

Sampling-Based Motion Planning: Multi-Query Planners

CSCI 545 Introduction to Robotics
Instructor: Stefanos Nikolaidis

Motion Planning



Motion Planning

Automatic Feeding with Assistive Robot Arm

Laura V. Herlant
The Robotics Institute
Carnegie Mellon University

Siddhartha S. Srinivasa
University of Washington

Taxonomy of Motion Planning Problems

- Plans: Discrete / Continuous / Hybrid
- Environment: Immovable / Moving / Movable

Taxonomy of Motion Planning Problems

- Plans: Discrete / Continuous / Hybrid
- Environment: Immovable / Moving / Movable

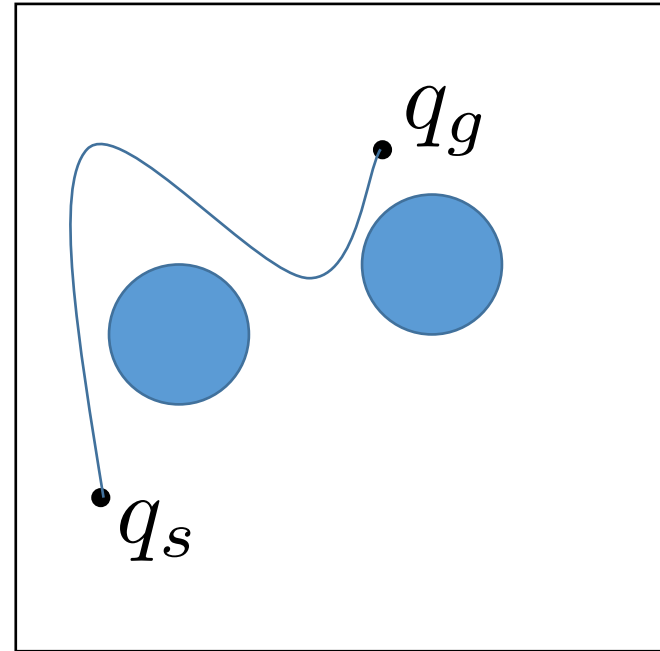
Examples?

Taxonomy of Motion Planning Problems

- Plans: Discrete / Continuous / Hybrid
- Environment: Immovable (Walls) / Moving (People) / Movable (Bottle)
- Interaction: Non-contact (avoid) / Contact (Grasping / Pushing)
- Uncertainty: None / Known / Bounded / Unknown
 x $p(x)$ $x \in X$ *Where is x ?*
- Static ($x = q$) / Dynamic ($x = [q, \dot{q}]$)

Motion Planning Problem

1. A world W in \mathbb{R}^2 or \mathbb{R}^3
2. An obstacle region O in W
3. A robot A and its configuration space Q
4. $Q_{free} = Q \setminus Q_{obs}$
5. $q_s \in Q_{free}$: Initial Configuration
6. $q_g \in Q_{free}$: Goal Configuration
7. Compute a path $\tau : [0, 1] \rightarrow Q_{free}$ so that $q_s = \tau(0)$, $q_g = \tau(1)$



Roadmap Planning

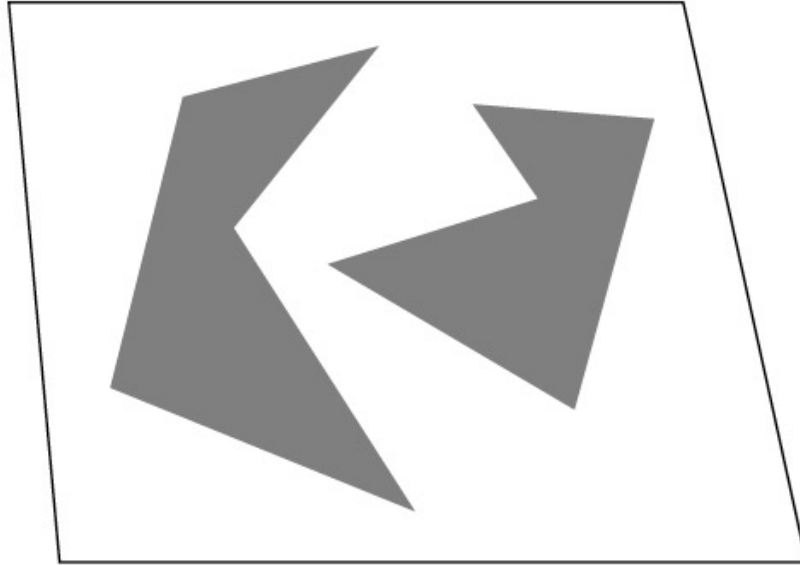


- Construct a network of “highways”
- Store the network as a data structure
- A planner can resolve a query by simply planning a collision-free path to / from the roadmap

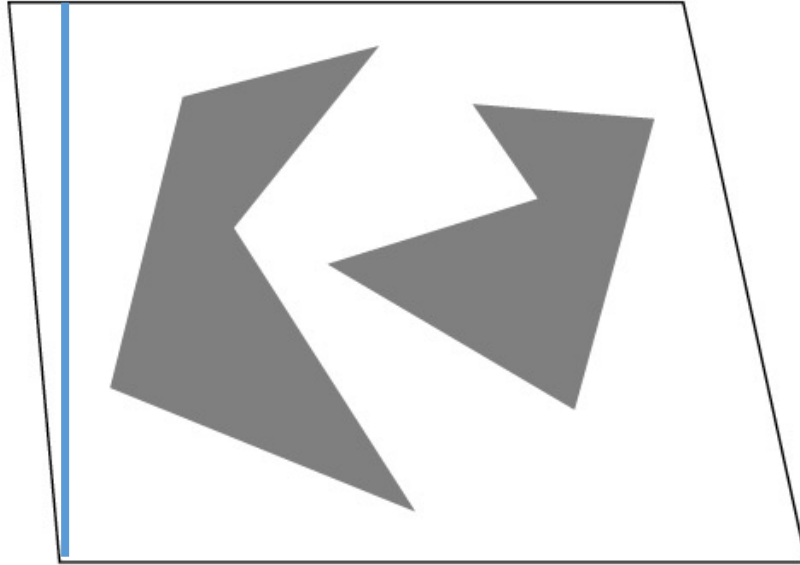
Cell Decomposition Algorithms

- Partition a space Q_{free} into a set of regions named cells
- Example: vertical cell decomposition
 - Let P denote a set of vertices
 - At every $p \in P$, extend a ray in Q_{free} until you hit Q_{obs}

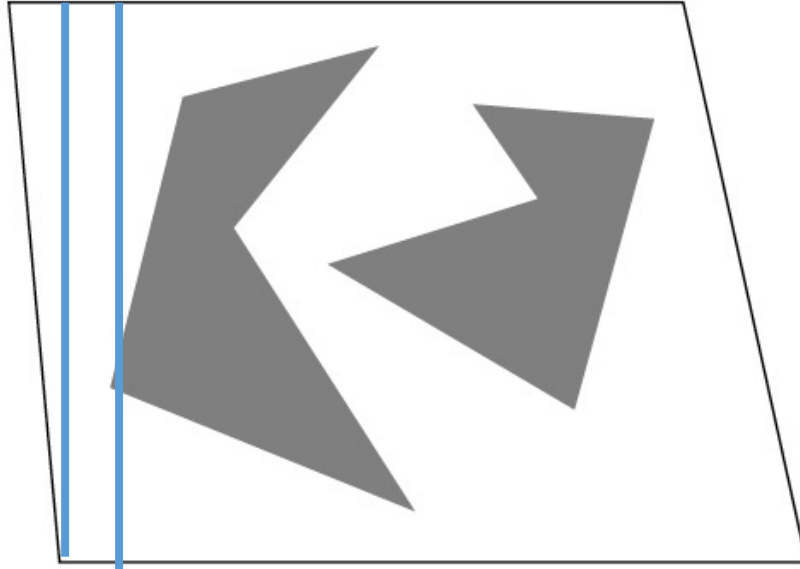
Vertical Cell Decomposition



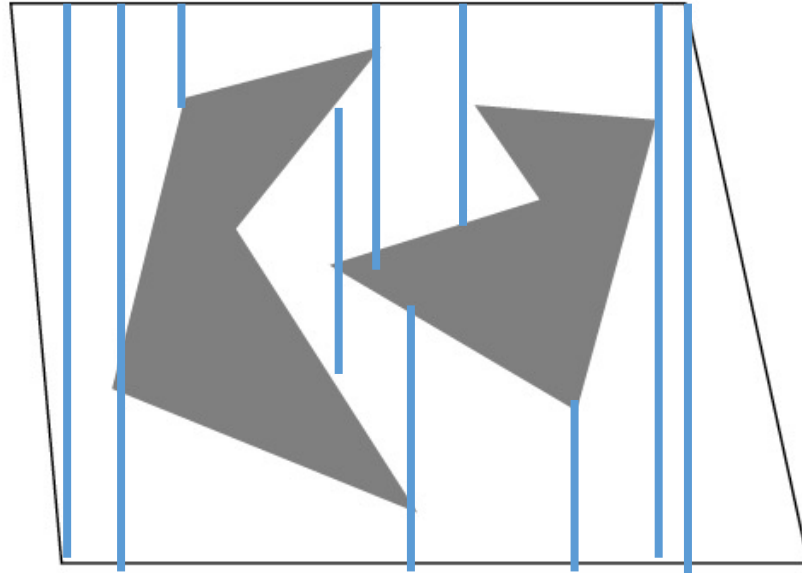
Vertical Cell Decomposition



Vertical Cell Decomposition



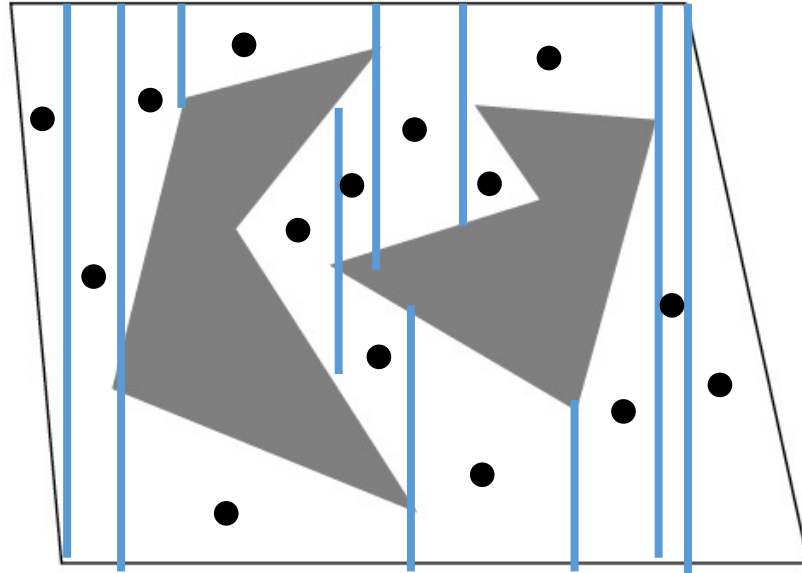
Vertical Cell Decomposition



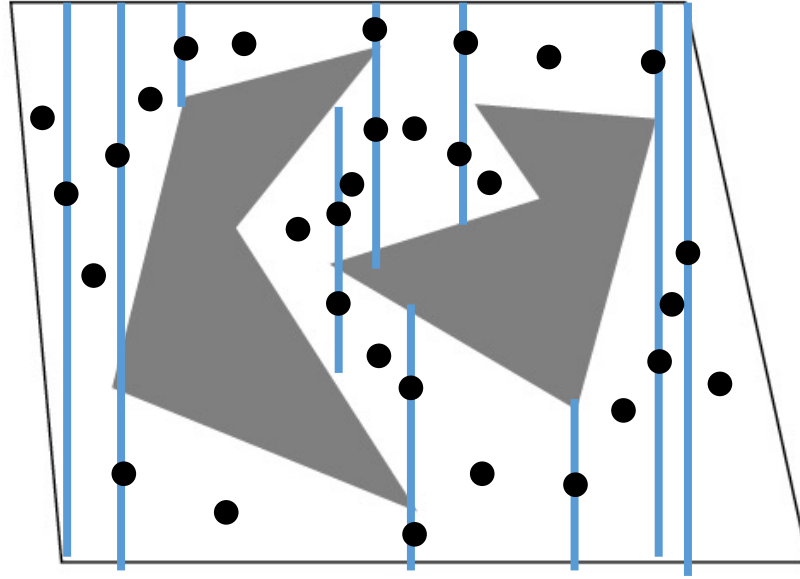
Cell Decomposition Algorithms

- Partition a space Q_{free} into a set of regions named cells
- Example: vertical cell decomposition
 - Let P denote a set of vertices
 - At every $p \in P$, extend a ray in Q_{free} until you hit Q_{obs}
 - For each cell i , select a point q_i , e.g., centroids
 - 1-cell: line segment
 - 2-cell: trapezoid

Vertical Cell Decomposition



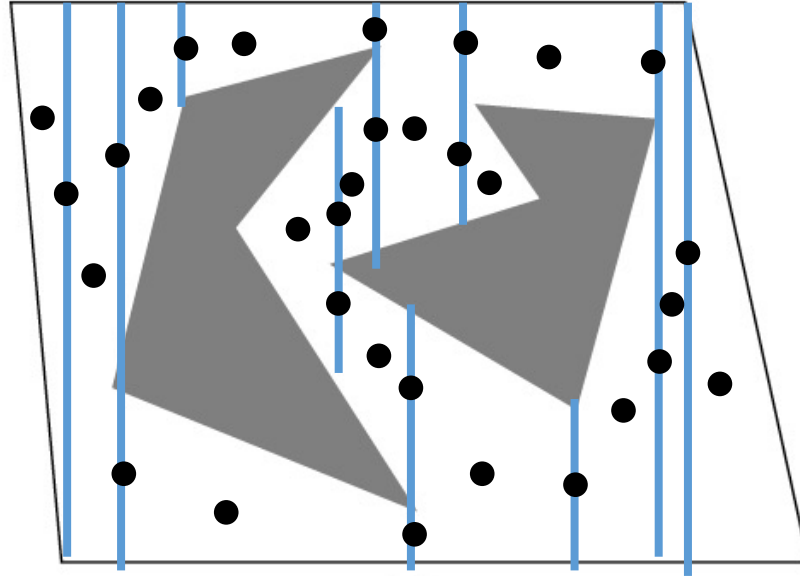
Vertical Cell Decomposition



Cell Decomposition Algorithms

- Partition a space Q_{free} into a set of regions named cells
- Example: vertical cell decomposition
 - Let P denote a set of vertices
 - At every $p \in P$, extend a ray in Q_{free} until you hit Q_{obs}
 - For each cell i , select a point q_i , e.g., centroids
 - 1-cell: line segment
 - 2-cell: trapezoid
 - Build a graph $G(V,E)$, where each point q_i is a 2-cell connected with the point of each 1-cell that lies along its boundary.

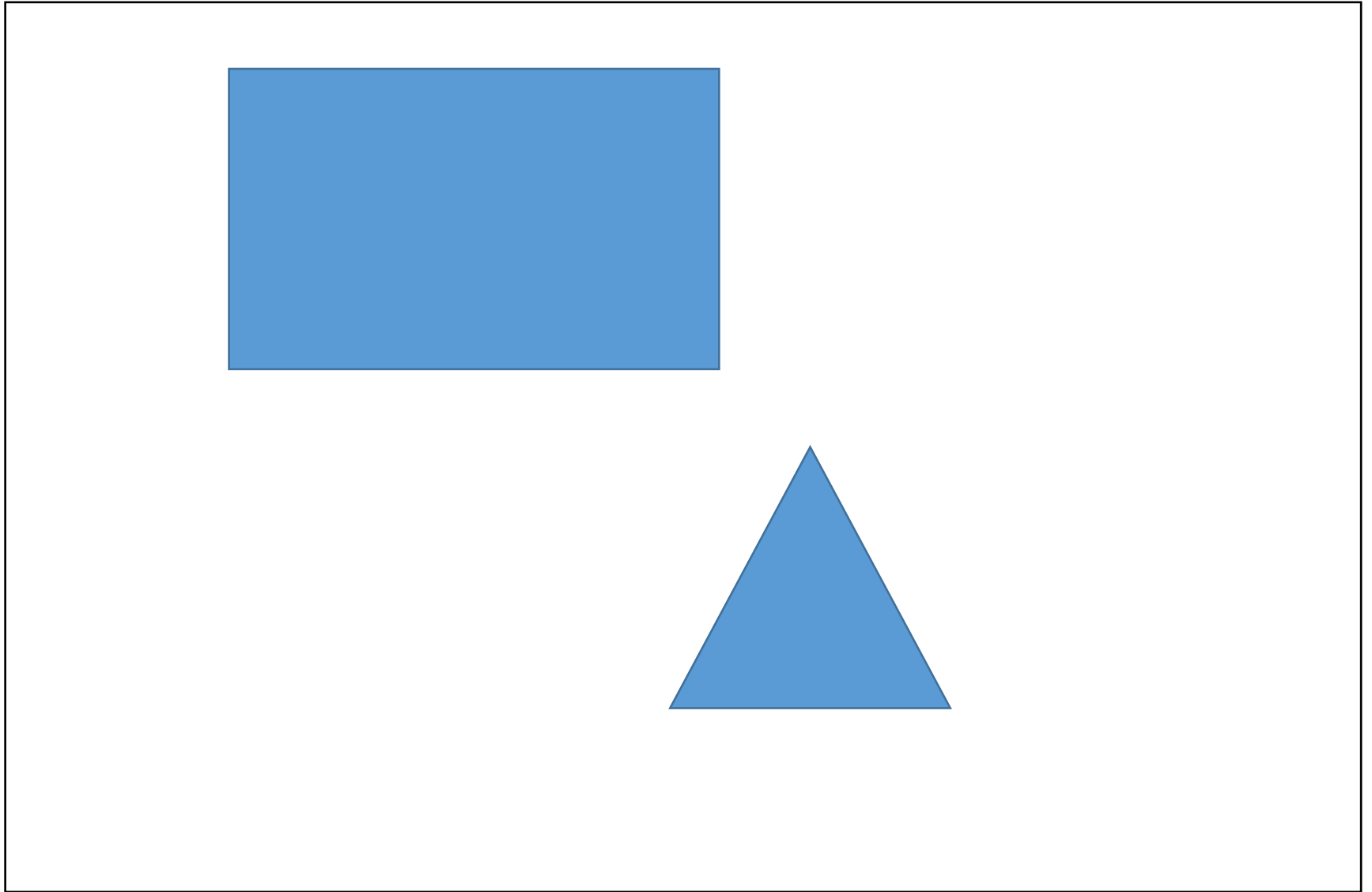
Vertical Cell Decomposition



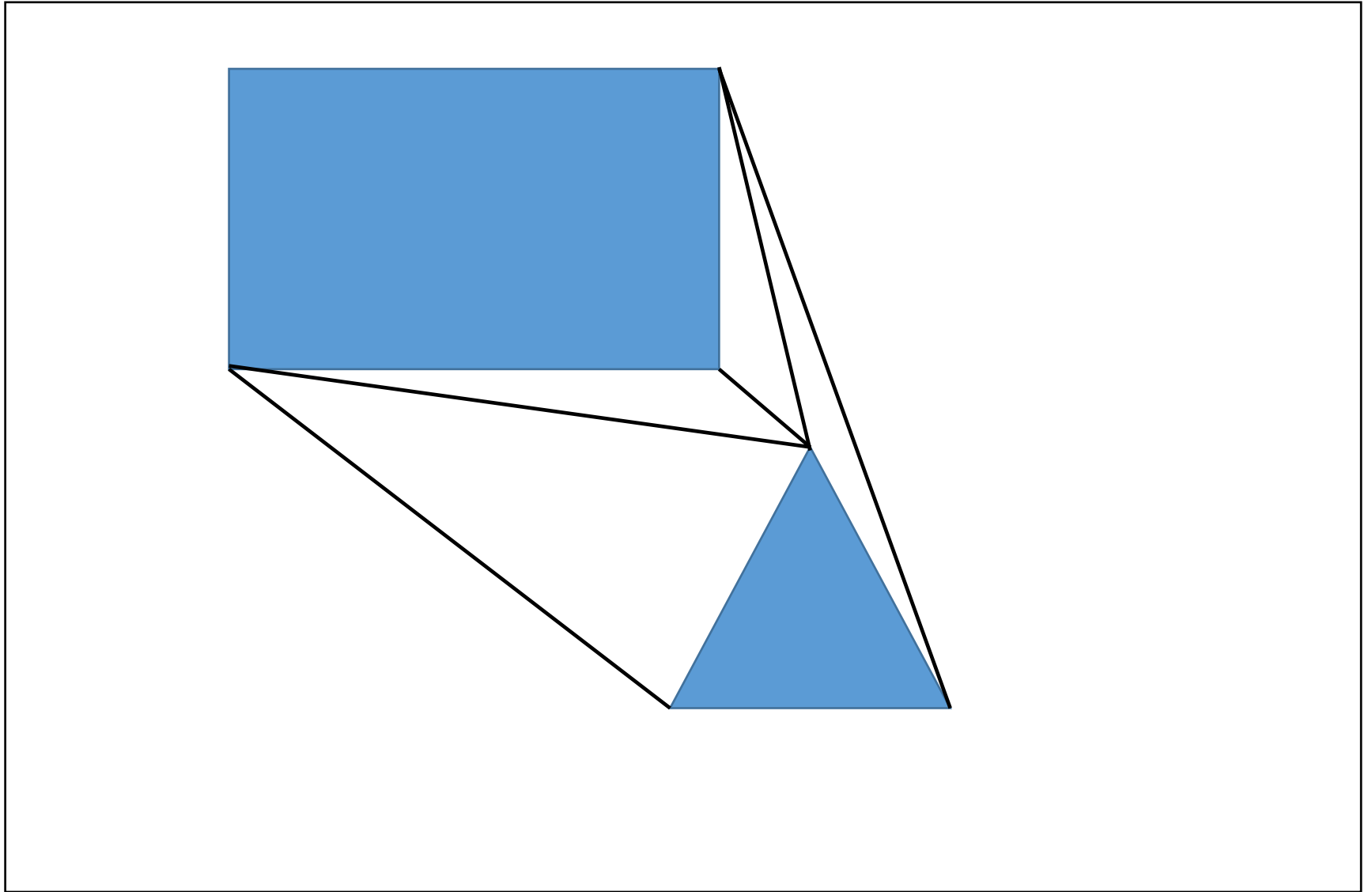
Visibility Graph

- Connect each vertex v to the set of vertices that are visible from v

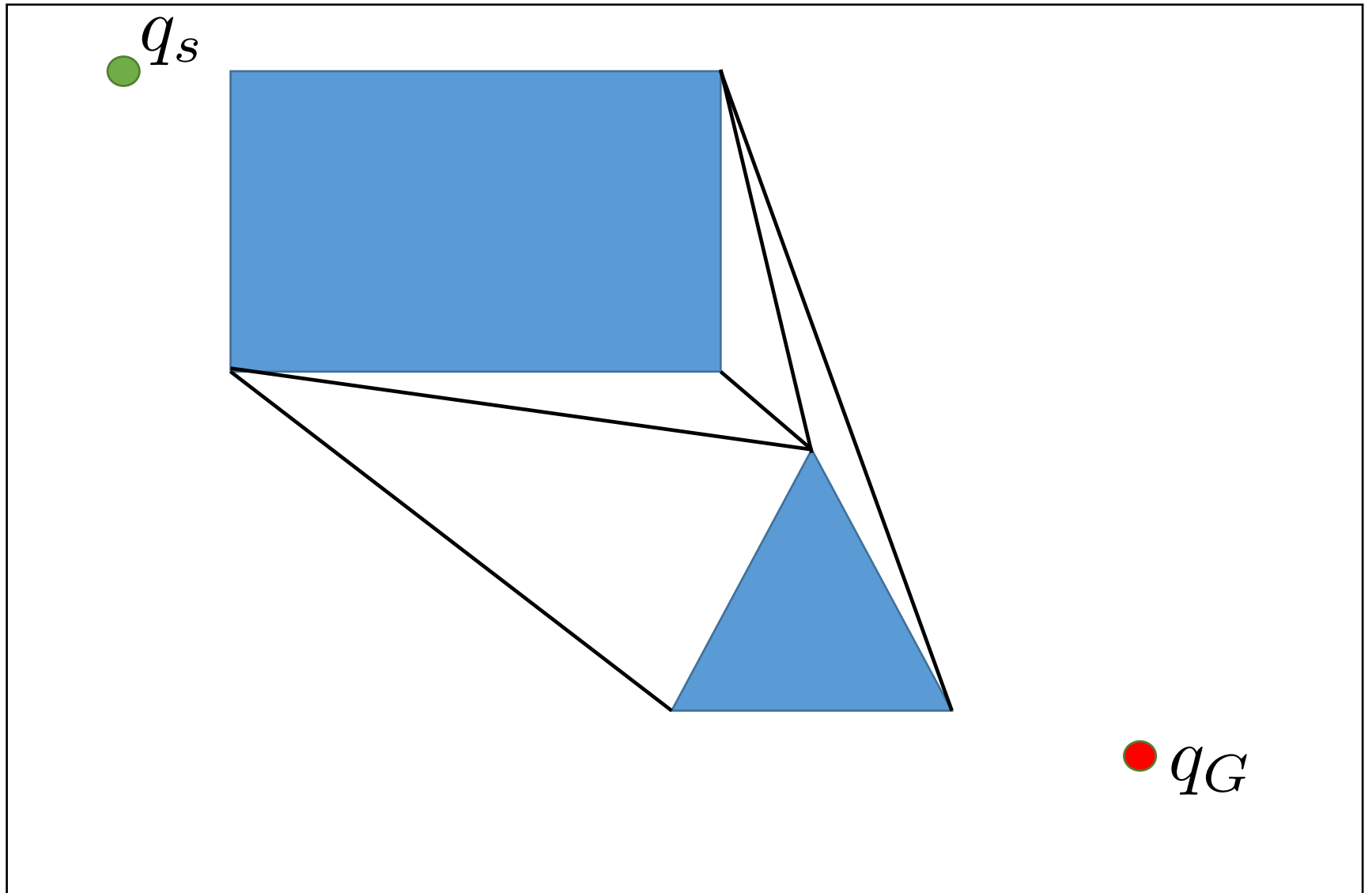
Visibility Graph



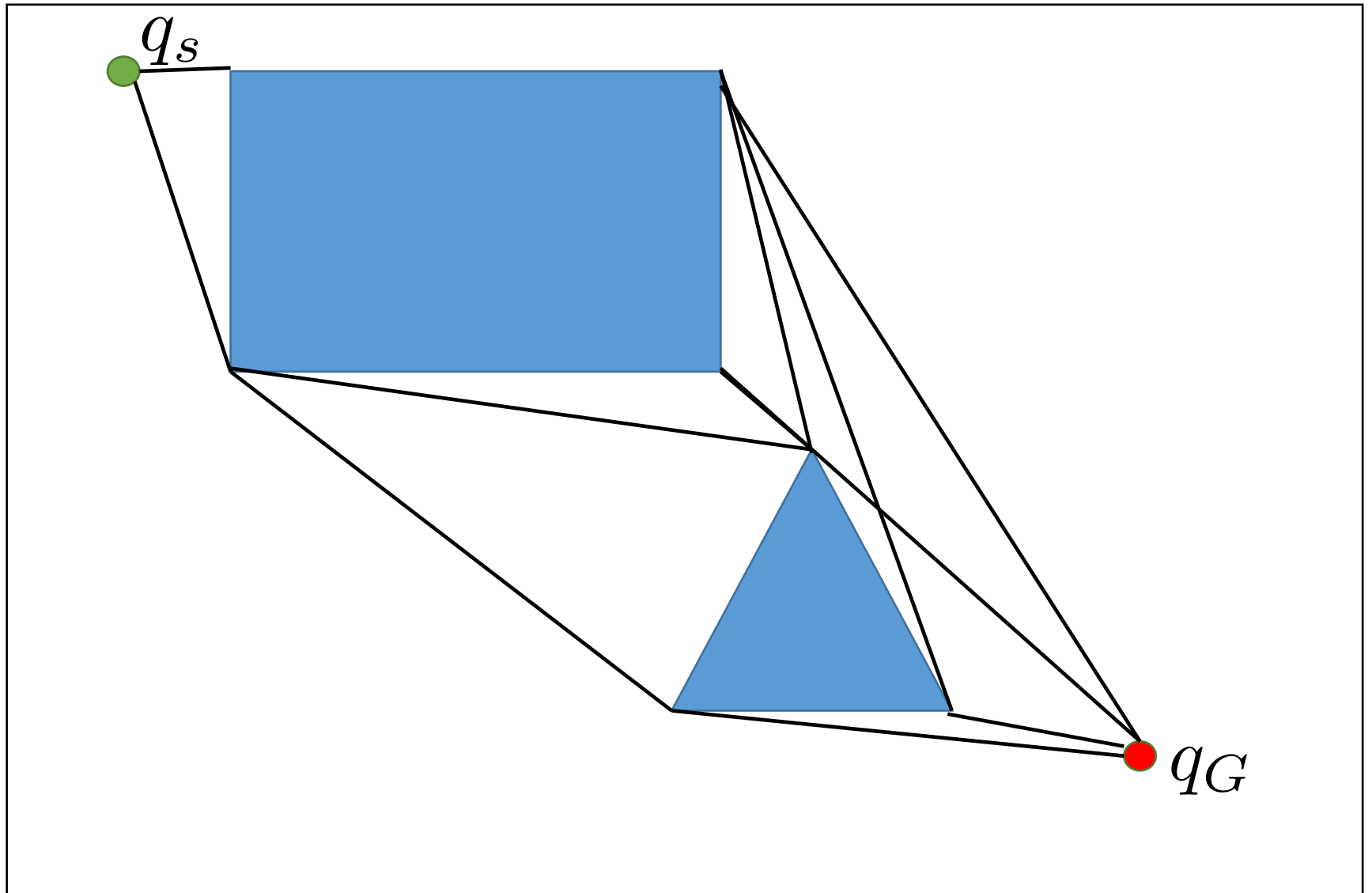
Visibility Graph



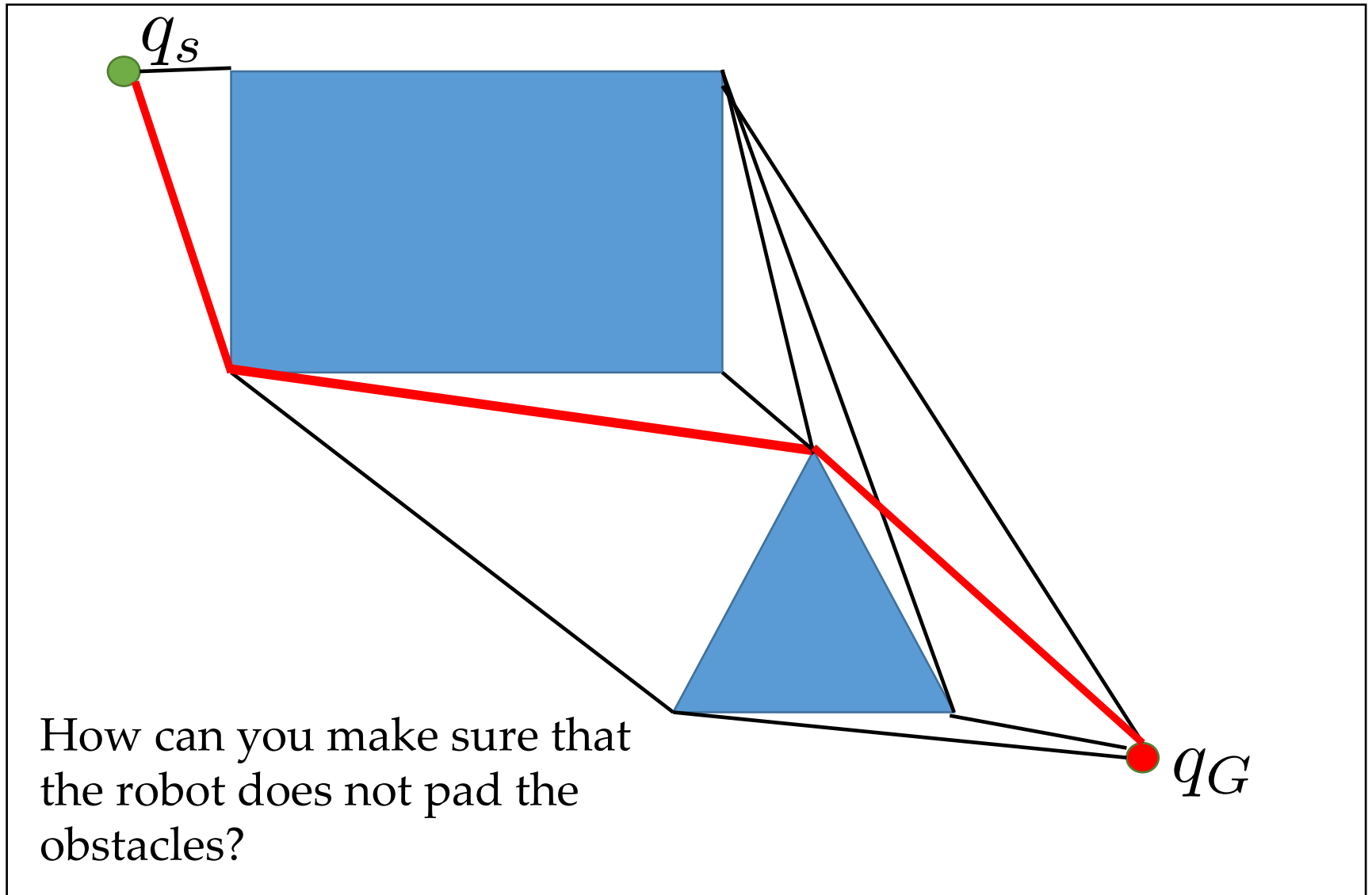
Visibility Graph



Visibility Graph

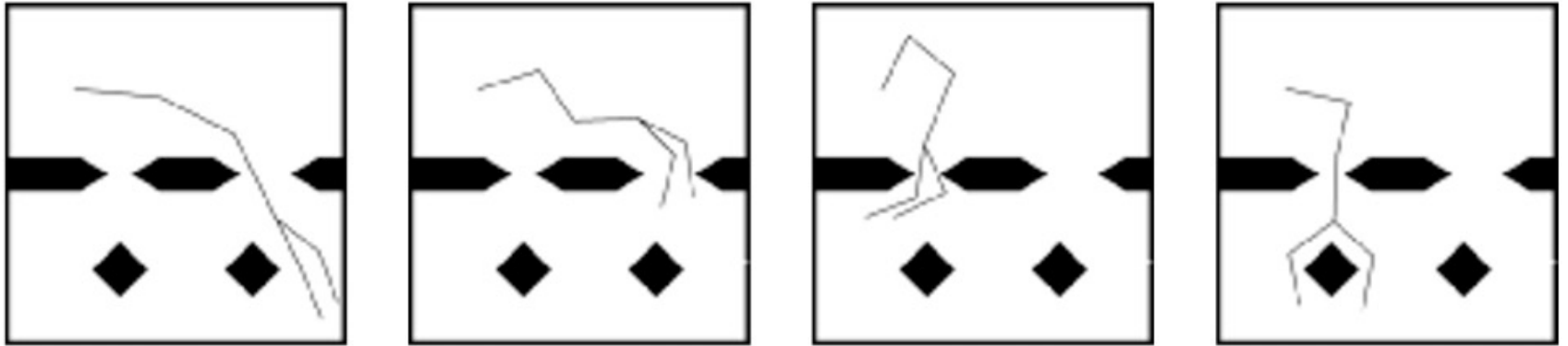


Visibility Graph



Sampling-Based Motion Planning

- As the dimension of the configuration space increases, planning algorithms based on the geometry of the space become impractical.



Probabilistic Roadmap Planner

- Preprocessing phase: construct Roadmap
- Query phase: search roadmap for q_s and q_g
- Does not attempt to construct explicitly the boundaries of the configuration space obstacles!

Probabilistic Roadmap Planner (PRM)

Algorithm 6 Roadmap Construction Algorithm

Input:

n : number of nodes to put in the roadmap

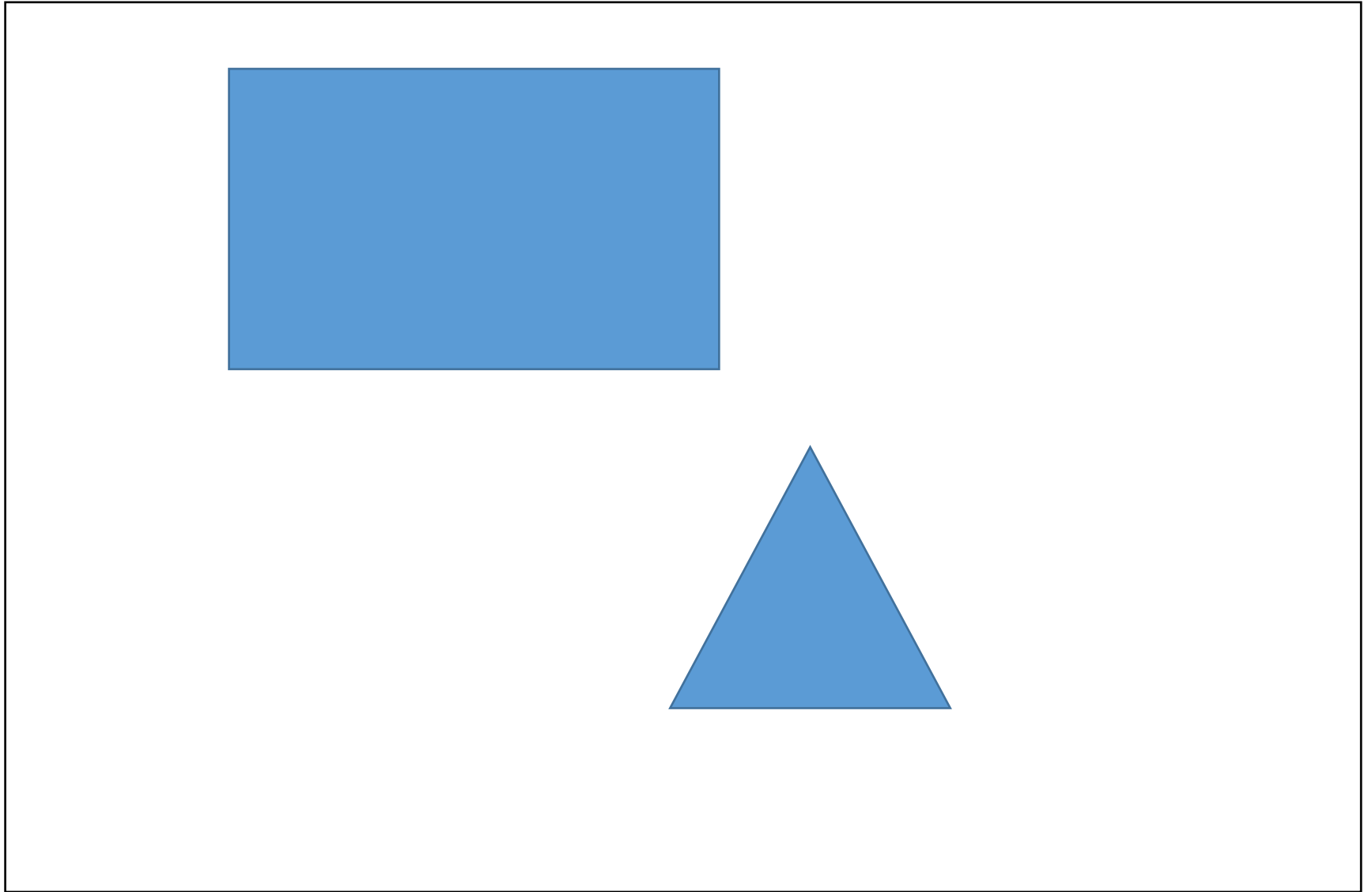
k : number of closest neighbors to examine for each configuration

Output:

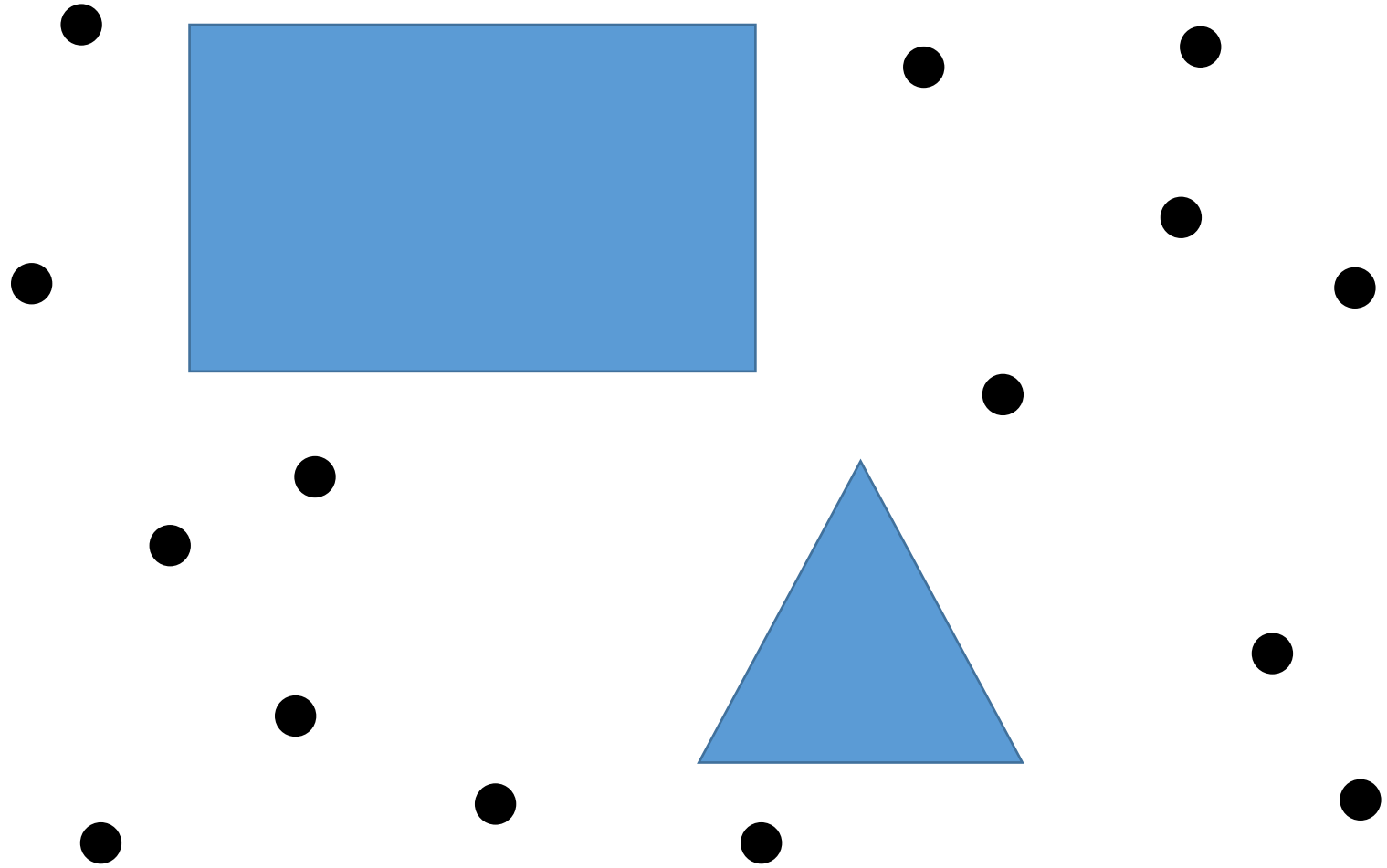
A roadmap $G = (V, E)$

```
1:  $V \leftarrow \emptyset$ 
2:  $E \leftarrow \emptyset$ 
3: while  $|V| < n$  do
4:   repeat
5:      $q \leftarrow$  a random configuration in  $Q$ 
6:   until  $q$  is collision-free
7:    $V \leftarrow V \cup \{q\}$ 
8: end while
9: for all  $q \in V$  do
10:   $N_q \leftarrow$  the  $k$  closest neighbors of  $q$  chosen from  $V$  according to  $dist$ 
11:  for all  $q' \in N_q$  do
12:    if  $(q, q') \notin E$  and  $\Delta(q, q') \neq \text{NIL}$  then
13:       $E \leftarrow E \cup \{(q, q')\}$ 
14:    end if
15:  end for
16: end for
```

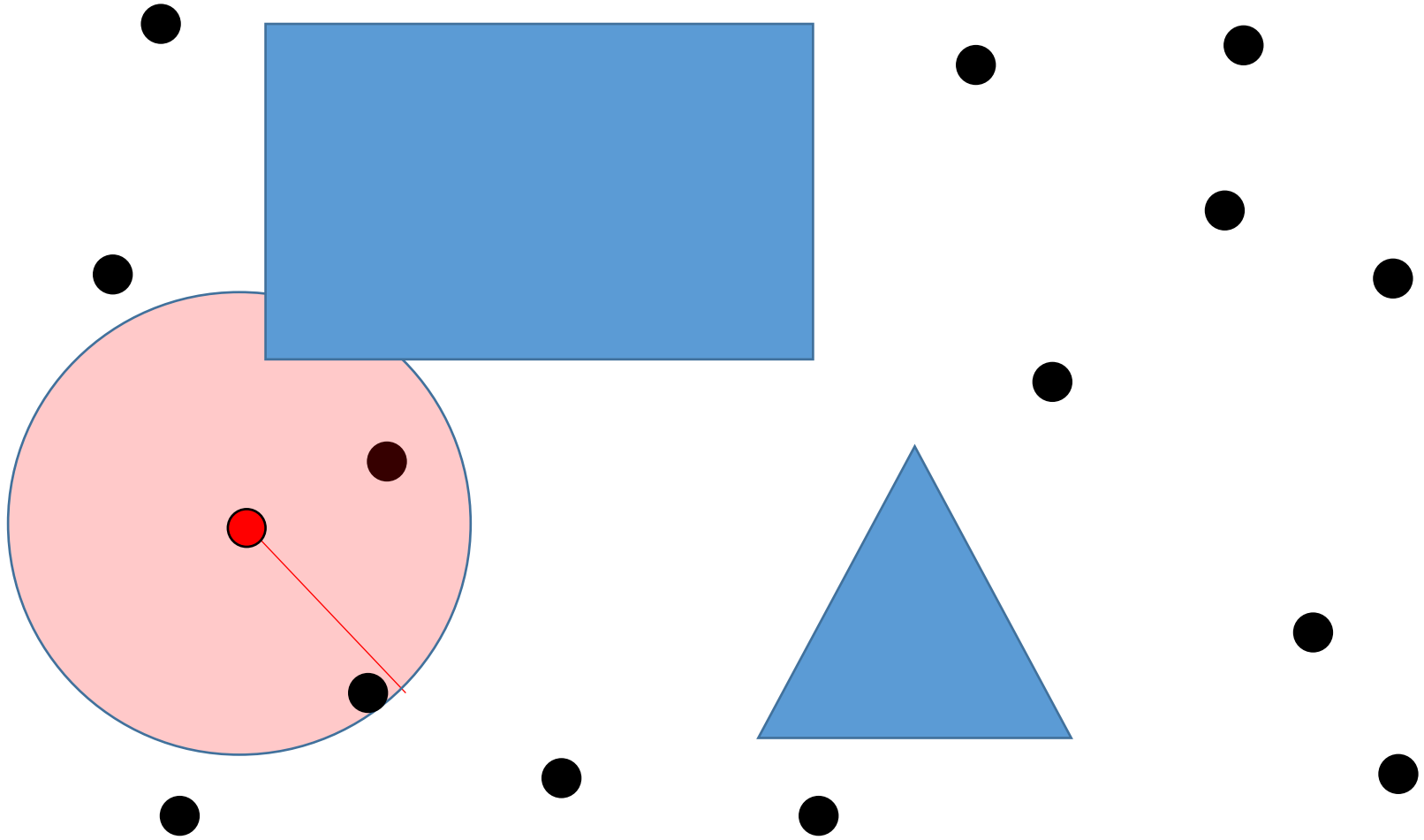
Probabilistic Roadmap Planner (PRM)



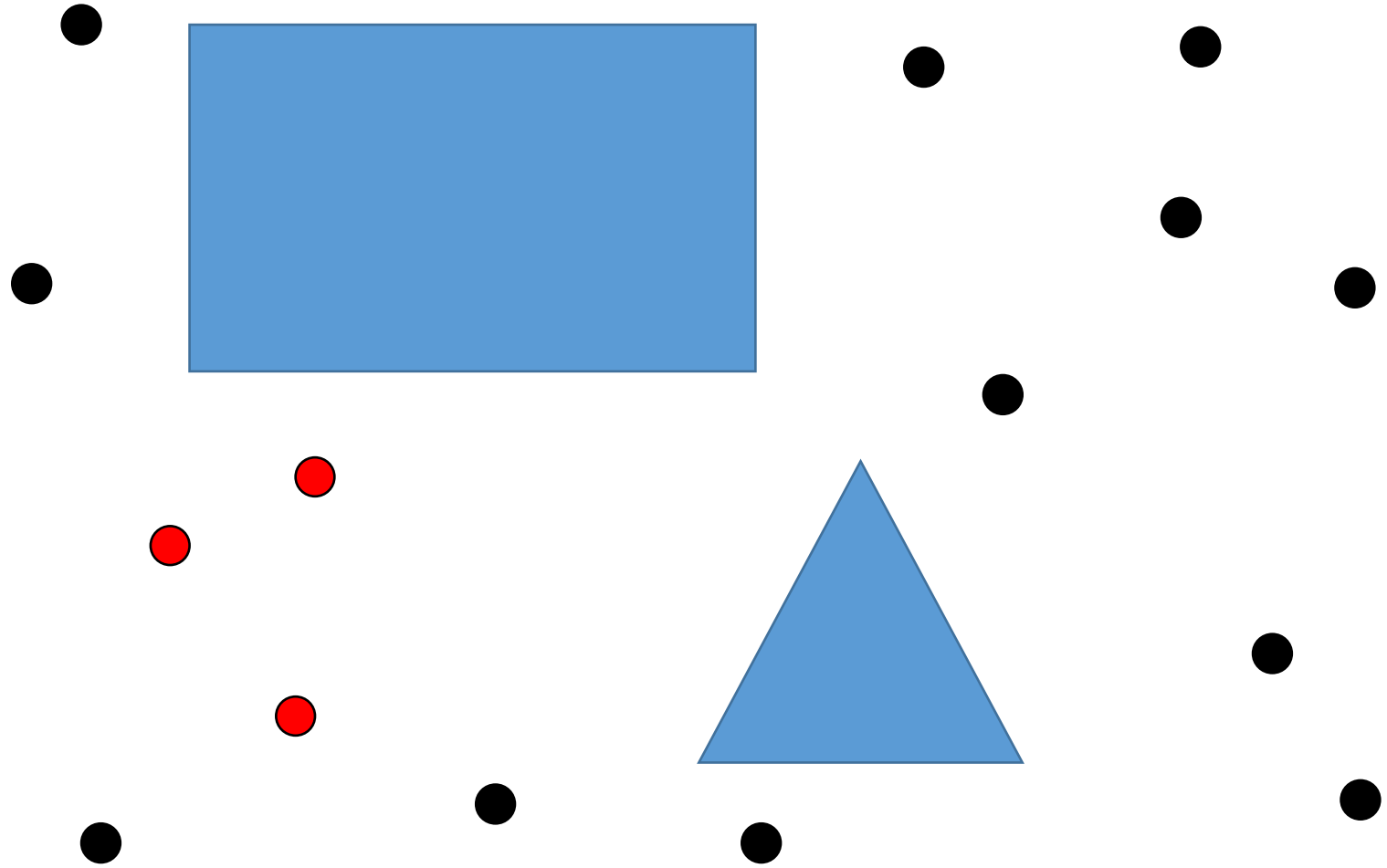
Probabilistic Roadmap Planner (PRM)



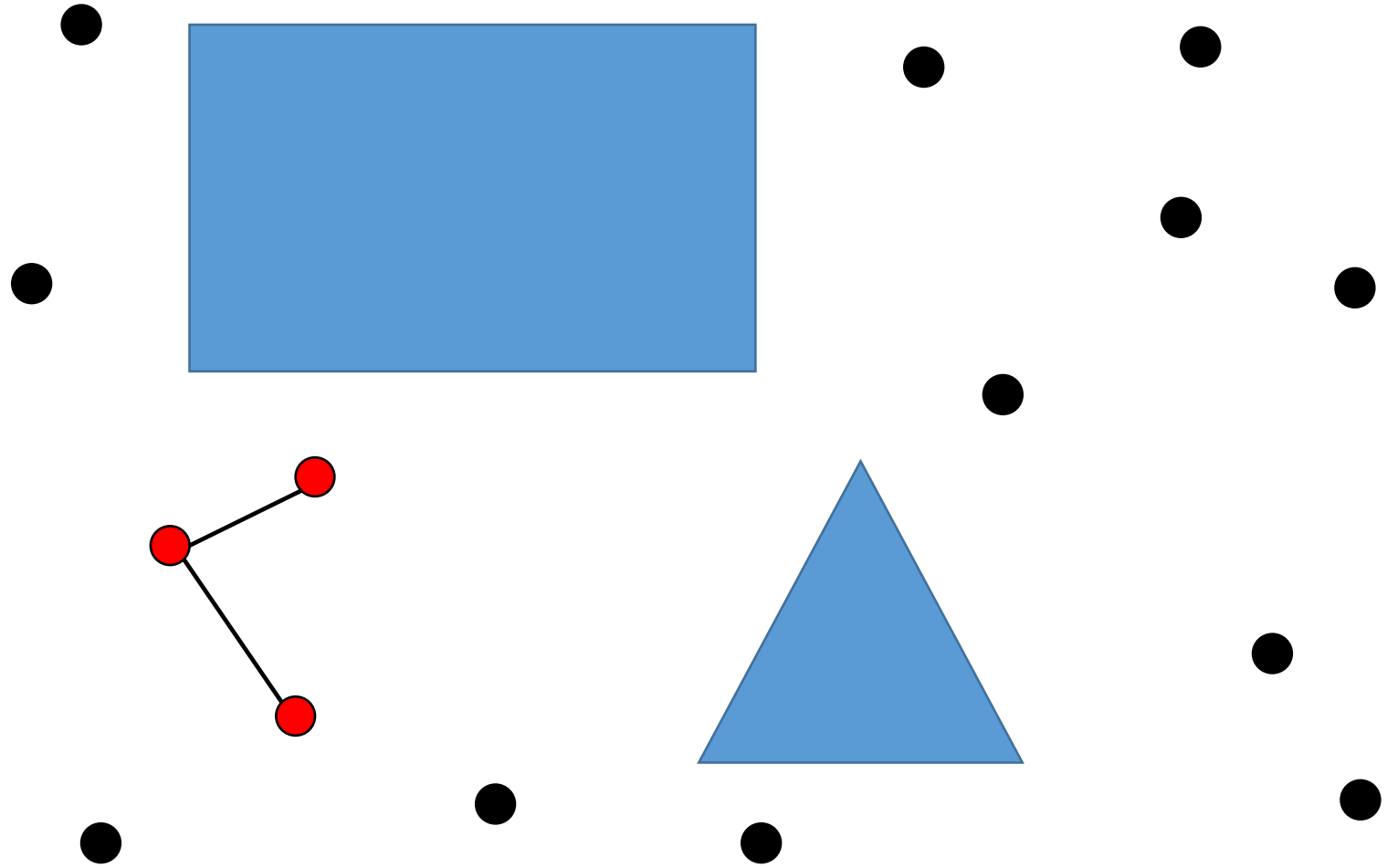
Probabilistic Roadmap Planner (PRM)



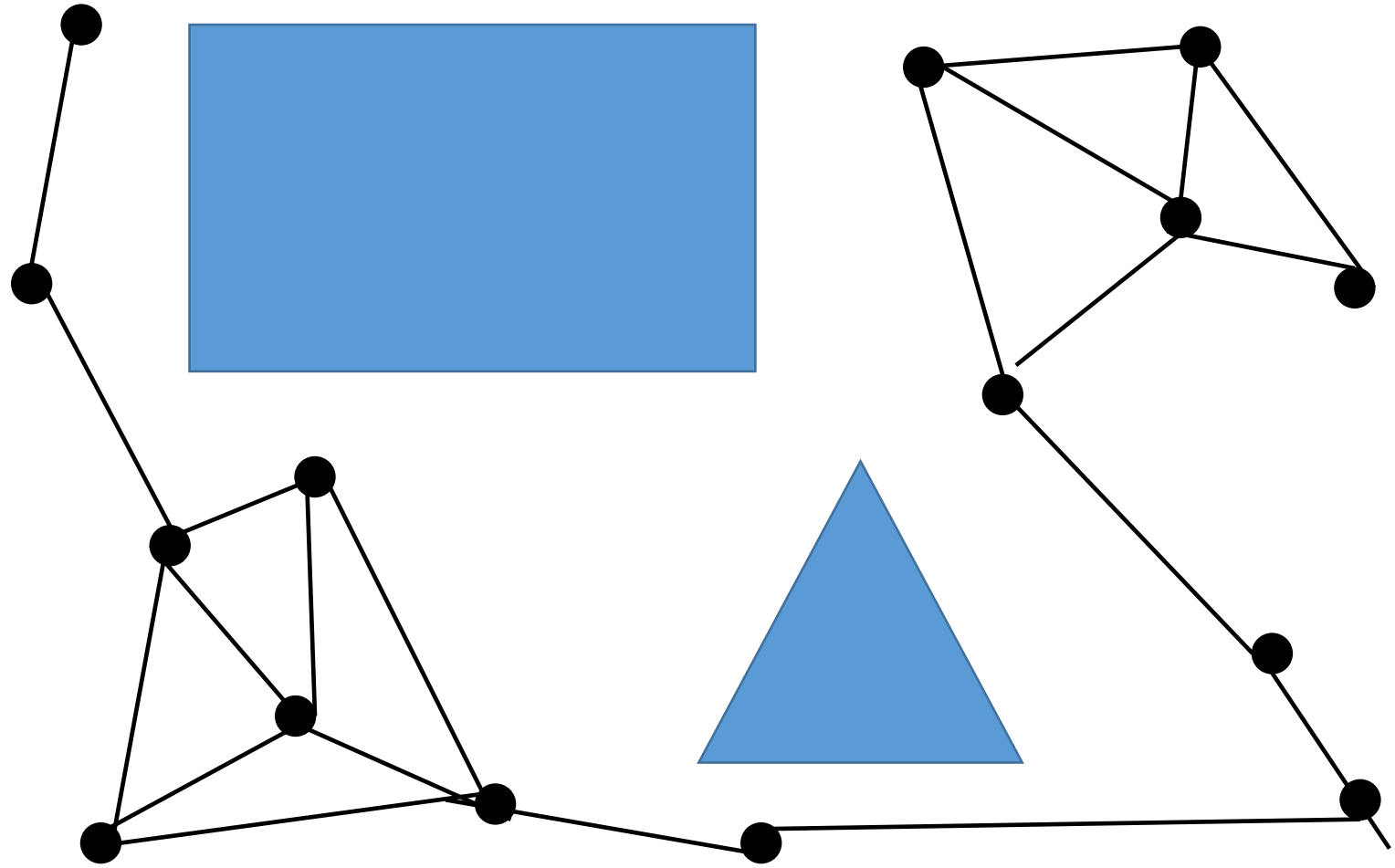
Probabilistic Roadmap Planner (PRM)



Probabilistic Roadmap Planner (PRM)

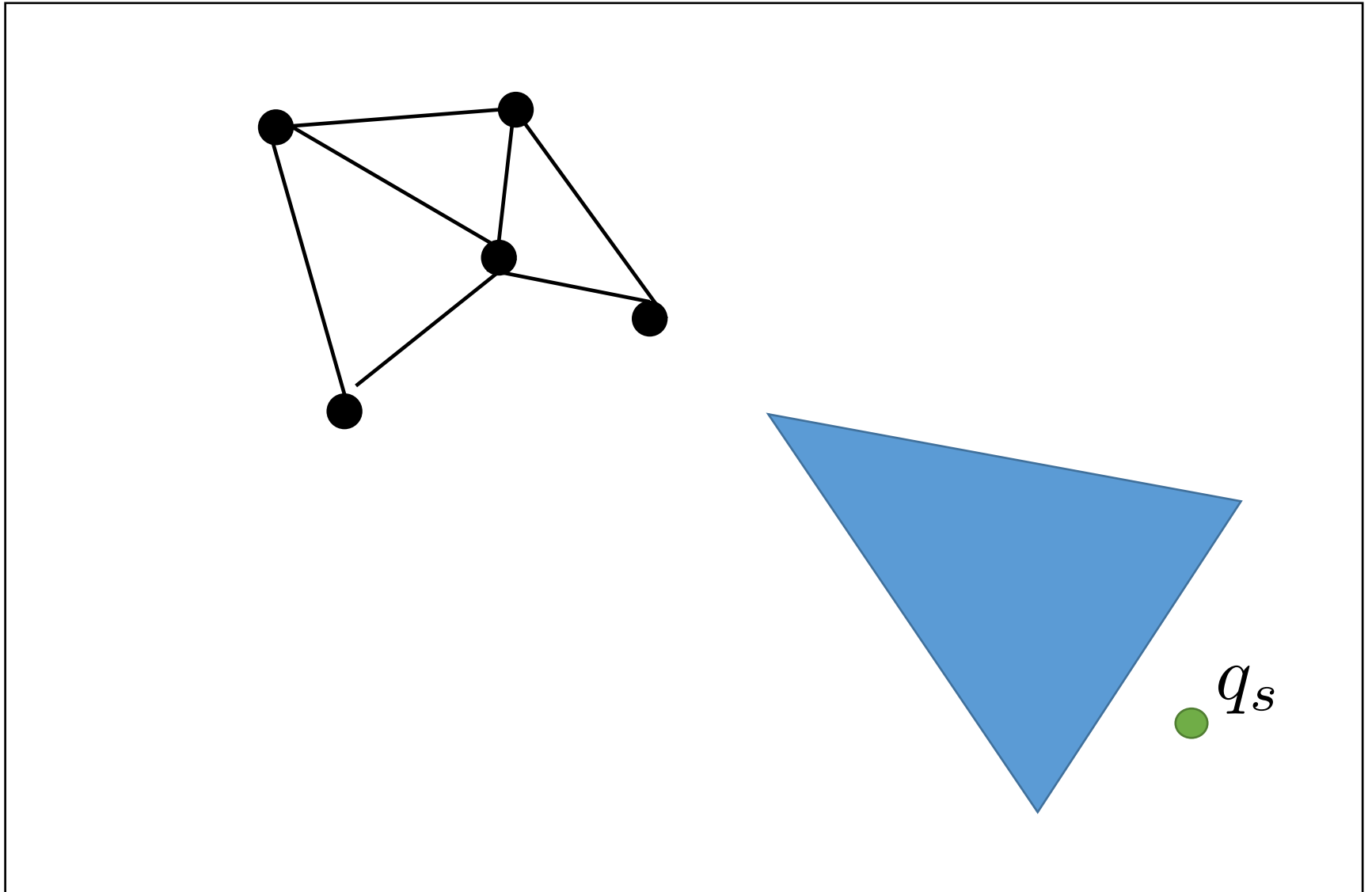


Probabilistic Roadmap Planner (PRM)

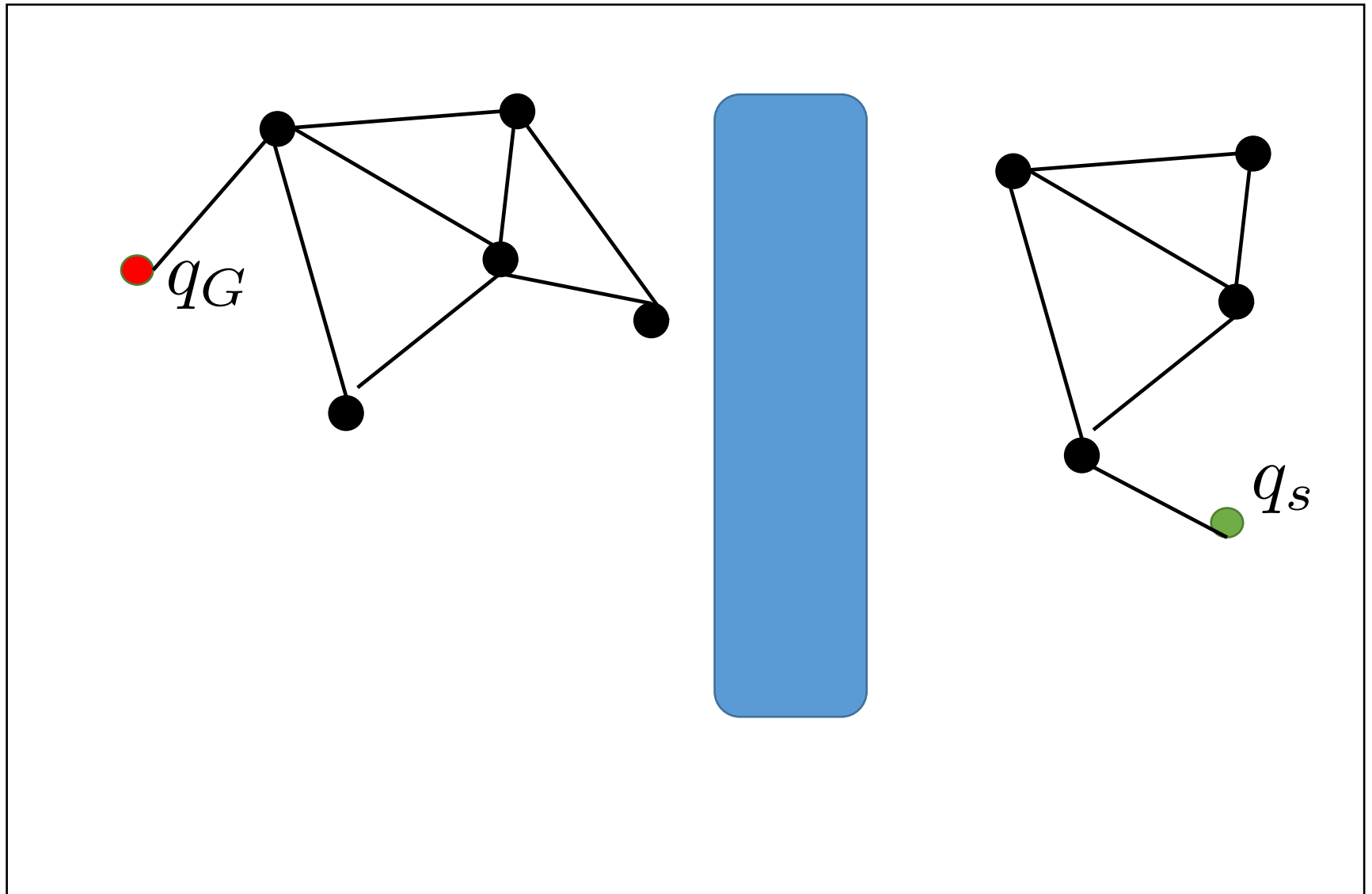


Limitations

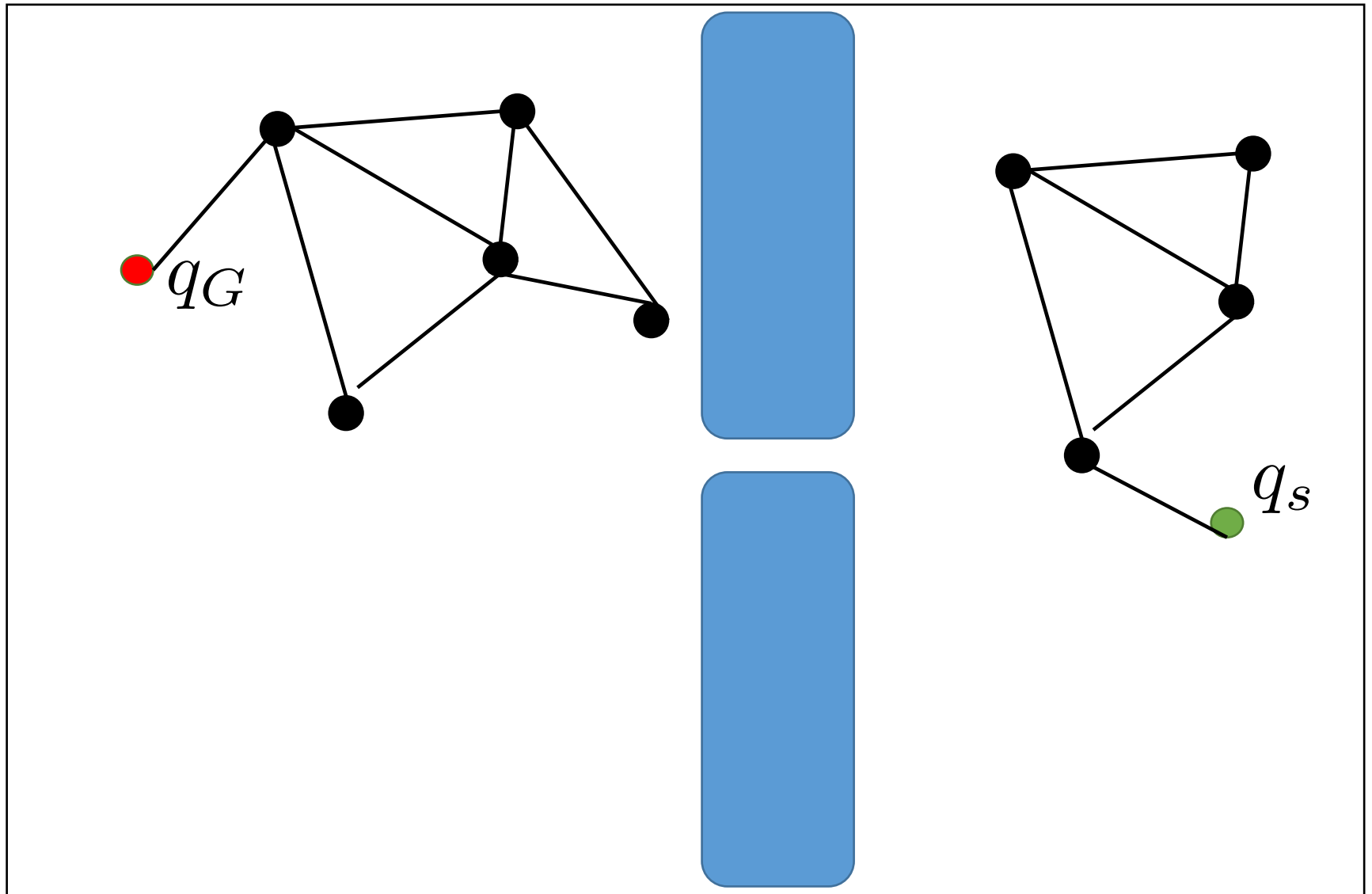
Limitations: Coverage



Limitations: Connectivity



Example: Narrow Passage Problem



Solutions

Solutions

- Sample near obstacles
- Visibility-based PRM

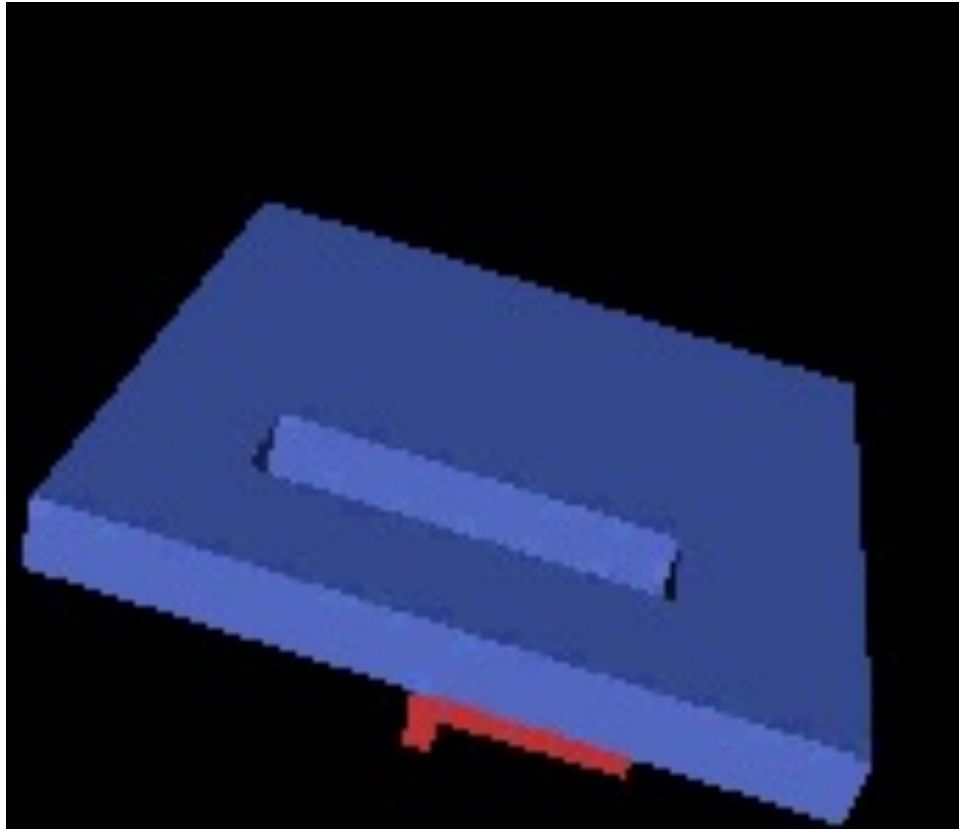
Visibility-based probabilistic roadmaps for motion planning

T.Simeon, J.-P. Laumond and C. Nissoux
Advanced Robotics, 2000

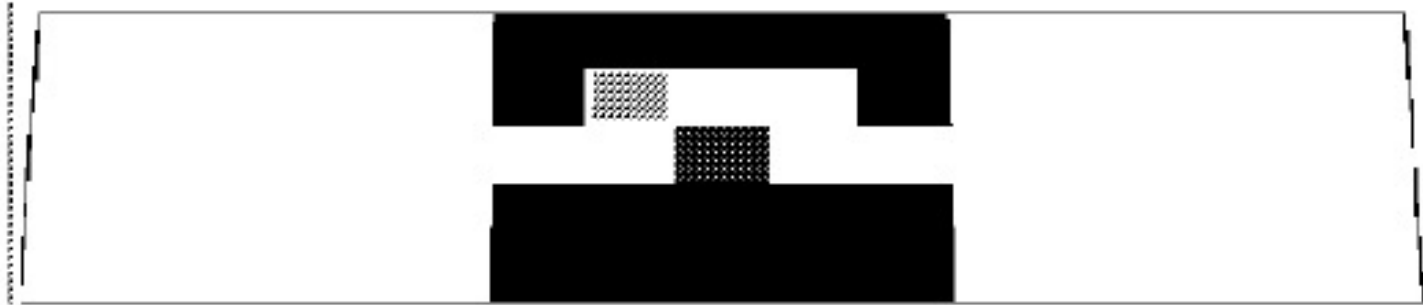
Problem

Improve performance of multi-query
sampling-based motion-planning algorithms

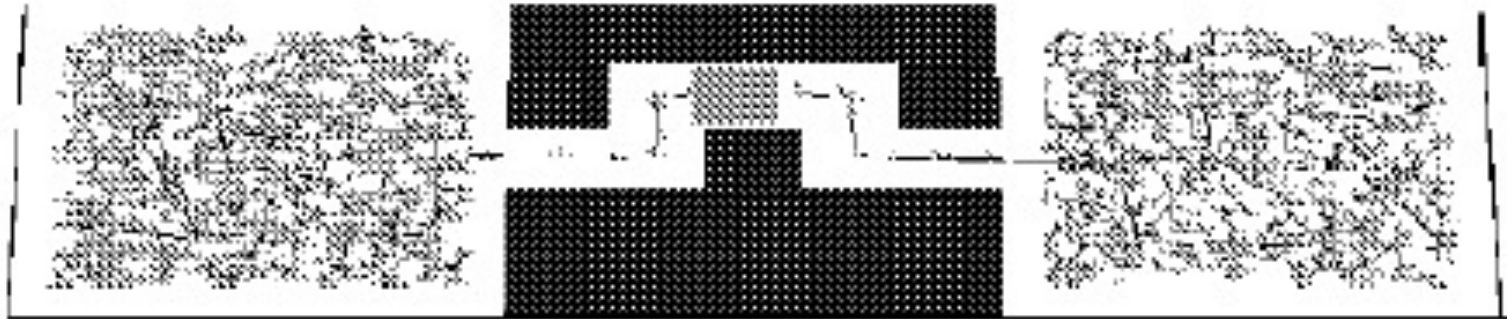
Problem



Problem

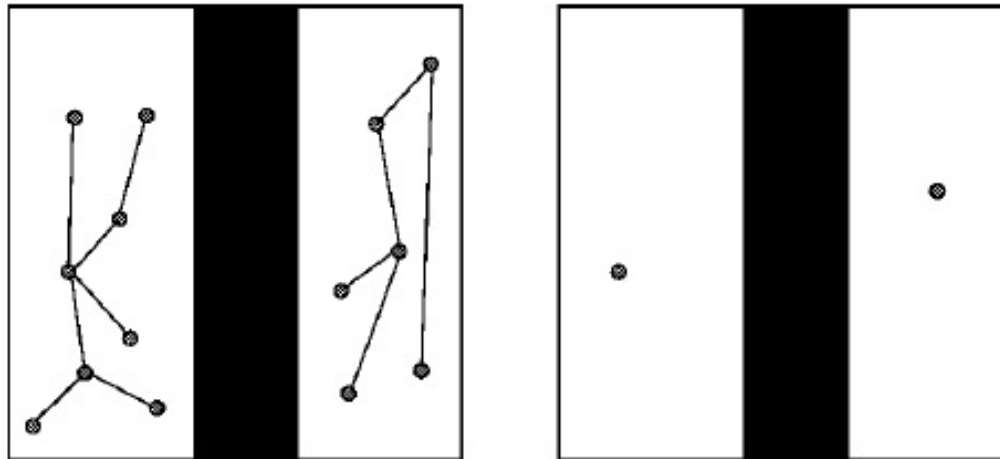


Problem



Key Insight

Use notion of “visibility” to cover the free space, while keeping the size of the roadmap small

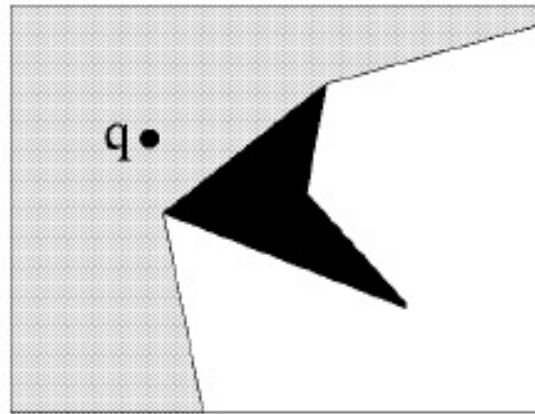


Definitions

$L(q, q')$: A feasible path between configurations q and q' , in the absence of any obstacle

Definitions

$$Vis_{\mathcal{L}}(q) = \{q' \in CS_{\text{free}} \text{ such that } \mathcal{L}(q, q') \subset CS_{\text{free}}\}$$



Configuration q : Guard of $Vis_{\mathcal{L}}(q)$

Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

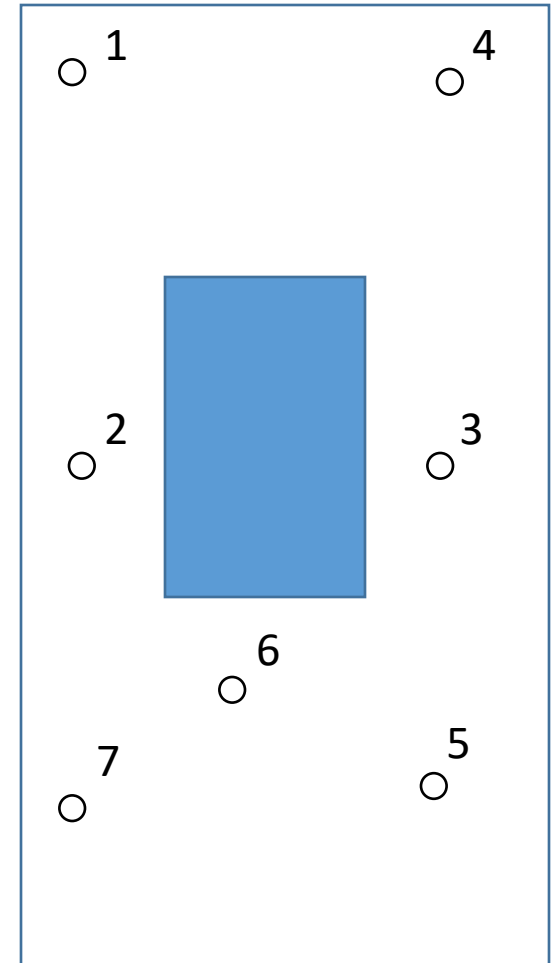
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

 Add q to $Connection$

 Create edges (q, g) and (q, g_{vis})

 Merge components G_{vis} and G_i ;

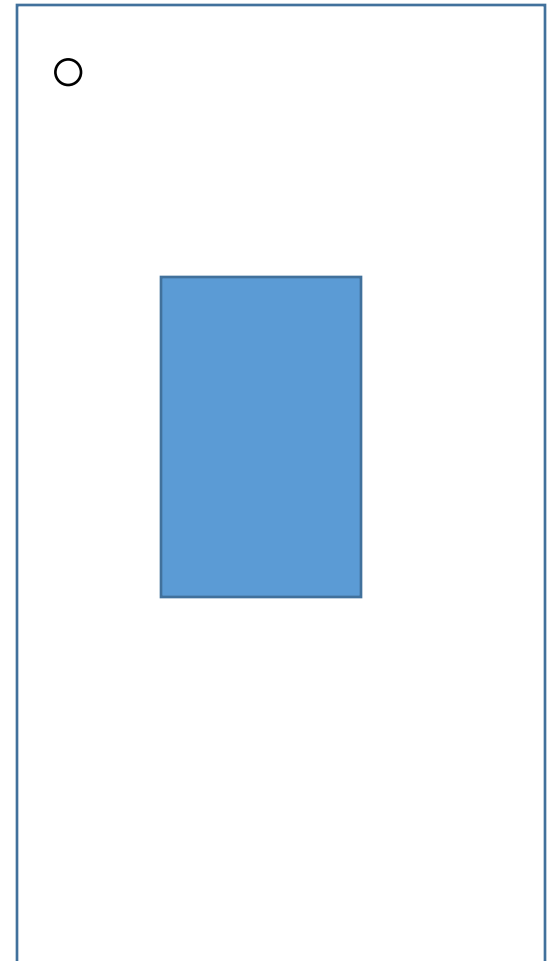
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

 Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

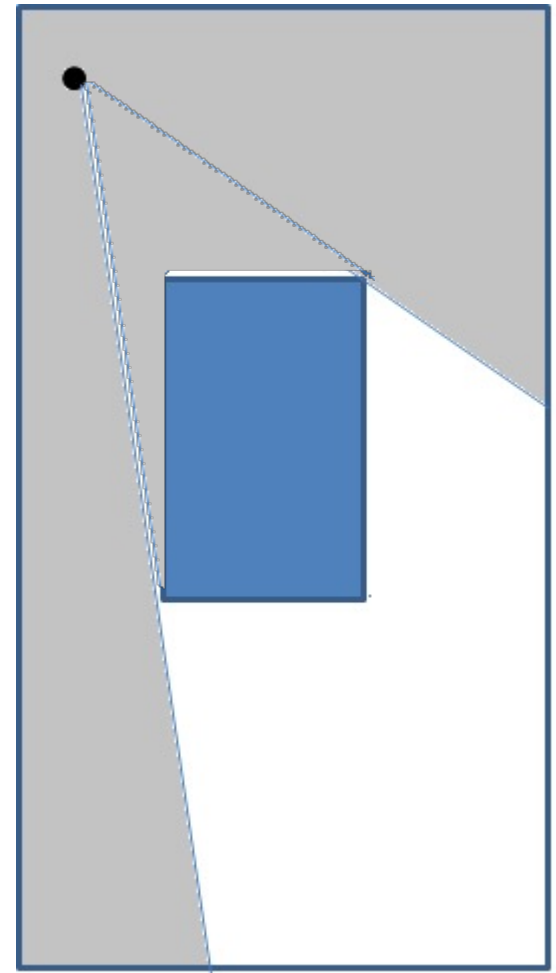
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

 Add q to $Connection$

 Create edges (q, g) and (q, g_{vis})

 Merge components G_{vis} and G_i ;

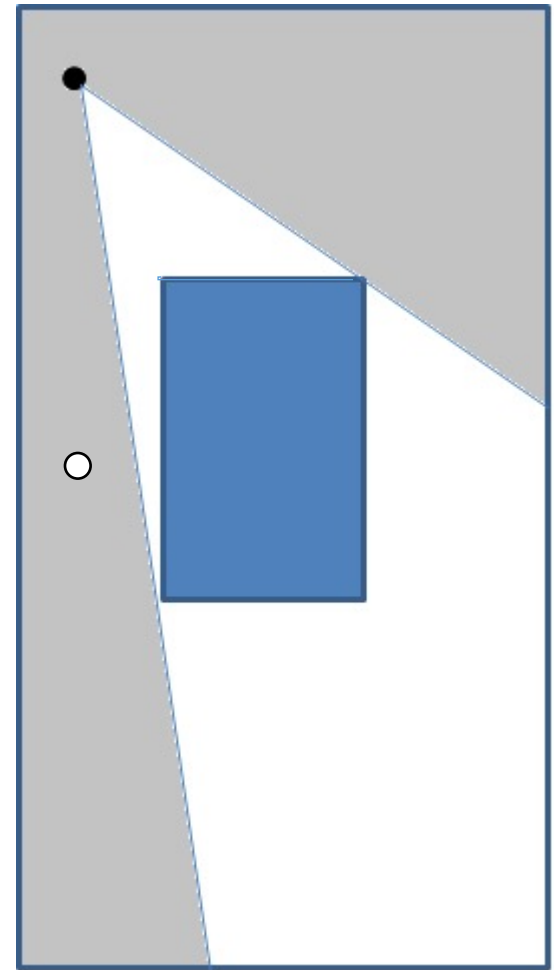
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

 Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

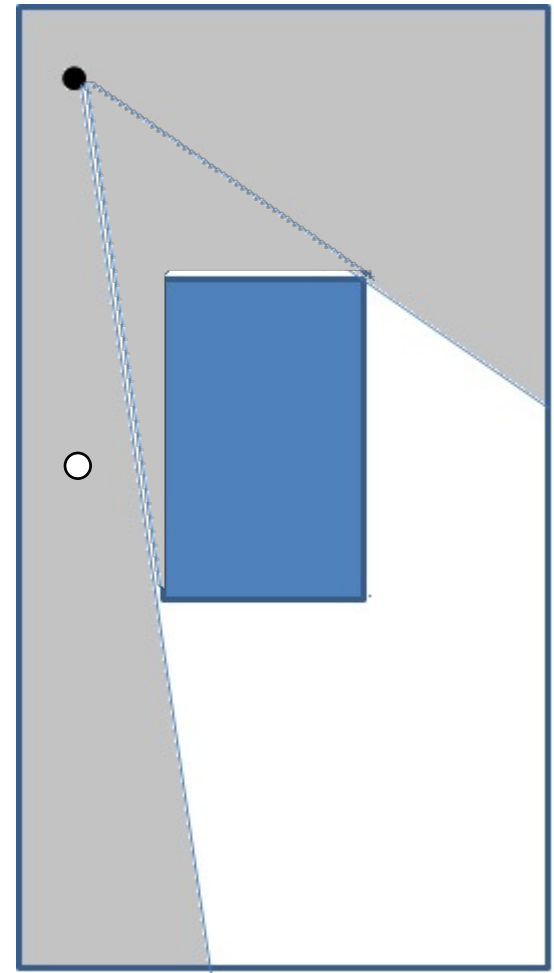
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

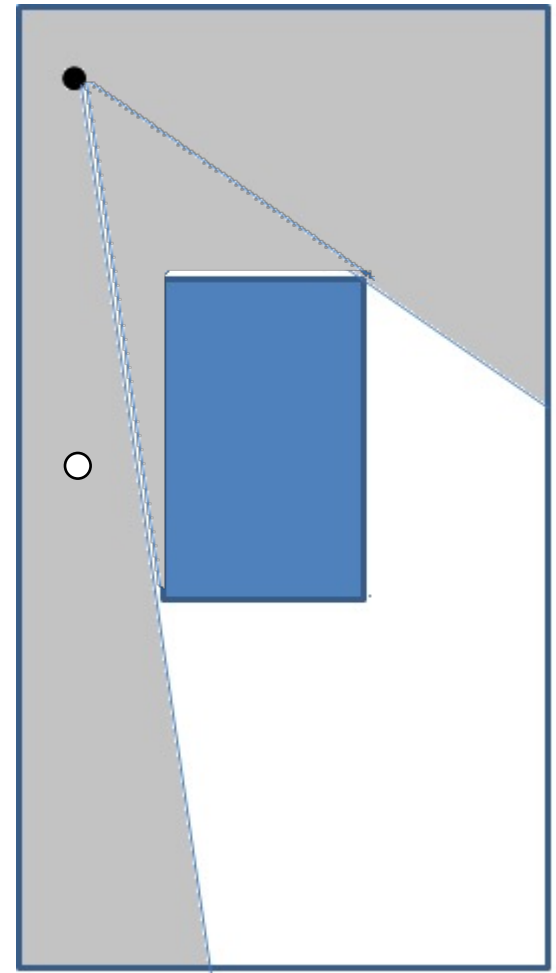
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

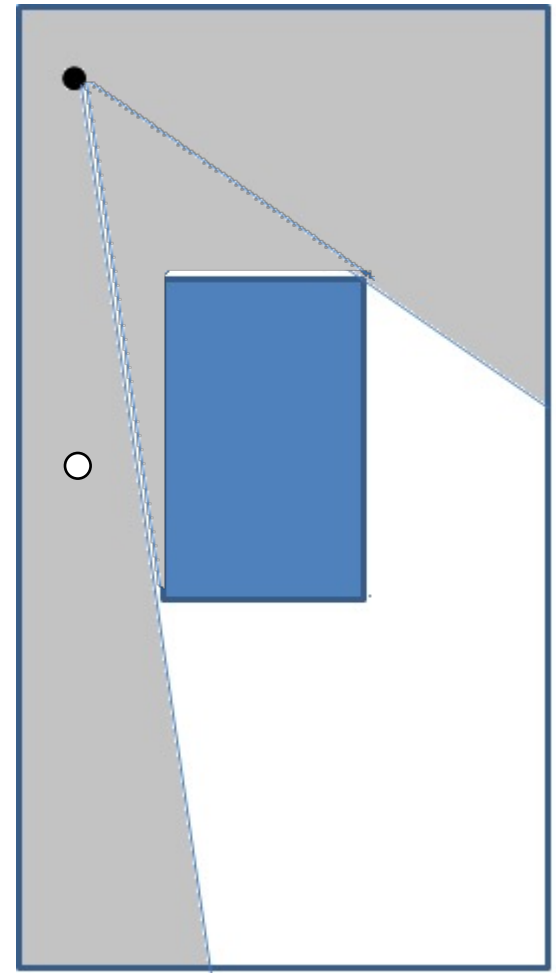
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

 Add q to $Connection$

 Create edges (q, g) and (q, g_{vis})

 Merge components G_{vis} and G_i ;

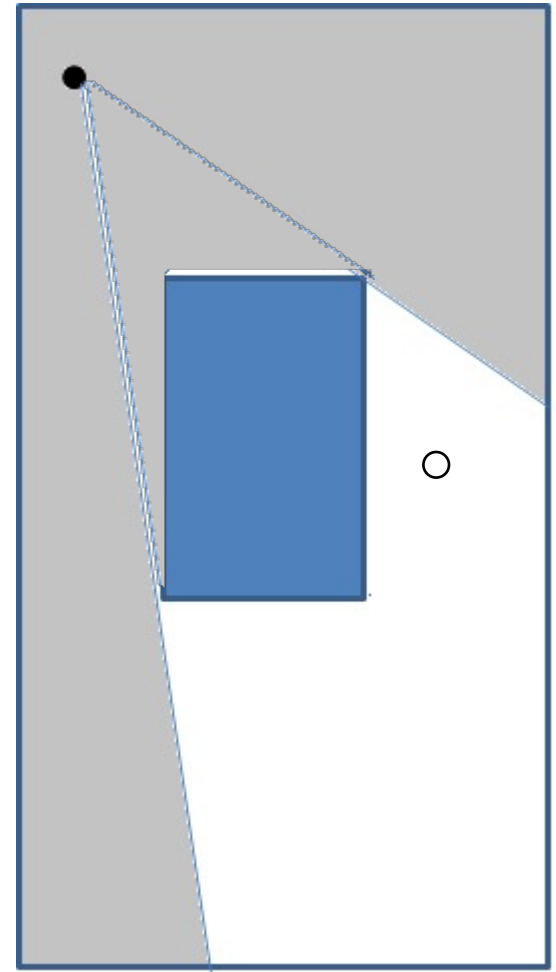
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

 Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

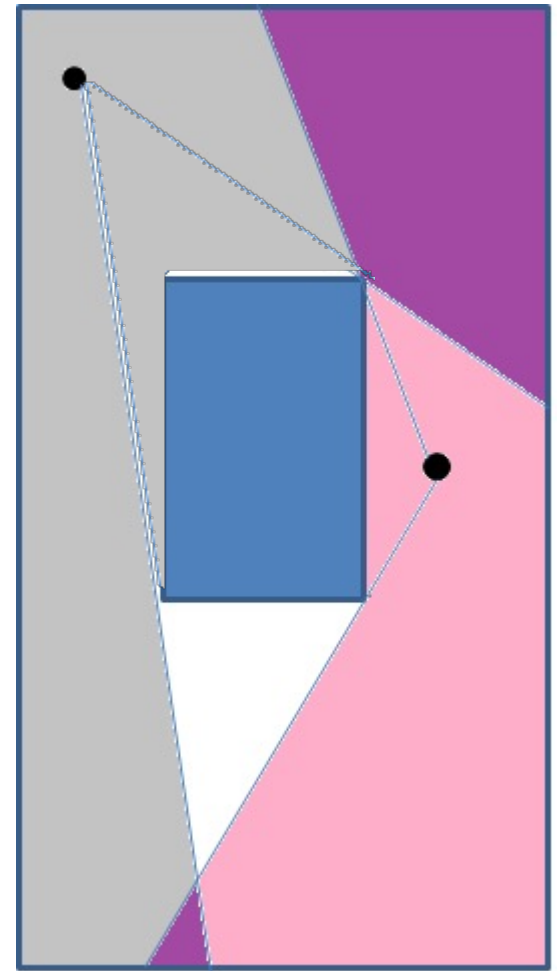
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

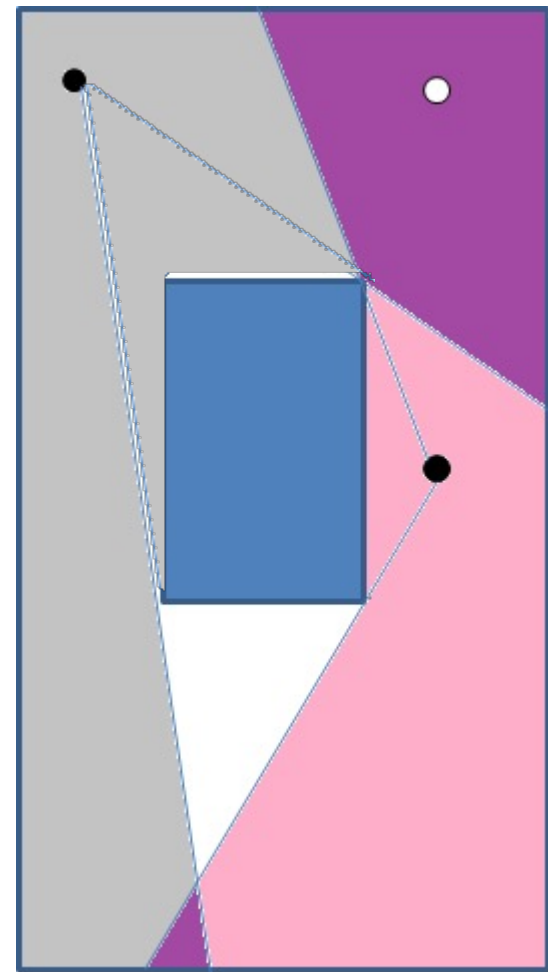
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

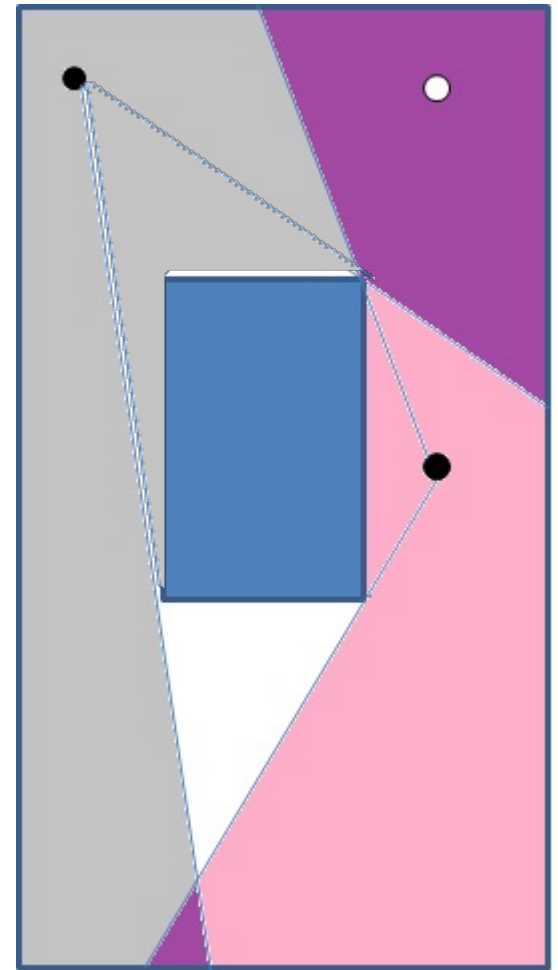
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

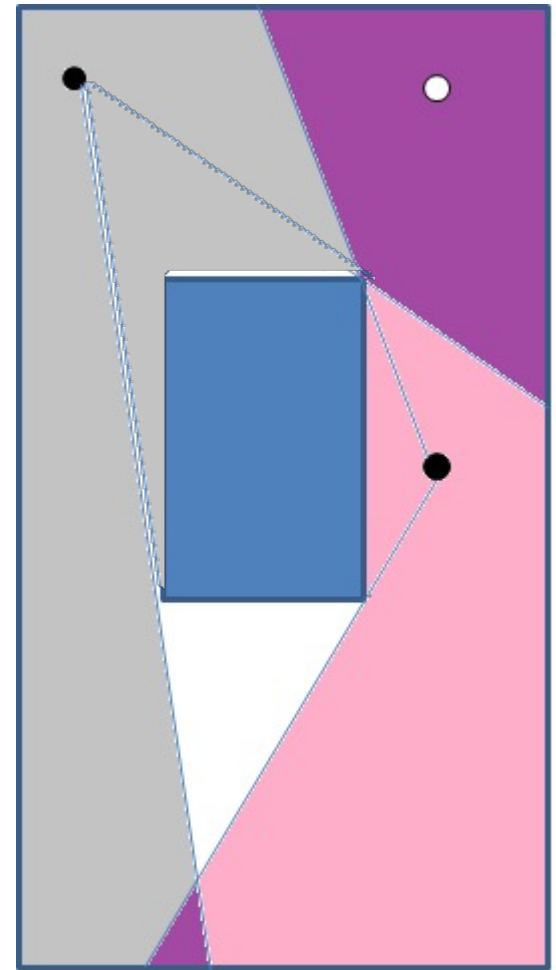
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

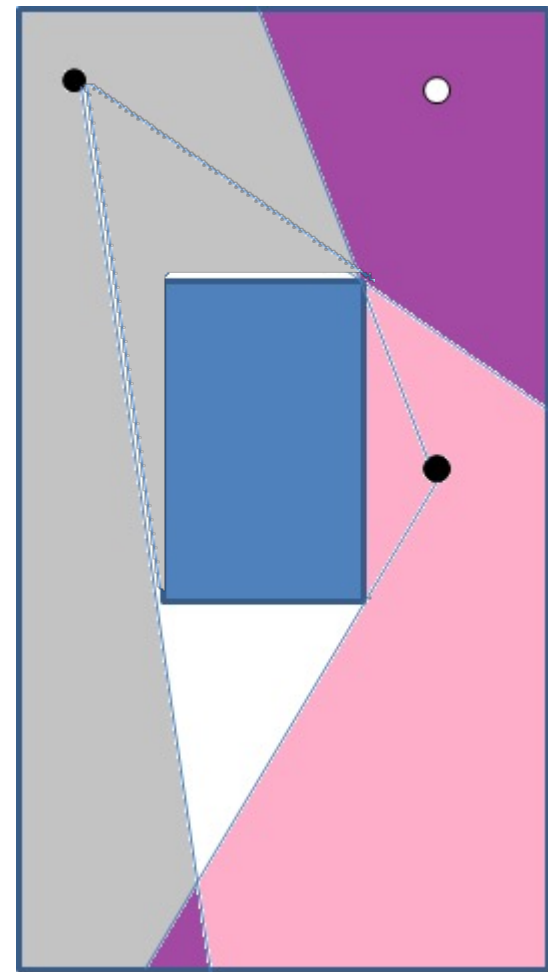
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

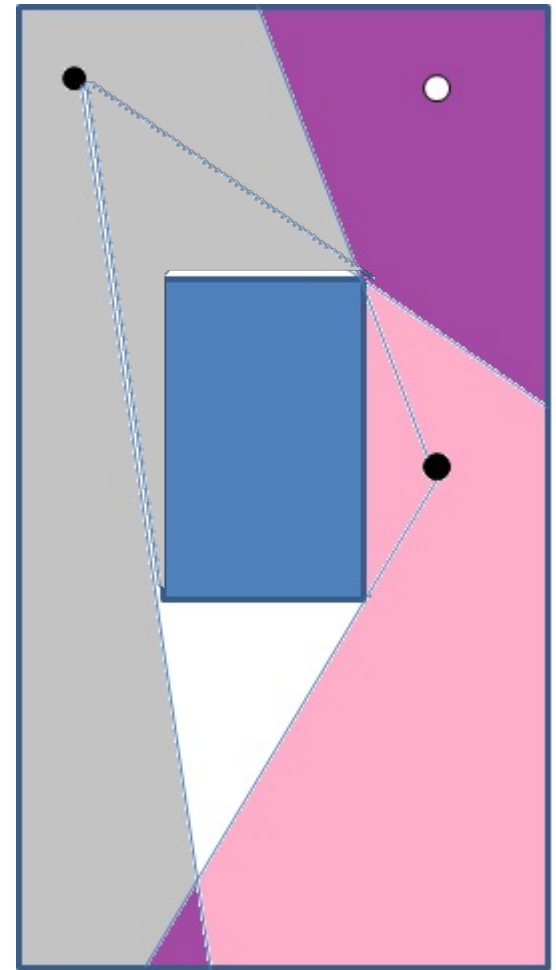
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

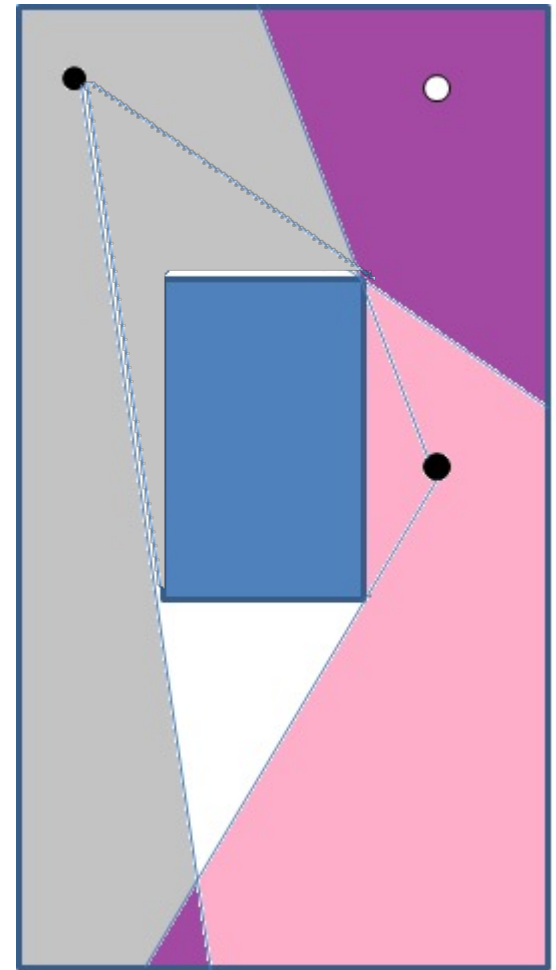
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**
We want to make the smallest change in the archive and theta that results in the max increase of QD score, the way we change the archive is theta, so we search for the smallest perturbation of theta that adds to the archive.

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $G_{vis} \leftarrow G_i$

Else /* for the smallest perturbation of theta that adds to the archive.

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

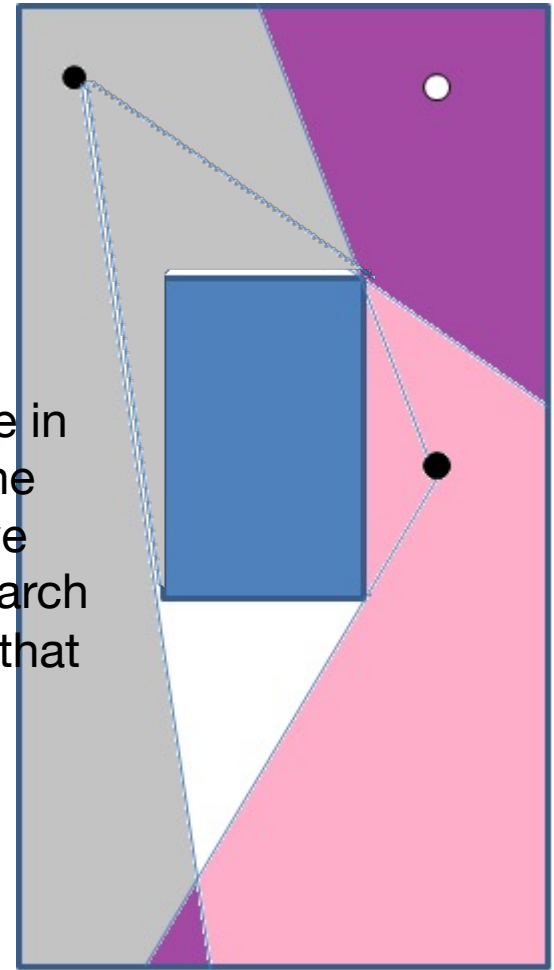
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node /*

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

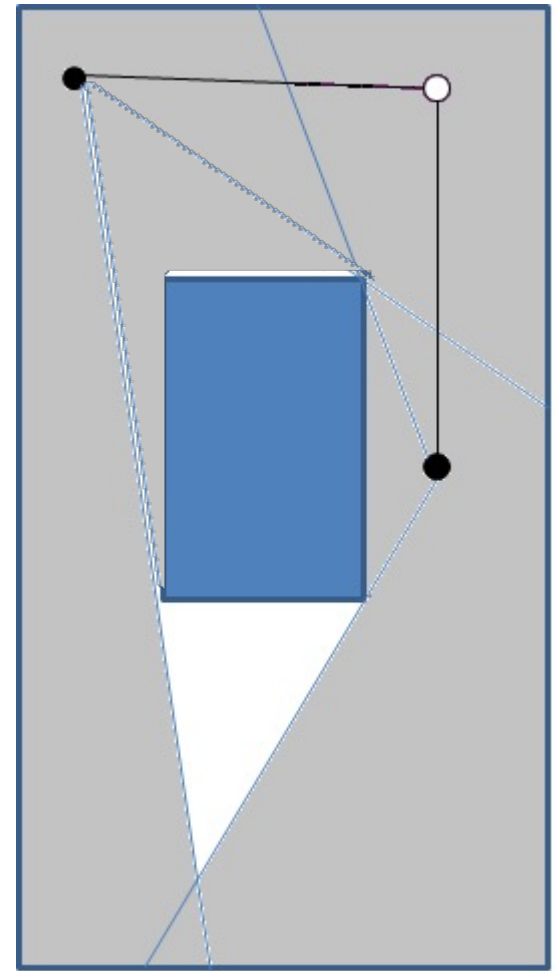
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

 Add q to $Connection$

 Create edges (q, g) and (q, g_{vis})

 Merge components G_{vis} and G_i ;

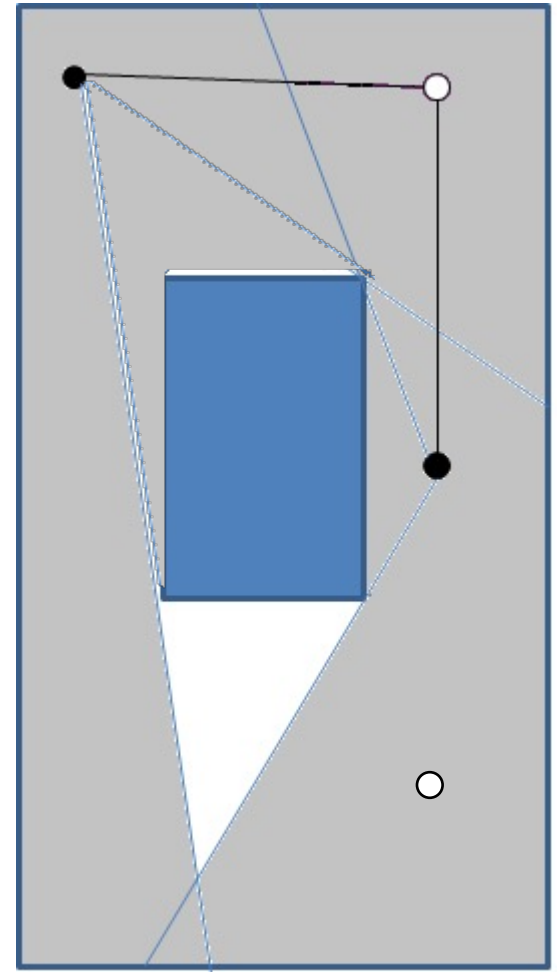
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

 Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

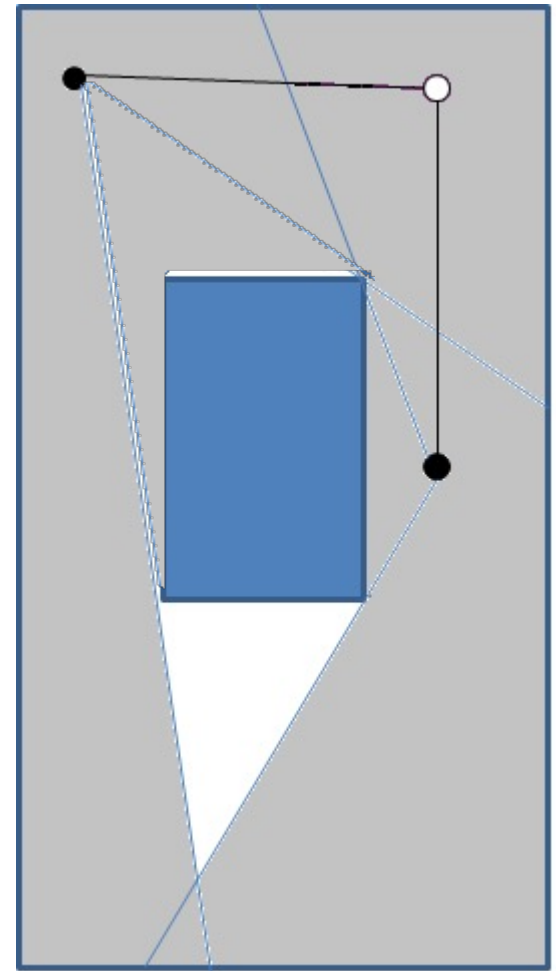
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

1d



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

 Add q to $Connection$

 Create edges (q, g) and (q, g_{vis})

 Merge components G_{vis} and G_i ;

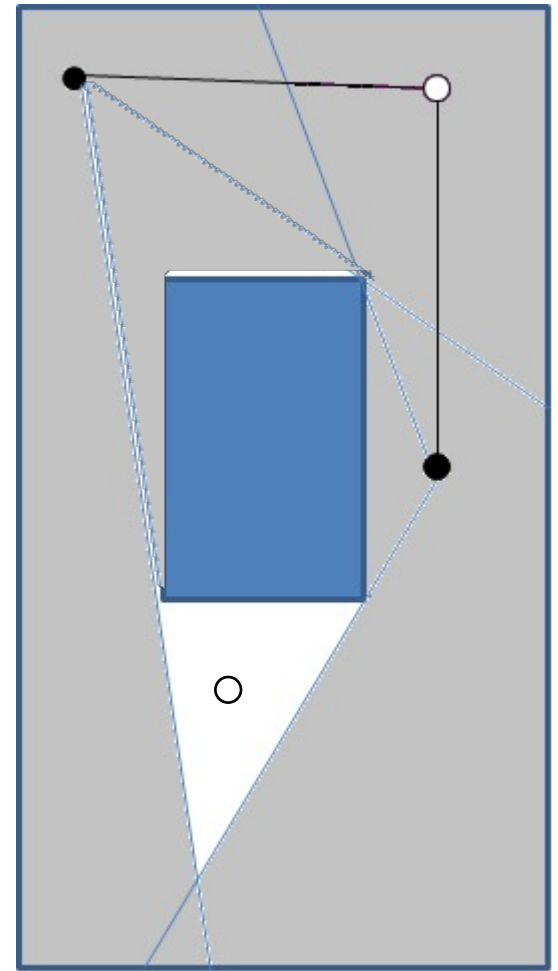
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

 Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

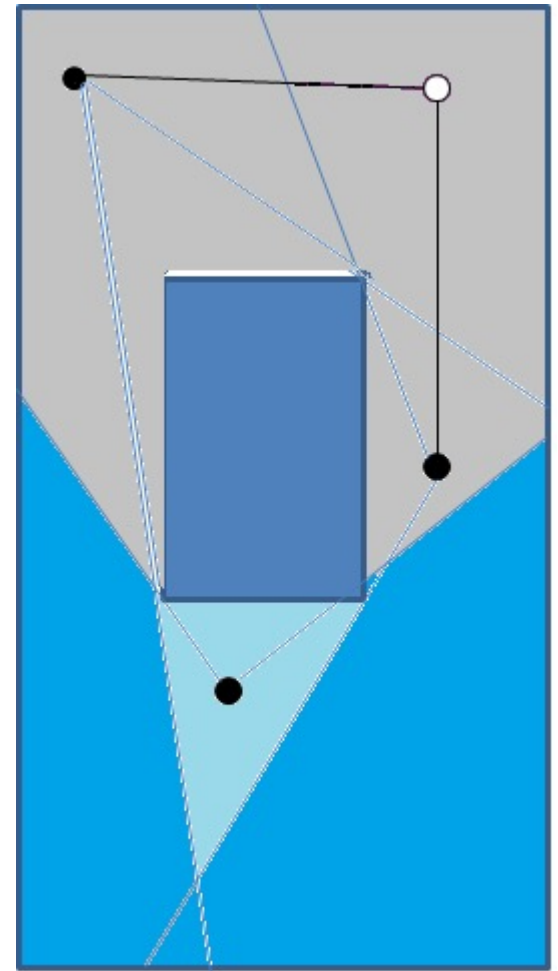
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset; Connection \leftarrow \emptyset; ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset; G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g; G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

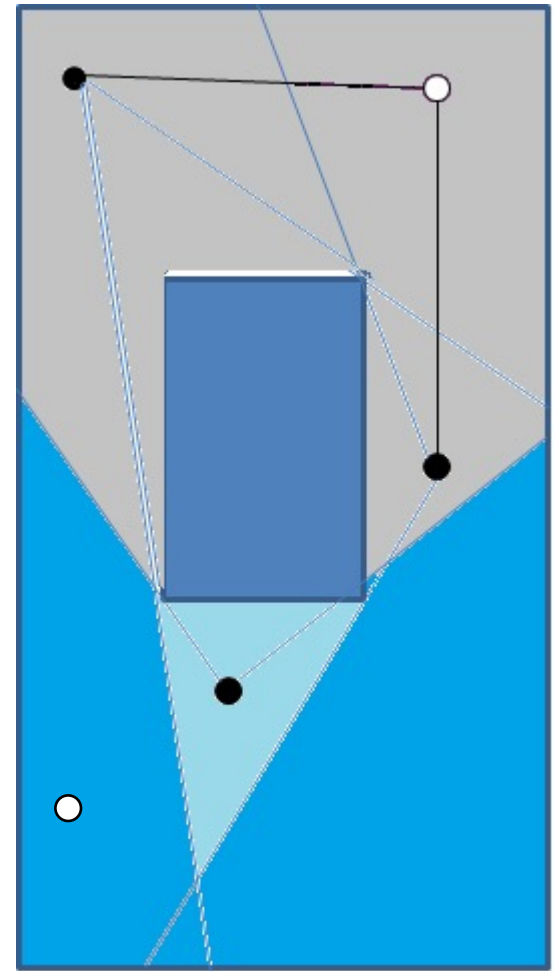
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

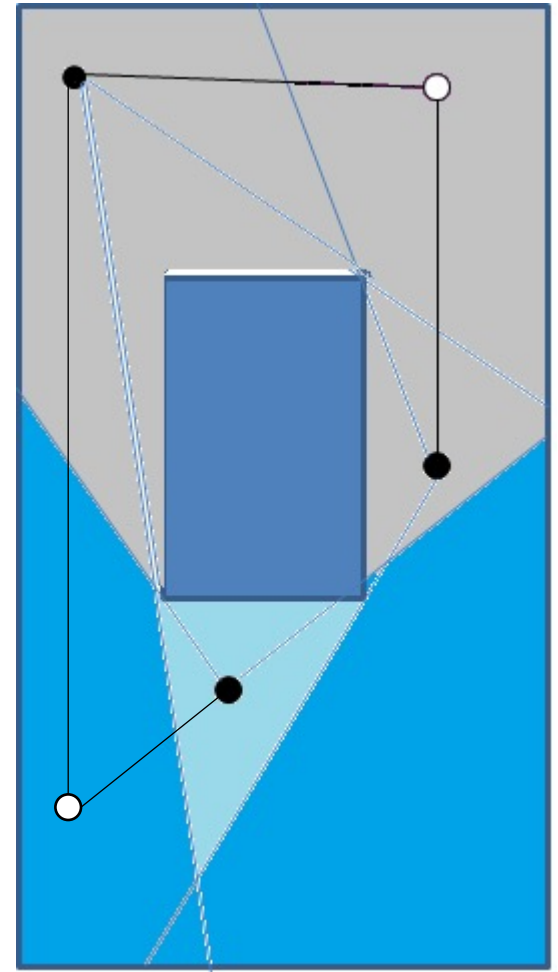
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

End



Algorithm Visib-PRM

$Guard \leftarrow \emptyset$; $Connection \leftarrow \emptyset$; $ntry \leftarrow 0$

While ($ntry < M$)

Select a random free configuration q

$g_{vis} \leftarrow \emptyset$; $G_{vis} \leftarrow \emptyset$

For all components G_i of $Guard$ **do**

$found \leftarrow FALSE$

For all nodes g of G_i **do**

If (q belongs to $Vis(g)$) **then**

$found \leftarrow TRUE$

If ($g_{vis} = \emptyset$) **then** $g_{vis} \leftarrow g$; $G_{vis} \leftarrow G_i$

Else /* q is a connection node */

Add q to $Connection$

Create edges (q, g) and (q, g_{vis})

Merge components G_{vis} and G_i ;

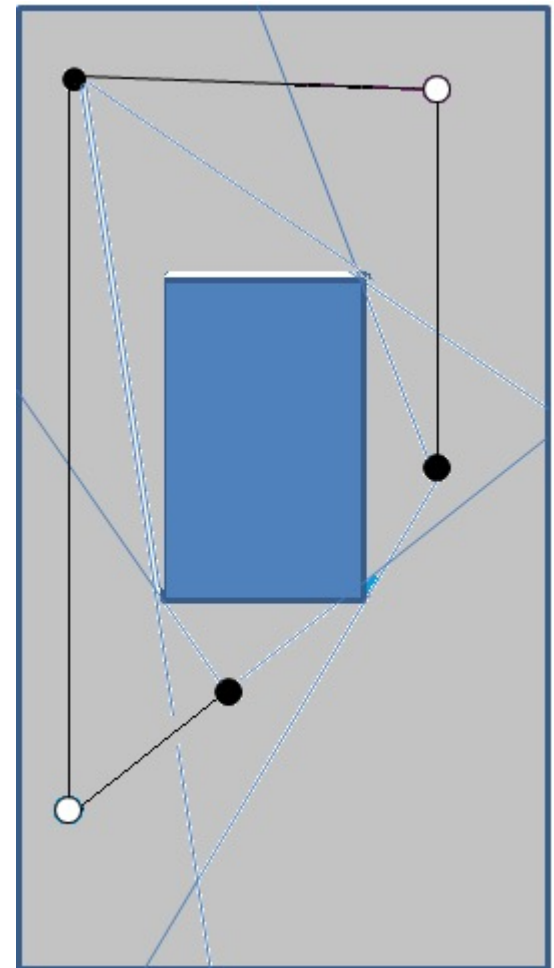
until $found = TRUE$

If ($g_{vis} = \emptyset$) **then** /* q is a guard node */

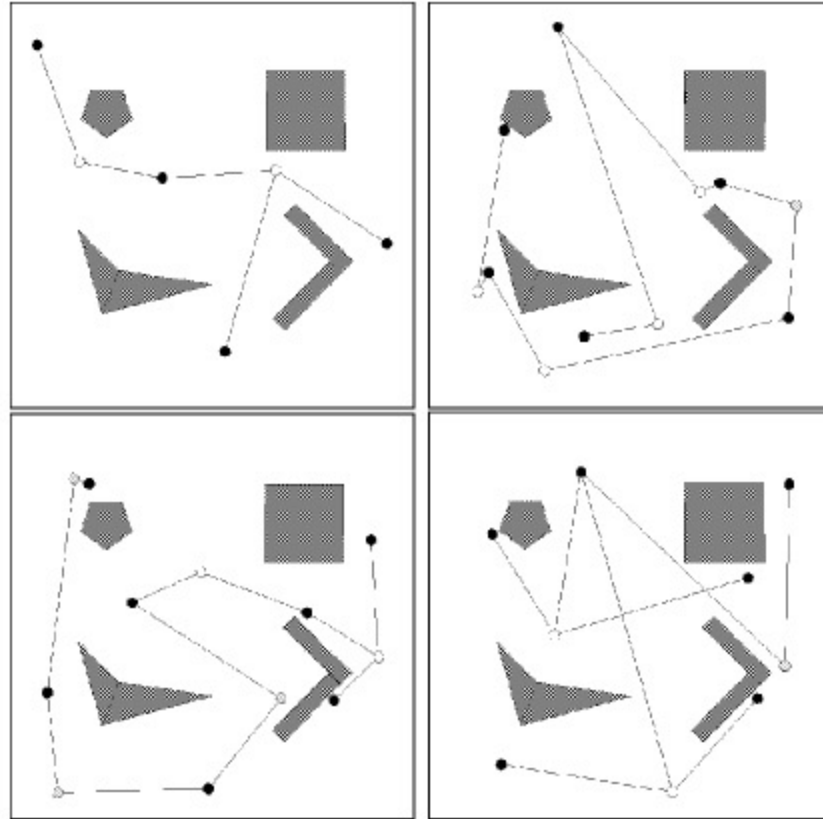
Add $\{q\}$ to $Guard$; $ntry \leftarrow 0$

Else $ntry \leftarrow ntry + 1$

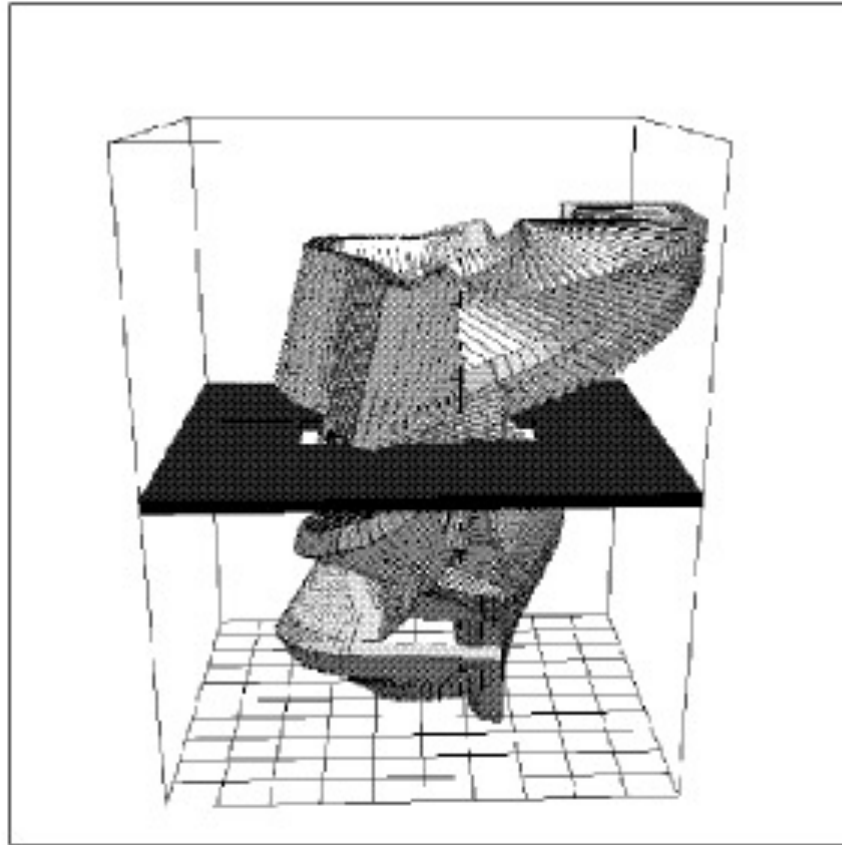
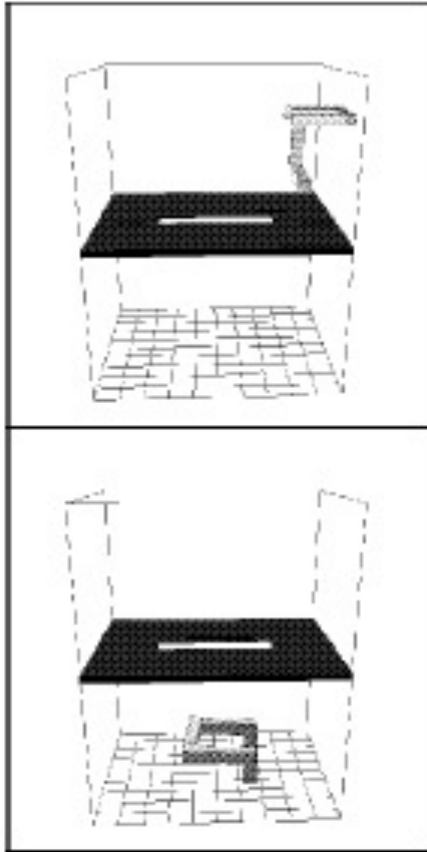
End



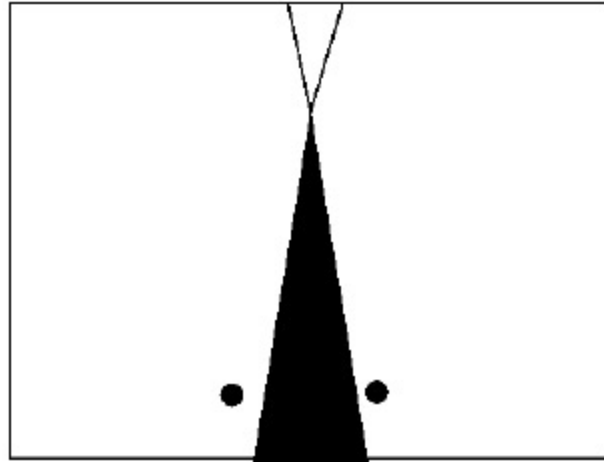
Results



Results



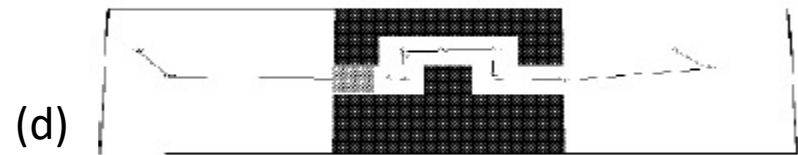
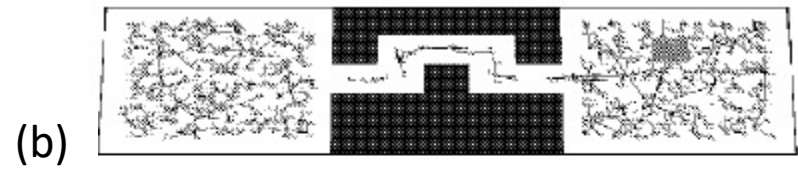
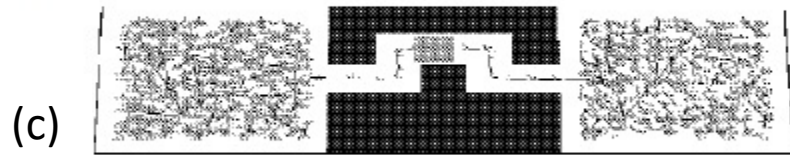
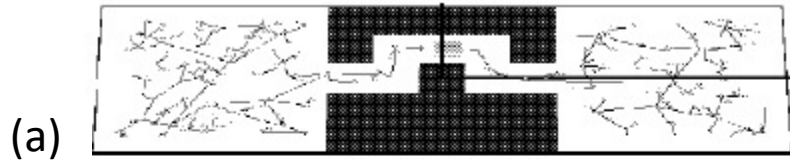
Pathologic Case



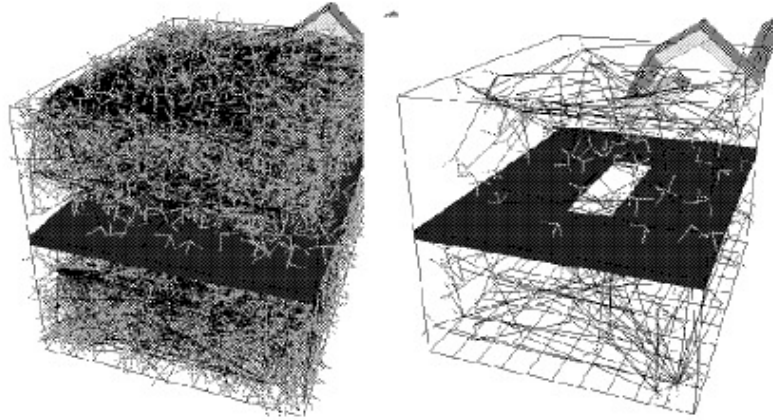
Comparison to Basic-PRM

- More expensive to generate nodes
- Cheaper to connect each node to the roadmap

Comparison to Basic-PRM



Comparison to Basic-PRM



PRM	Basic	Visibility
Roadmap size	4723	103
CS_{free} coverage	99.9%	99.7%
Random confs	14169	14369
Free confs	4723	4753
Local method #calls	700610	57622
Col. checker #calls	8.725985	1.121790
CPU time	3367 sec	281 sec

Experiments

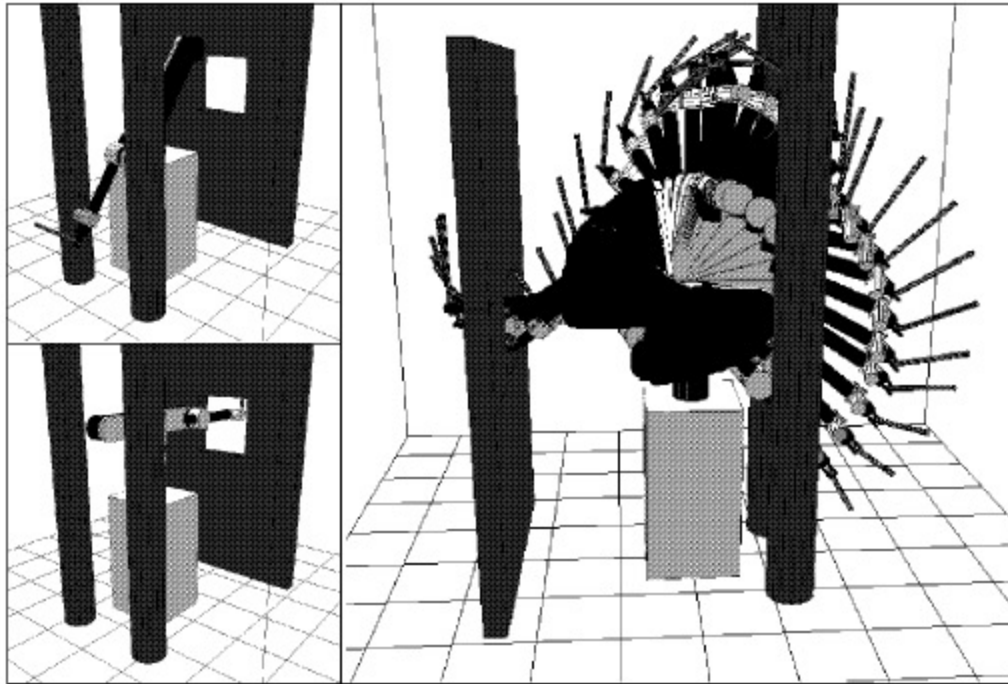


Table 1.

Mechanical system	d.o.f.	Local method	Roadmap's size	CPU time (s)
<u>Robot arm</u>	6	linear	26	370
Mobile manipulator	9	linear/ReedsShepp	75	55
Rolling bridge	4	Manhattan	25	2
Articulated hand	25	linear	70	90

Experiments

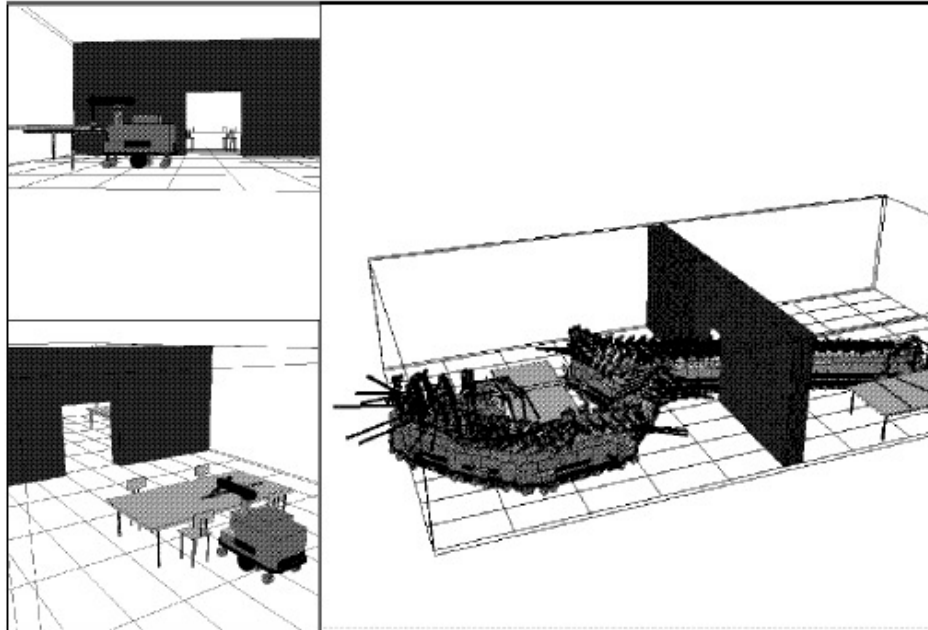


Table 1.

Mechanical system	d.o.f.	Local method	Roadmap's size	CPU time (s)
Robot arm	6	linear	26	370
<u>Mobile manipulator</u>	9	linear/ReedsShepp	75	55
Rolling bridge	4	Manhattan	25	2
Articulated hand	25	linear	70	90

Experiments

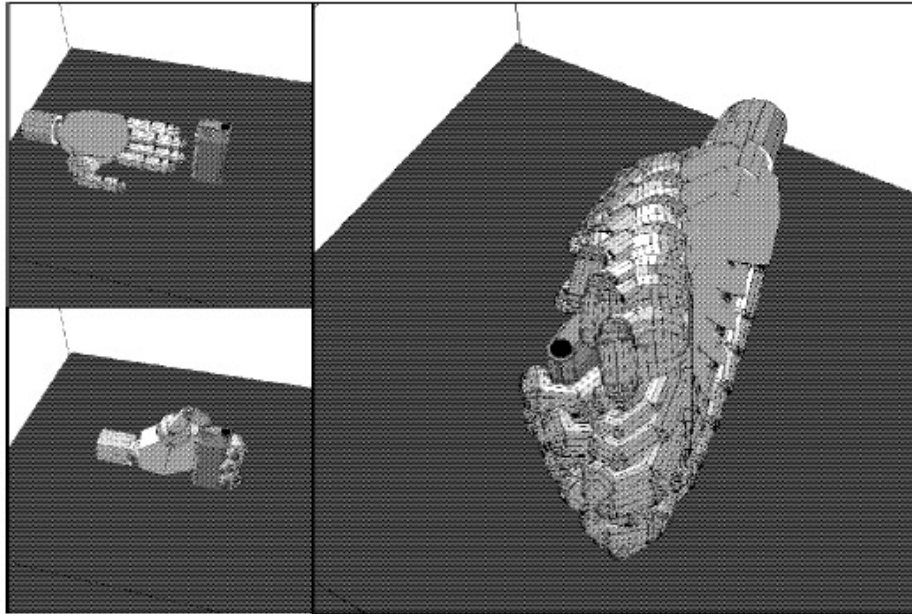


Table 1.

Mechanical system	d.o.f.	Local method	Roadmap's size	CPU time (s)
Robot arm	6	linear	26	370
Mobile manipulator	9	linear/ReedsShepp	75	55
Rolling bridge	4	Manhattan	25	2
<u>Articulated hand</u>	25	linear	70	90