4-4 샘플링 기법 3:
Lazy-Probabilistic Roadmap (Lazy-PRM)

강의 요약

01

Obstacle-Based Probabilistic Roadmap (OB-PRM)

- PRM 과 샘플링 기법에서 차이
- C-obstacle 근처에 샘플링

02

주요 특징

- Multi-query
- Narrow Passage :)
- 최적 경로 보장 X
- Probabilistically Complete
- 충돌검출 횟수 비교적 많음

03

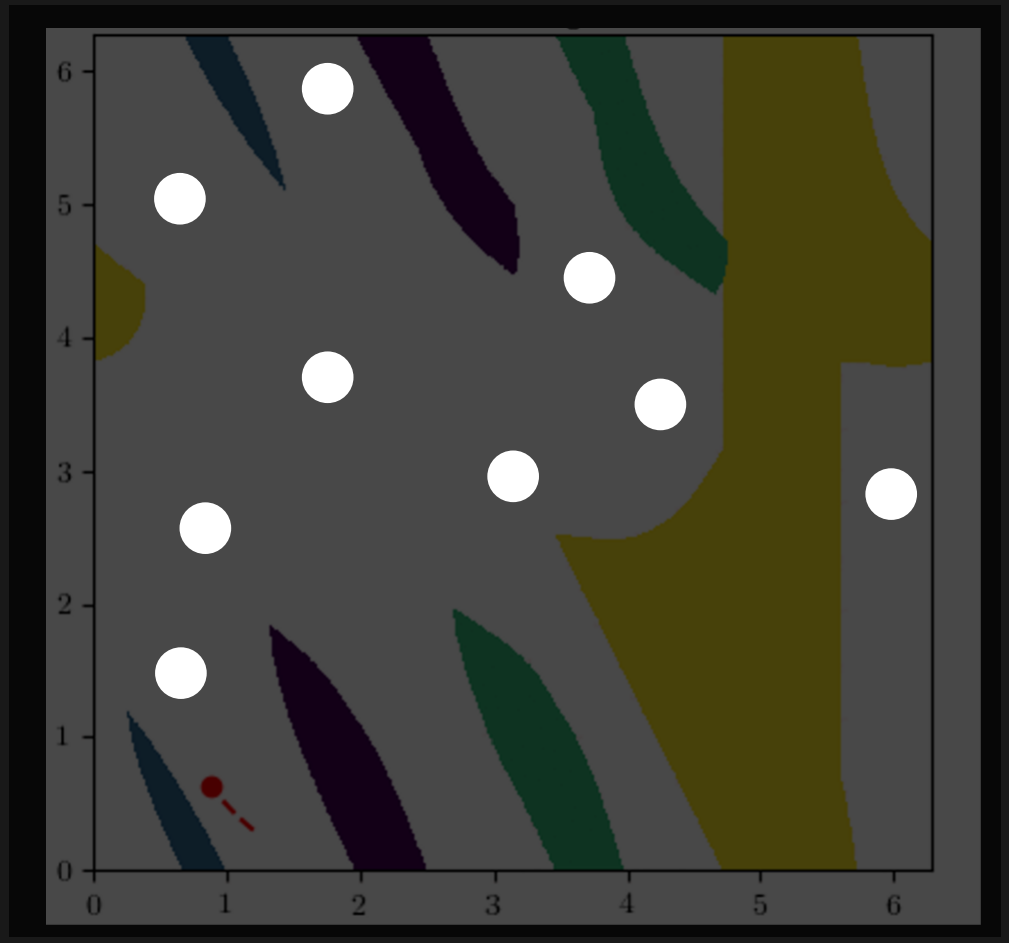
알고리즘

04

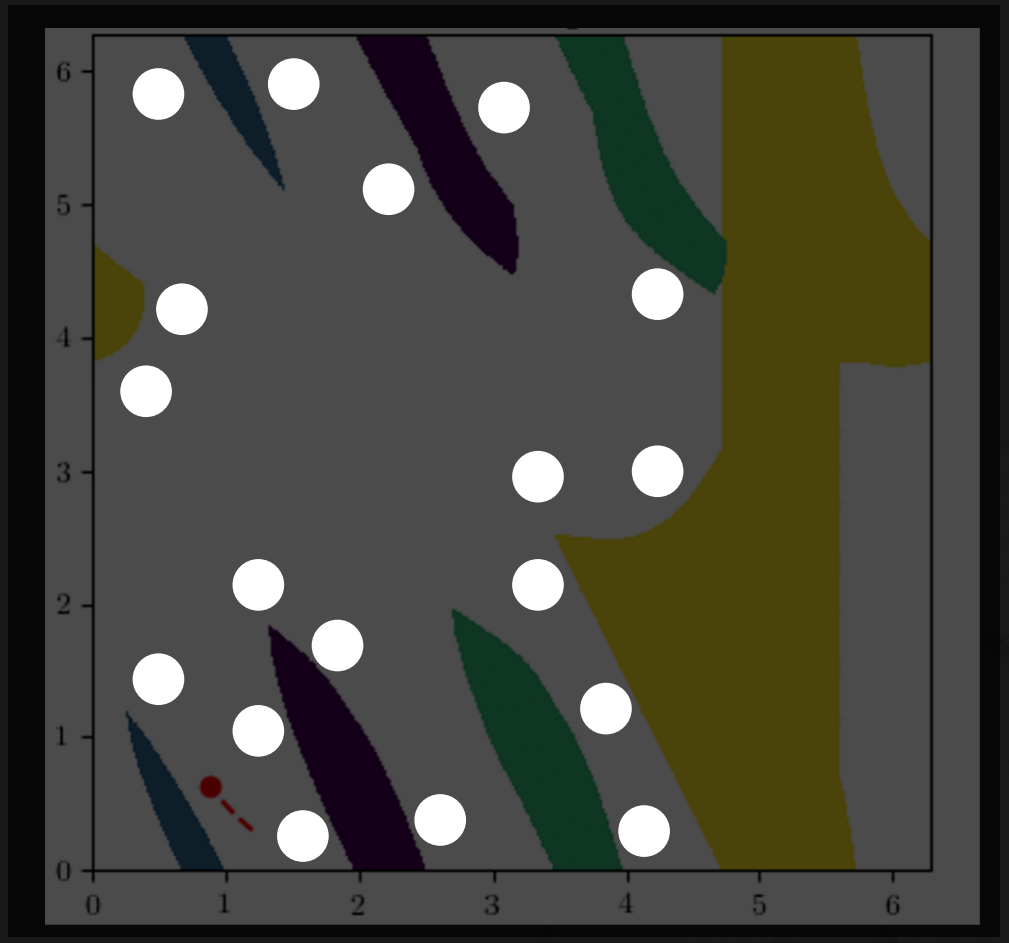
코드 분석

PRM vs. OB-PRM

PRM

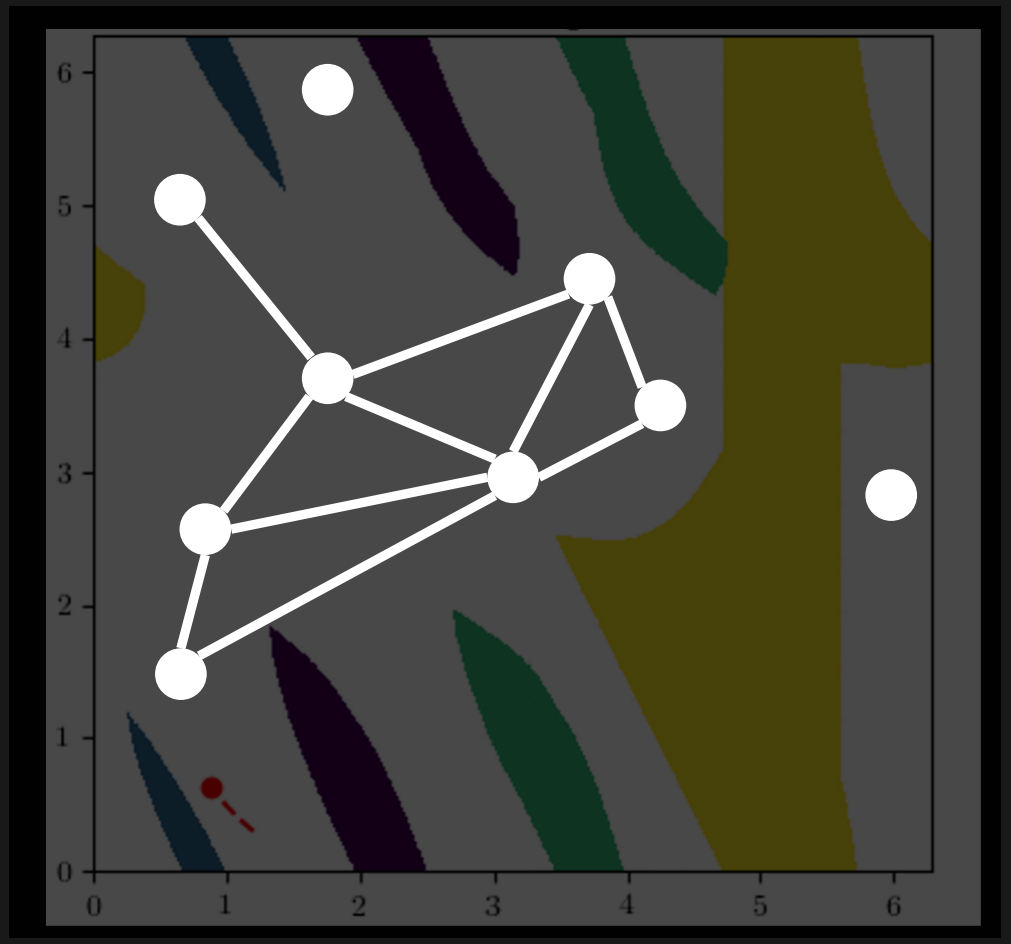


OB-PRM

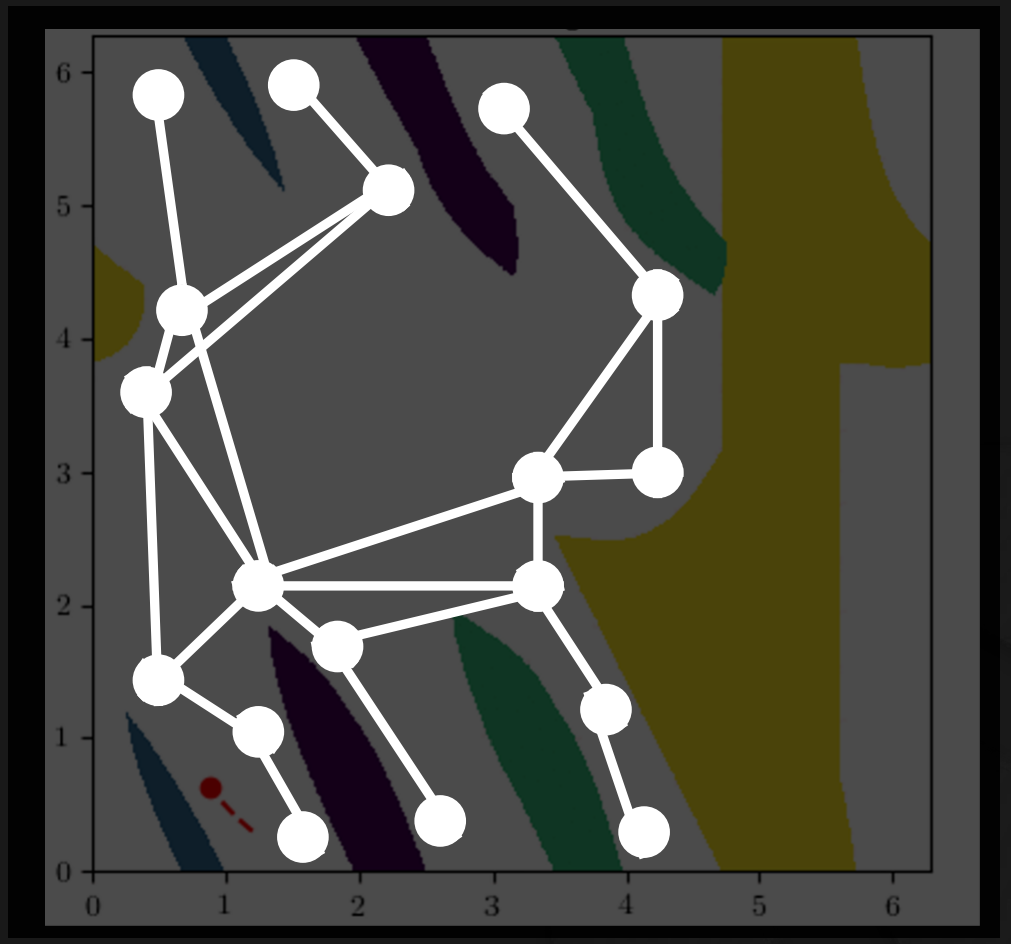


PRM vs. OB-PRM

PRM



OB-PRM



Obstacle-Based Probabilistic Roadmap (OB-PRM)

Algorithm 1 Probabilistic Roadmap (PRM)

Require: Number of samples N, number of neighbors k, start q_{start}, goal q_{goal}
Ensure: A path from q_{start} to q_{goal}, if one exists

Initialize roadmap graph $G = (V, E) \leftarrow \emptyset$

while $|V| < N$ **do**
    Sample a random configuration $q \in \mathcal{C}$
    if $q \in \mathcal{C}_{\text{free}}$ **then**
        $V \leftarrow V \cup \{q\}$
    end if
end while
for all $q \in V$ **do**
    Find k-nearest neighbors $\text{NN}_k(q) \subset V$
    for all $q_{\text{near}} \in \text{NN}_k(q)$ **do**
        if LocalPlanner(q, q_{near}) is collision-free **then**
            $E \leftarrow E \cup \{(q, q_{\text{near}})\}$
        end if
    end for
end for

if $q_{\text{start}}, q_{\text{goal}} \in \mathcal{C}_{\text{free}}$ **then**
    $V \leftarrow V \cup \{q_{\text{start}}, q_{\text{goal}}\}$
    Connect $q_{\text{start}}, q_{\text{goal}}$ to k-nearest neighbors with collision-free edges
    Use graph search (e.g., Dijkstra or A*) to find path from q_{start} to q_{goal}
else
    return No valid path (start or goal in collision)
end if

Algorithm 2 Obstacle-Based PRM (OBPRM)

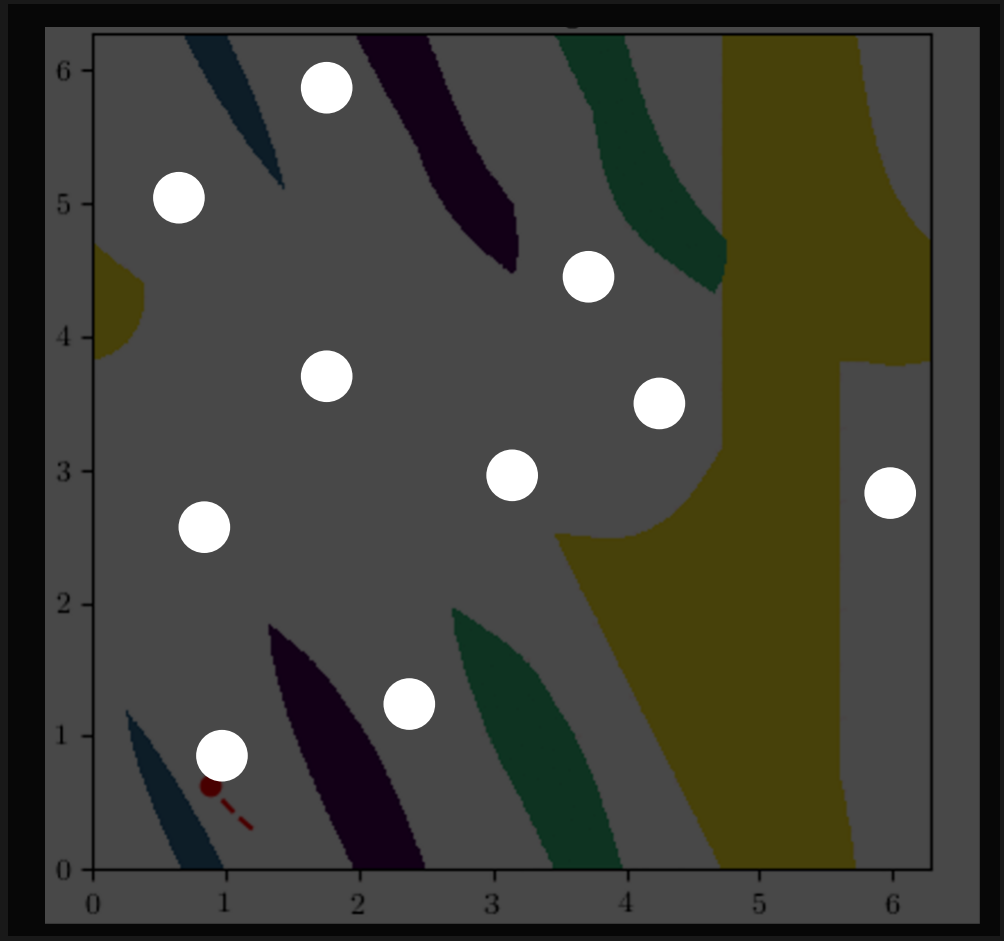
Require: Number of samples N, number of neighbors k
Ensure: Roadmap graph $G = (V, E)$

Initialize roadmap $G = (V, E) \leftarrow \emptyset$

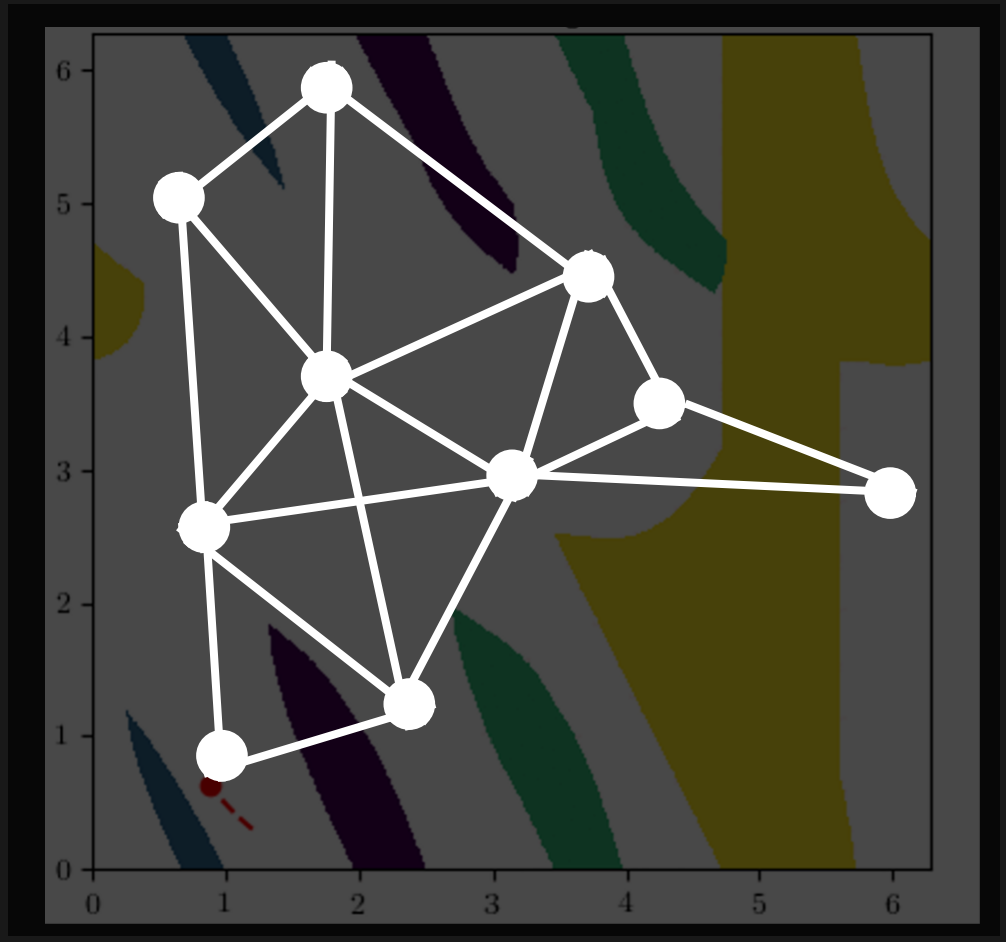
while $|V| < N$ **do**
    Sample a random configuration $q \in \mathcal{C}$
    if $q \in \mathcal{C}_{\text{obs}}$ **then**
        Generate random direction d
        for $i = 1$ to m **do**
            $q' \leftarrow q + \epsilon_i d$
            if $q' \in \mathcal{C}_{\text{free}}$ **then**
                $V \leftarrow V \cup \{q'\}$
                break
            end if
        end for
    end if
end while
for all $q \in V$ **do**
    Find k-nearest neighbors $\text{NN}_k(q) \subset V$
    for all $q_{\text{near}} \in \text{NN}_k(q)$ **do**
        if LocalPlanner(q, q_{near}) is collision-free **then**
            $E \leftarrow E \cup \{(q, q_{\text{near}})\}$
        end if
    end for
end for

if $q_{\text{start}}, q_{\text{goal}} \in \mathcal{C}_{\text{free}}$ **then**
    $V \leftarrow V \cup \{q_{\text{start}}, q_{\text{goal}}\}$
    Connect $q_{\text{start}}, q_{\text{goal}}$ to k-nearest neighbors with collision-free edges
    Use graph search to find path from q_{start} to q_{goal}
else
    return No valid path (start or goal in collision)
end if

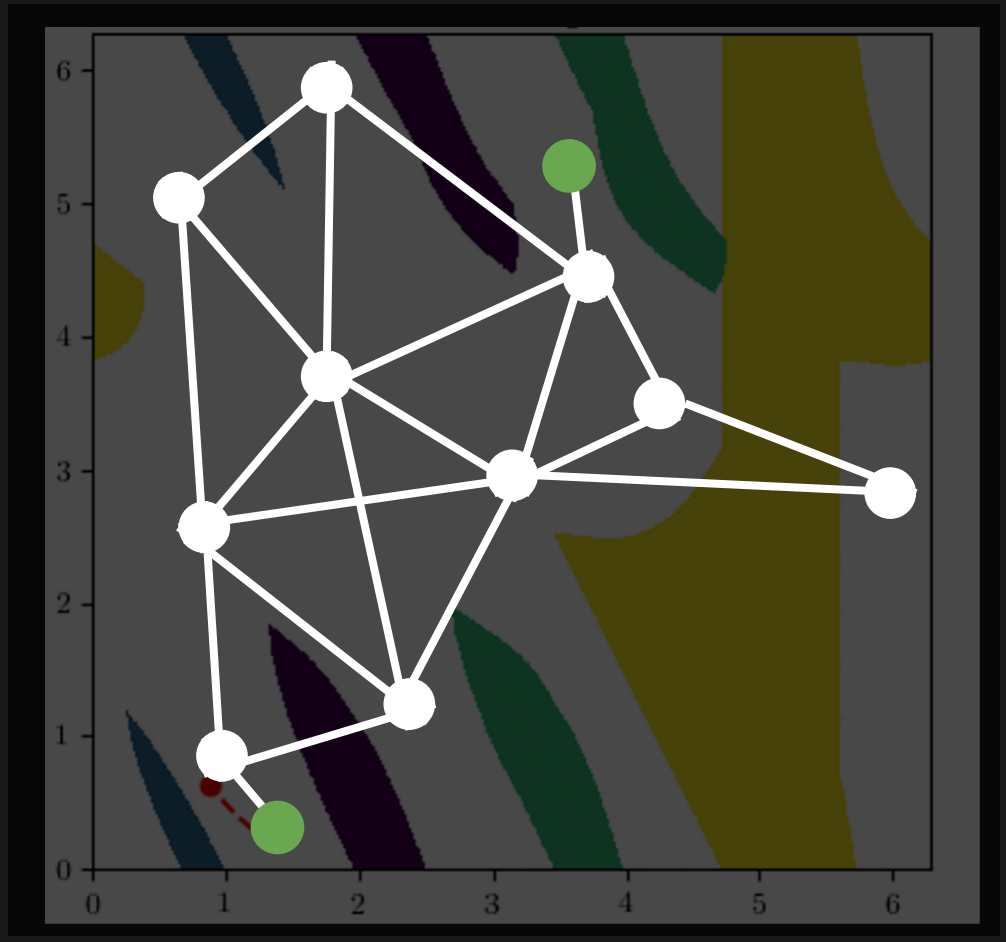
Lazy-Probabilistic Roadmap (Lazy-PRM)



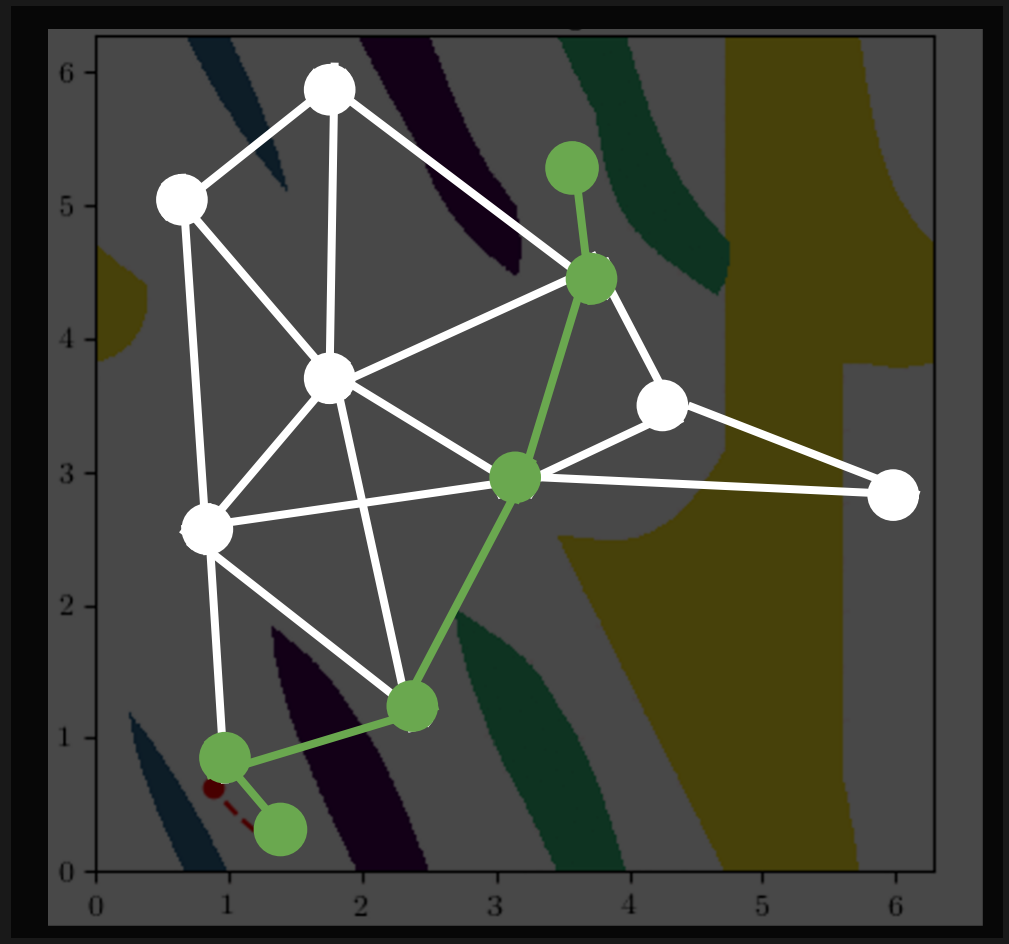
Lazy-Probabilistic Roadmap (Lazy-PRM)



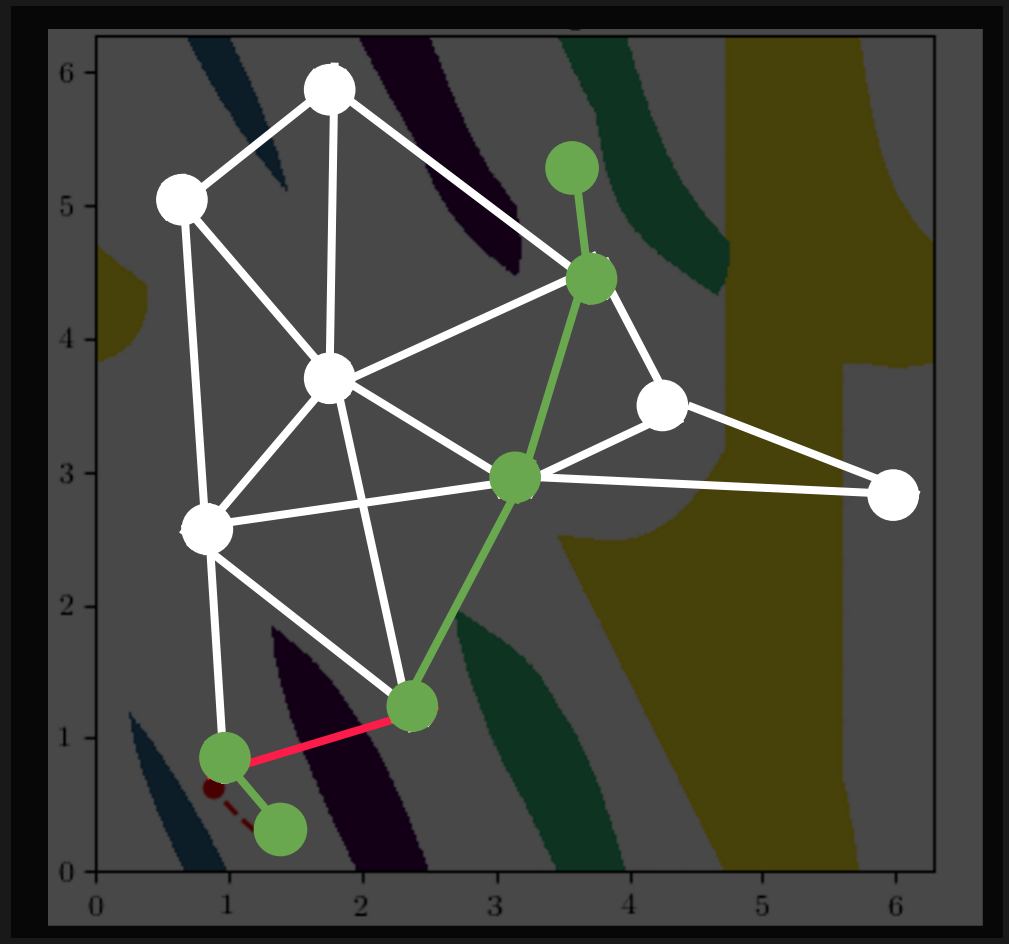
Lazy-Probabilistic Roadmap (Lazy-PRM)



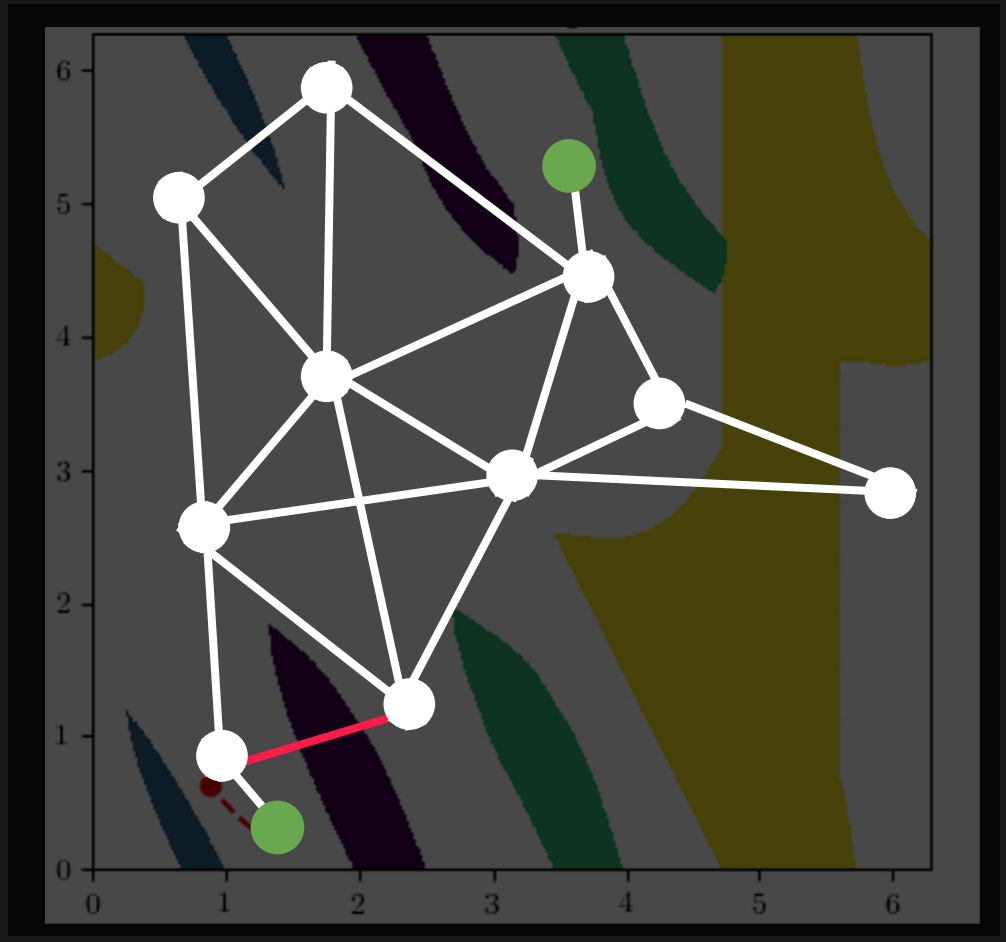
Lazy-Probabilistic Roadmap (Lazy-PRM)



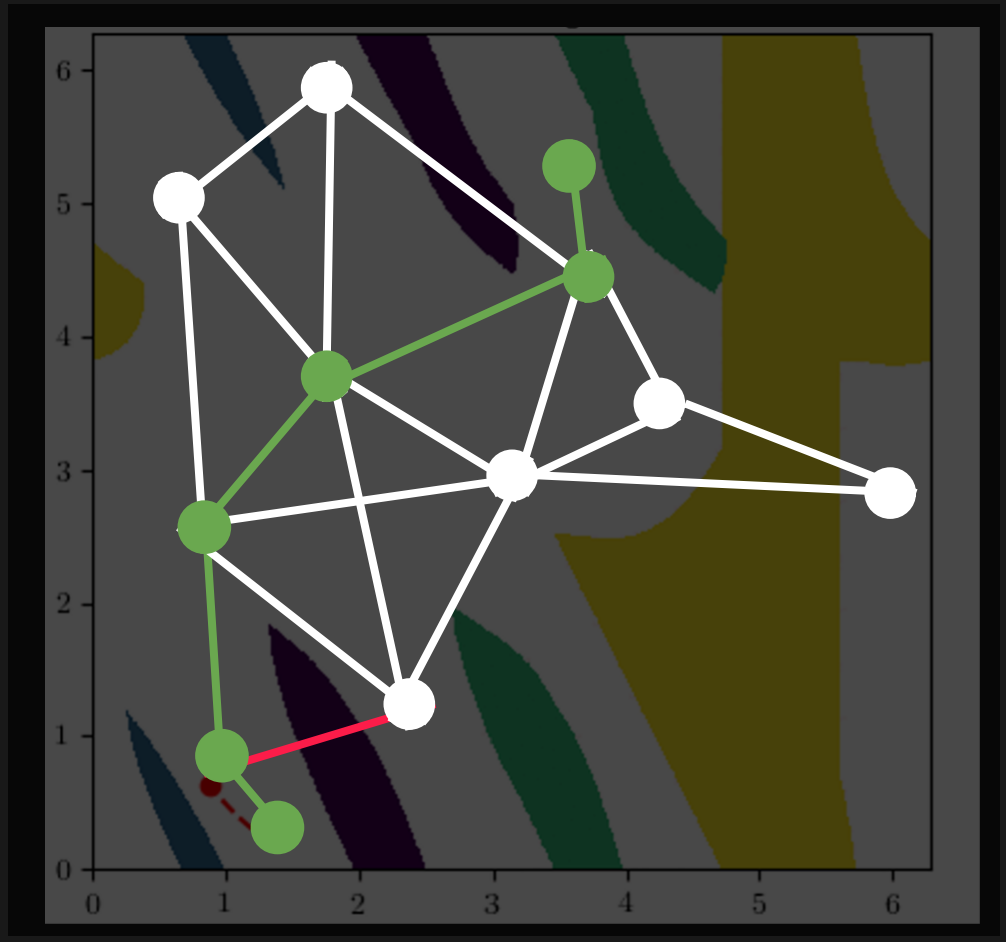
Lazy-Probabilistic Roadmap (Lazy-PRM)



Lazy-Probabilistic Roadmap (Lazy-PRM)



Lazy-Probabilistic Roadmap (Lazy-PRM)



Lazy-Probabilistic Roadmap (Lazy-PRM)

Algorithm 1 Probabilistic Roadmap (PRM)

Require: Number of samples N, number of neighbors k, start q_{start}, goal q_{goal}
Ensure: A path from q_{start} to q_{goal}, if one exists

Initialize roadmap graph $G = (V, E) \leftarrow \emptyset$

while $|V| < N$ **do**
    Sample a random configuration $q \in \mathcal{C}$
    if $q \in \mathcal{C}_{\text{free}}$ **then**
        $V \leftarrow V \cup \{q\}$
    end if
end while
for all $q \in V$ **do**
    Find k-nearest neighbors $\text{NN}_k(q) \subset V$
    for all $q_{\text{near}} \in \text{NN}_k(q)$ **do**
        if LocalPlanner(q, q_{near}) is collision-free **then**
            $E \leftarrow E \cup \{(q, q_{\text{near}})\}$
        end if
    end for
end for

if $q_{\text{start}}, q_{\text{goal}} \in \mathcal{C}_{\text{free}}$ **then**
    $V \leftarrow V \cup \{q_{\text{start}}, q_{\text{goal}}\}$
    Connect $q_{\text{start}}, q_{\text{goal}}$ to k-nearest neighbors with collision-free edges
    Use graph search (e.g., Dijkstra or A*) to find path from q_{start} to q_{goal}
else
    return No valid path (start or goal in collision)
end if

Algorithm 3 Lazy Probabilistic Roadmap (Lazy PRM)

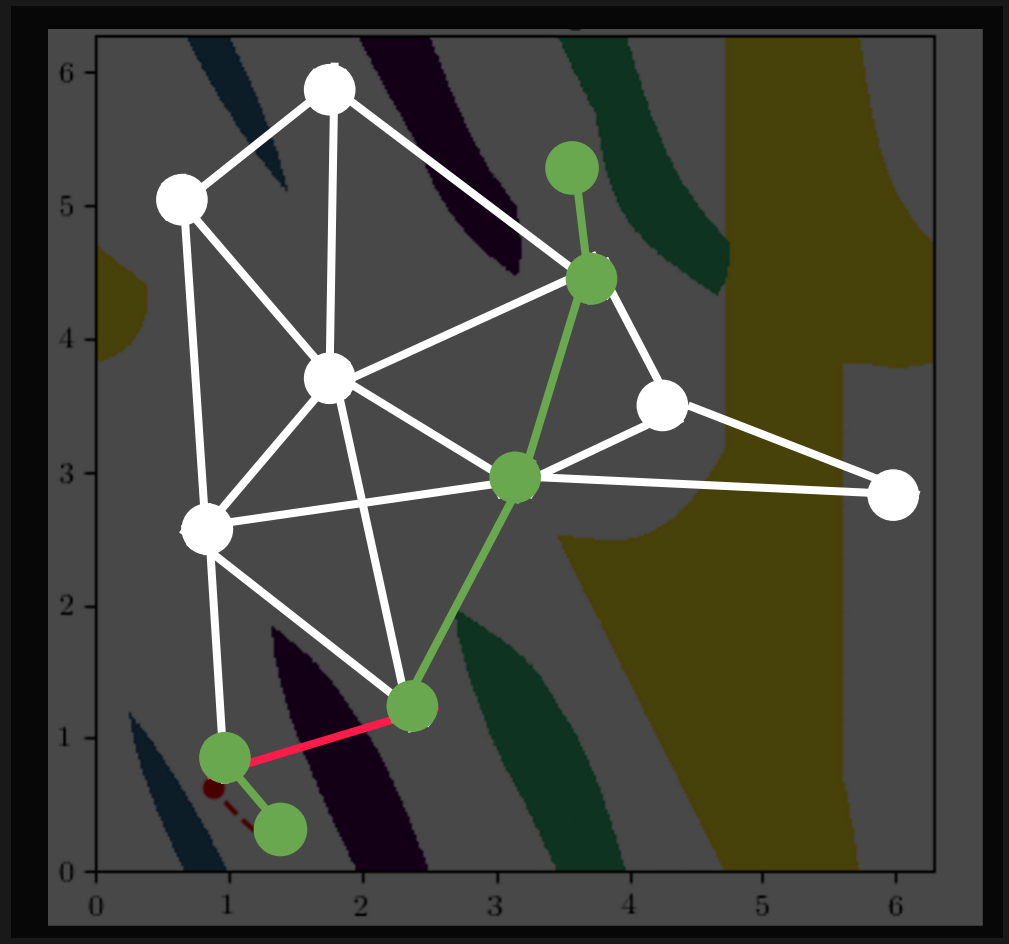
Require: Number of samples N, number of neighbors k, start q_{start}, goal q_{goal}
Ensure: A collision-free path from q_{start} to q_{goal}, if one exists

Initialize roadmap graph $G = (V, E) \leftarrow \emptyset$

while $|V| < N$ **do**
    Sample a random configuration $q \in \mathcal{C}$
    if $q \in \mathcal{C}_{\text{free}}$ **then**
        $V \leftarrow V \cup \{q\}$
    end if
end while
for all $q \in V$ **do**
    Find k-nearest neighbors $\text{NN}_k(q) \subset V$
    for all $q_{\text{near}} \in \text{NN}_k(q)$ **do**
        Add edge (q, q_{near}) to E without collision checking
    end for
end for

if $q_{\text{start}}, q_{\text{goal}} \in \mathcal{C}_{\text{free}}$ **then**
    Add $q_{\text{start}}, q_{\text{goal}}$ to V
    Connect them to k-nearest neighbors (without collision checking)
    while true **do**
        Find a path π from q_{start} to q_{goal} in G
        if no such path exists **then**
            return No valid path exists
        end if
        Check all edges in π for collisions
        if all edges are collision-free **then**
            return π
        else
            Remove all invalid edges from G
        end if
    end while
else
    return No valid path (start or goal in collision)
end if

Lazy-Probabilistic Roadmap (Lazy-PRM)



PRM vs. Lazy-PRM

PRM

- 충돌 검출 함수를 모든 edge에 대하여 사용
- 불필요한 충돌 검출로 인한 연산량 증가

- Probabilistic Completeness
- Narrow Passage Problem
- 샘플링 기법에 따라 성능이 달라짐

PRM
vs.
Lazy-PRM

Lazy-PRM

- 충돌 검출 함수를 특정 edge에만 사용
- 불필요한 충돌 검출을 피함

- Probabilistic Completeness
- Narrow Passage Problem
- 샘플링 기법에 따라 성능이 달라짐

강의 요약

01

PRM

- 불필요한 충돌 검출 함수 사용이 많음

02

Lazy-PRM

- 불필요한 충돌 검출 함수 사용량을 줄임

03

알고리즘

04

코드 분석