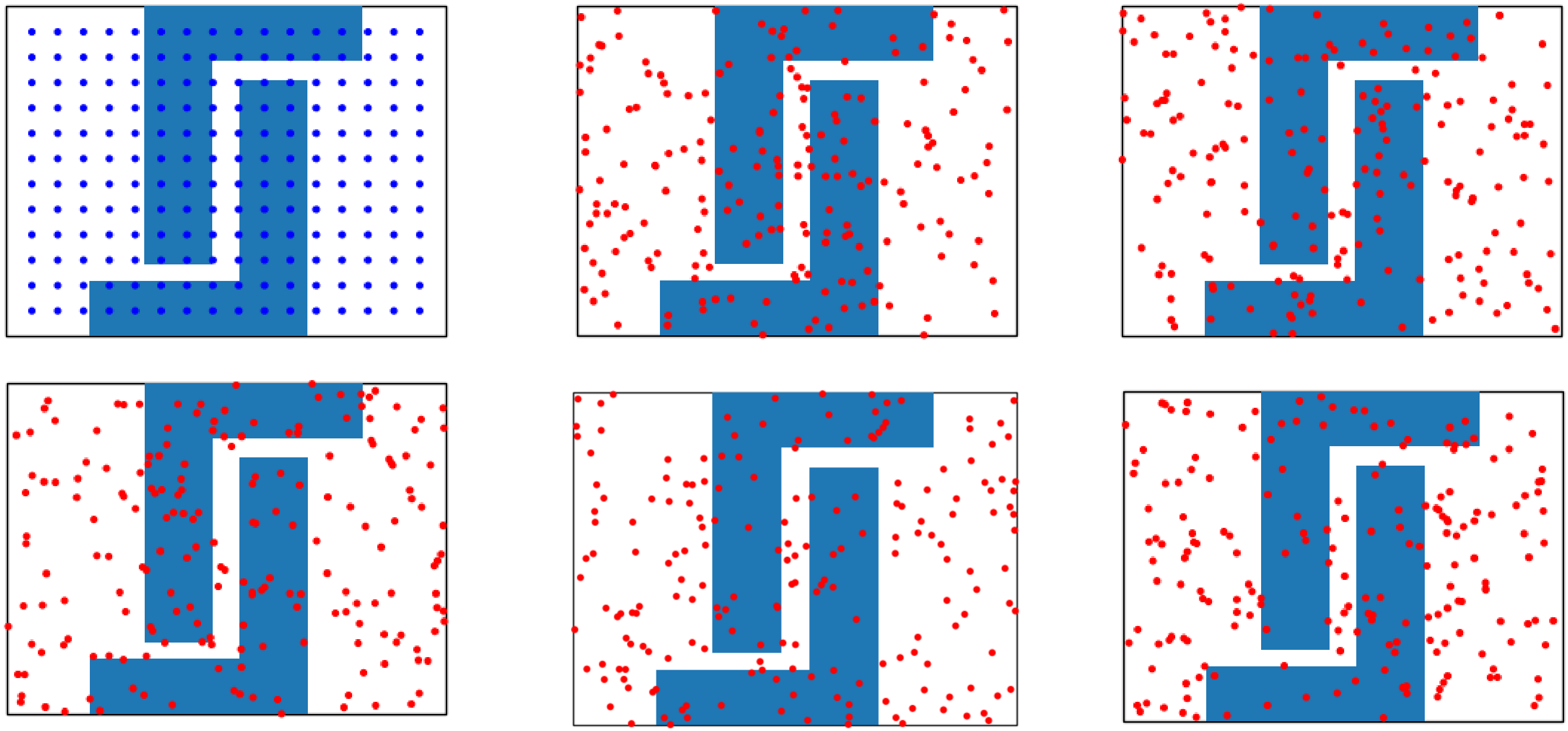
Image source: Planning Algorithms by LaValle

Deterministic Sampling Strategy

What if we simply use lattices, e.g., deterministic samples?

⇨Requires fewer samples (by a factor of $O(\log n)$) to reach the same dispersion

⇨But for any deterministic sampling strategy, there are environments that work against the strategy



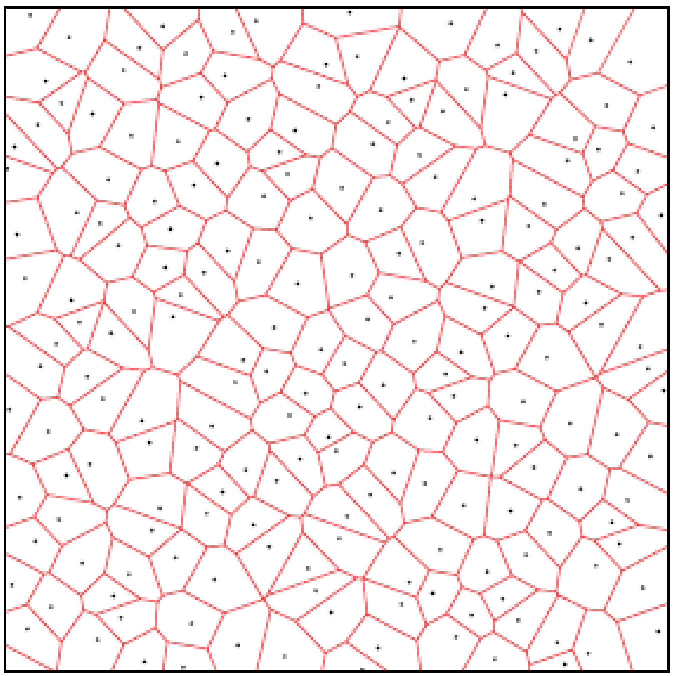
192 samples

Deterministic Sampling Strategy, Continued

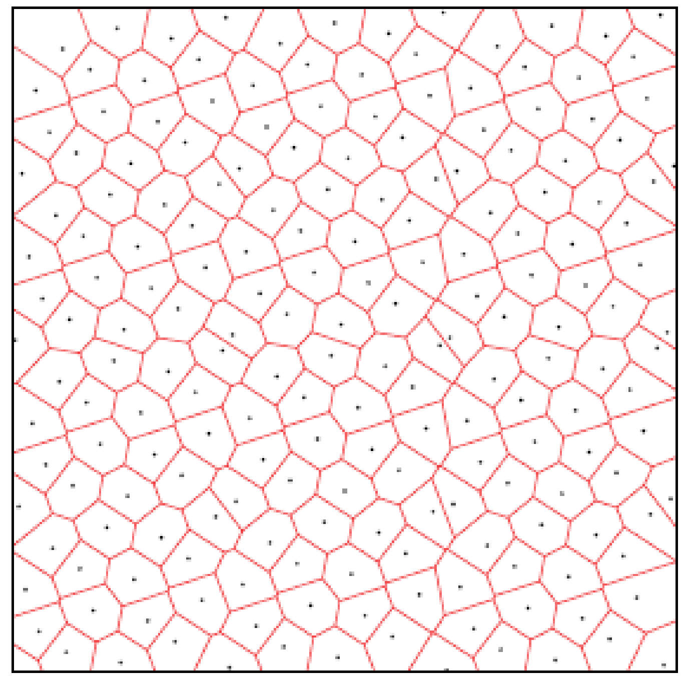
Good deterministic sequences can be designed, e.g.,

⇨Halton

⇨Hammersley



(a) 196 Halton points



(b) 196 Hammersley points

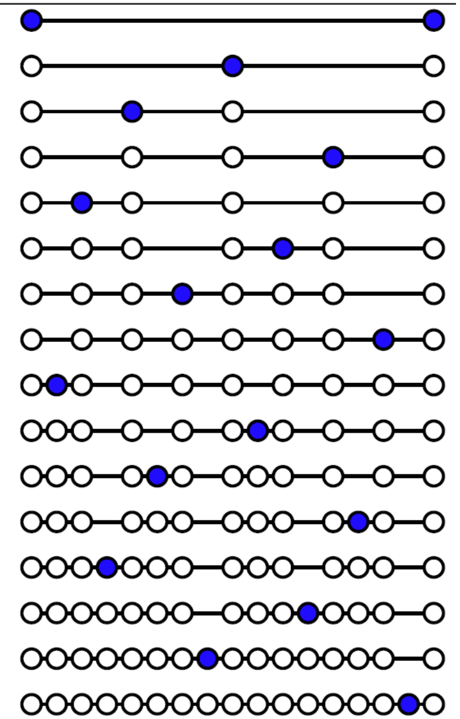
⇨Many of these sequences are also **incremental**

Image source: Planning Algorithms by LaValle

Halton Sequence in 1D

In 1D, also known as the van de Corput sequence

⇨Use a trick called reverse binary representation

i	Naive Sequence	Binary	Reverse Binary	Van der Corput	Points in $[0,1]/\sim$
1	0	.0000	.0000	0	
2	1/16	.0001	.1000	1/2	
3	1/8	.0010	.0100	1/4	
4	3/16	.0011	.1100	3/4	
5	1/4	.0100	.0010	1/8	
6	5/16	.0101	.1010	5/8	
7	3/8	.0110	.0110	3/8	
8	7/16	.0111	.1110	7/8	
9	1/2	.1000	.0001	1/16	
10	9/16	.1001	.1001	9/16	
11	5/8	.1010	.0101	5/16	
12	11/16	.1011	.1101	13/16	
13	3/4	.1100	.0011	3/16	
14	13/16	.1101	.1011	11/16	
15	7/8	.1110	.0111	7/16	
16	15/16	.1111	.1111	15/16	

⇨Very important: the sequence is incrementally created

⇨Not necessary to know how many samples are needed beforehand

Image source: Planning Algorithms by LaValle

Halton Sequence in 2D

In 2D, we add another coordinate

- ⇨For x coordinate, use reversed binary
- ⇨For y coordinate, use reversed ternary
- ⇨First few iterations

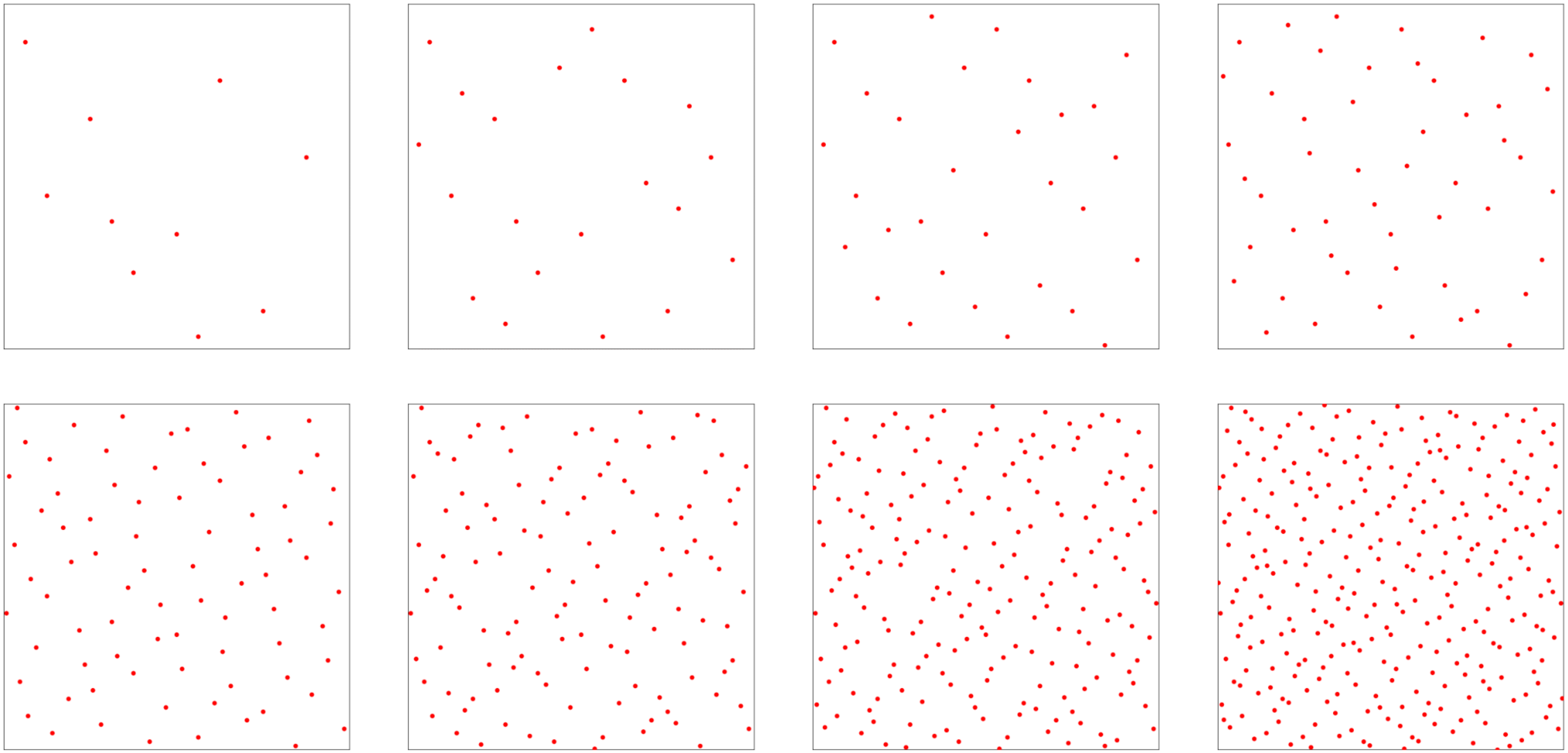
Number	Base 2 (x-coord)	Base 2 inverted	Decimal	Base 3 (y-coord)	Base 3 inverted	Decimal
1	1	0.1	0.5	1	0.1	0.33
2	10	0.01	0.25	2	0.2	0.67
3	11	0.11	0.75	10	0.01	0.11
4	100	0.001	0.125	11	0.11	0.44
5	101	0.101	0.625	12	0.21	0.78
6	110	0.011	0.375	20	0.02	0.22
7	111	0.111	0.875	21	0.12	0.55
8	1000	0.0001	0.0625	22	0.22	0.89
9	1001	0.1001	0.5625	100	0.001	0.04

Halton Sequence in 2D - Visualization

Halton samples using base 2 and 3 for x and y coordinates

⇨10, 20, 30, 50, 80, 120, 180, 250 samples

⇨Incremental and non-grid-like



Additional Aspects on Sampling Strategies

Dispersion is just one way to measure sample “goodness”

- ⇨Other measures exist, e.g., **discrepancy**

$$D_N(P) = \sup_B \left| \frac{|B \cap P|}{N} - \mu(B) \right|$$

- ⇨P: point set, $N = |P|$, B: a ball, $\mu(B)$: volume of B
- ⇨Also very hard to compute…

⇨Additional matters

- ⇨Uniform samples may not be the best with obstacles
- ⇨In the end, we want to construct a graph; edges are also important