4-12 최적화 기법 개요

강의 요약

01

샘플링 기법 정리

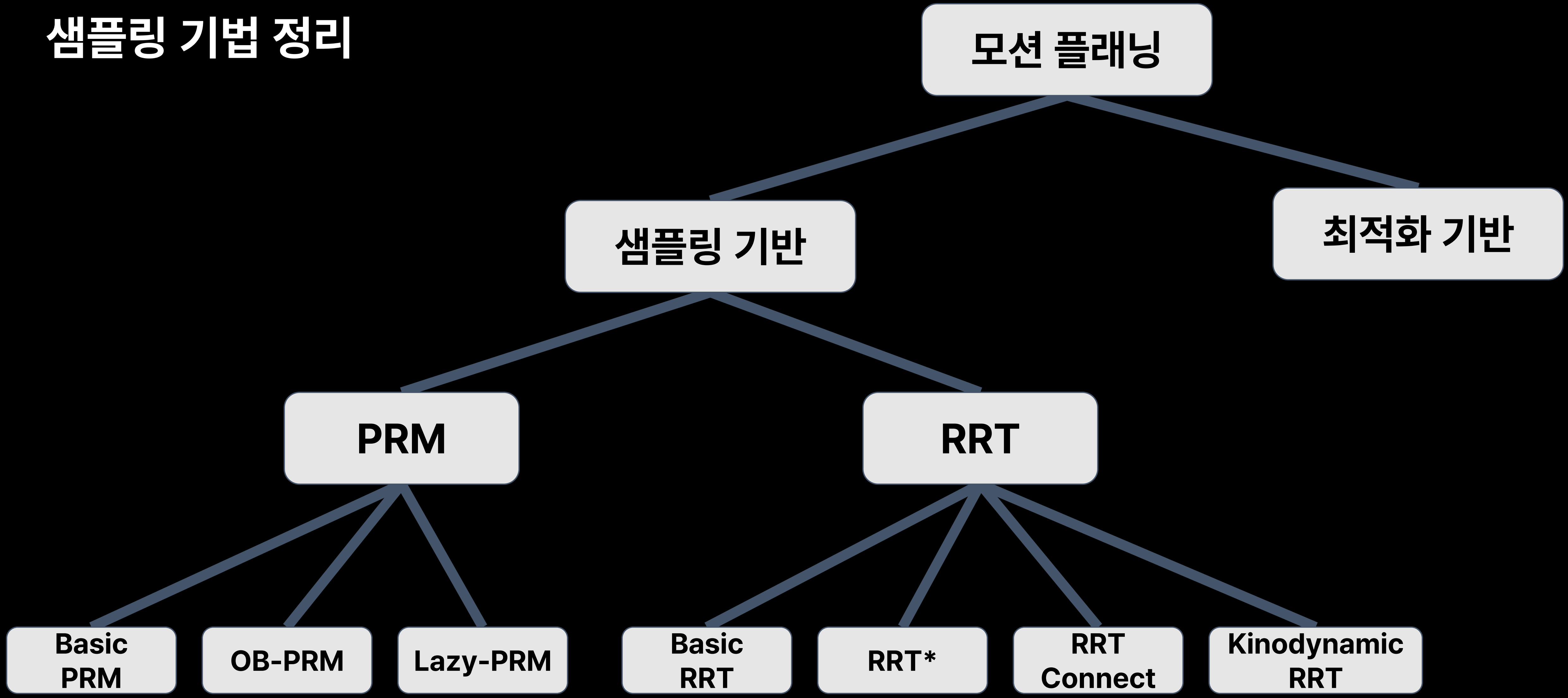
- PRM
- RRT

02

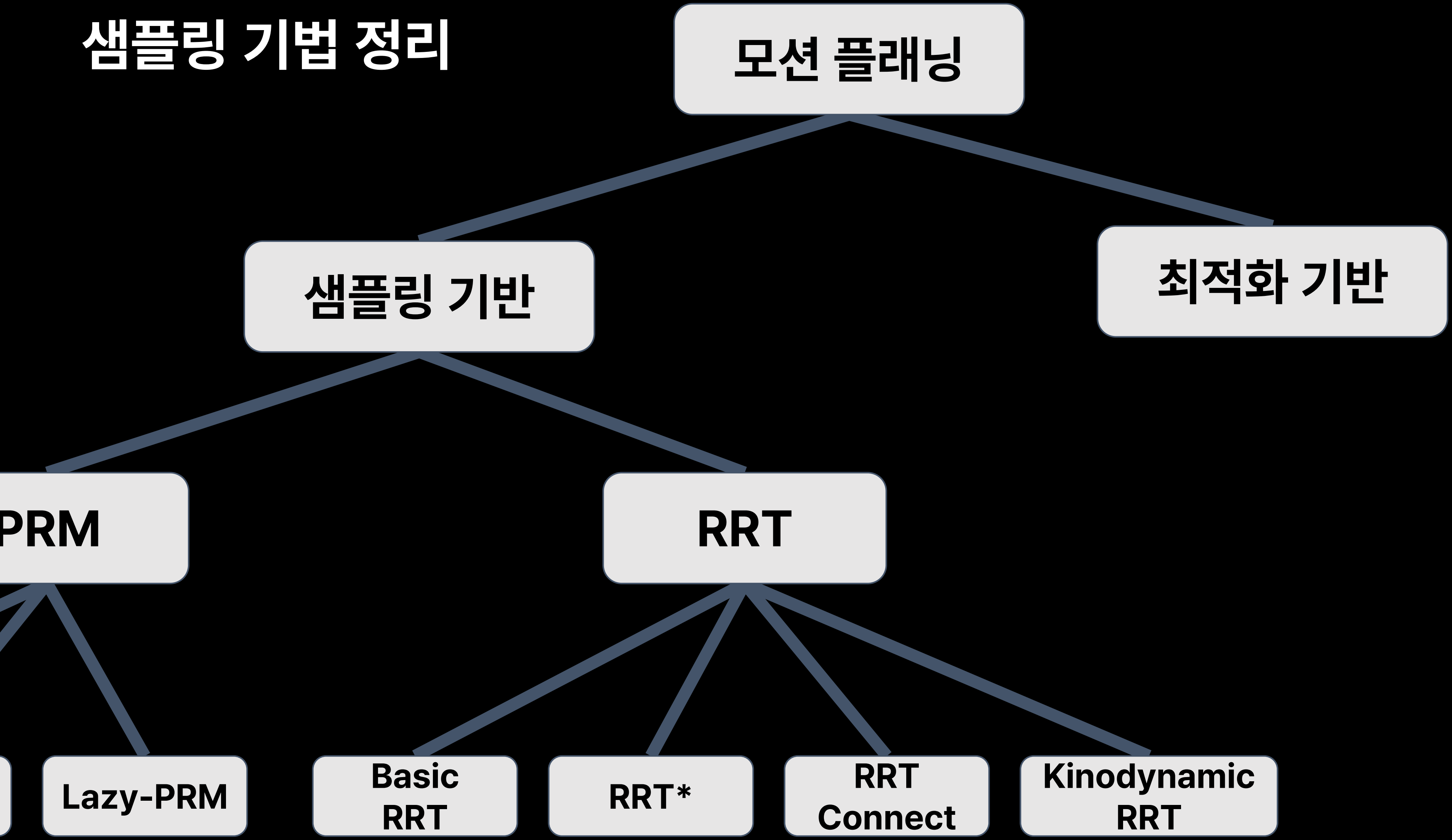
솔루션 퀄리티 향상 기법

- Shortcutting
- 최적화 기법 (다음 강의)

샘플링 기법 정리

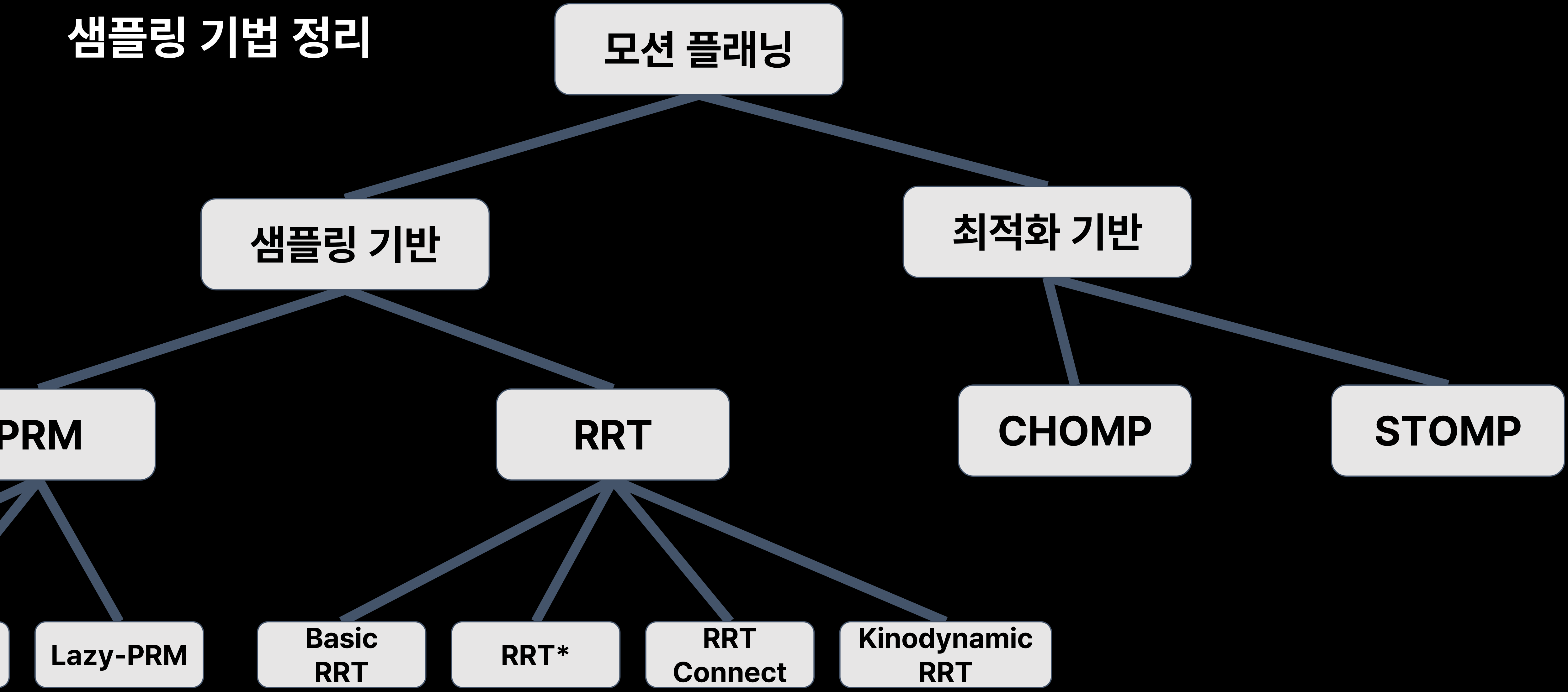

모션 플래닝
샘플링 기반
최적화 기반
PRM
RRT
Basic PRM
OB-PRM
Lazy-PRM
Basic RRT
RRT*
RRT Connect
Kinodynamic RRT

샘플링 기법 정리



모션 플래닝
샘플링 기반
최적화 기반
PRM
RRT
Lazy-PRM
Basic RRT
RRT*
RRT Connect
Kinodynamic RRT

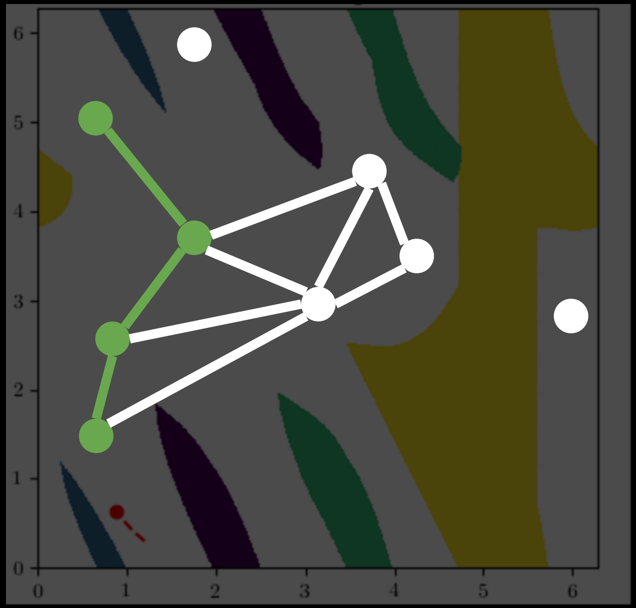
샘플링 기법 정리


모션 플래닝
샘플링 기반
최적화 기반
PRM
RRT
CHOMP
STOMP
Lazy-PRM
Basic RRT
RRT*
RRT Connect
Kinodynamic RRT

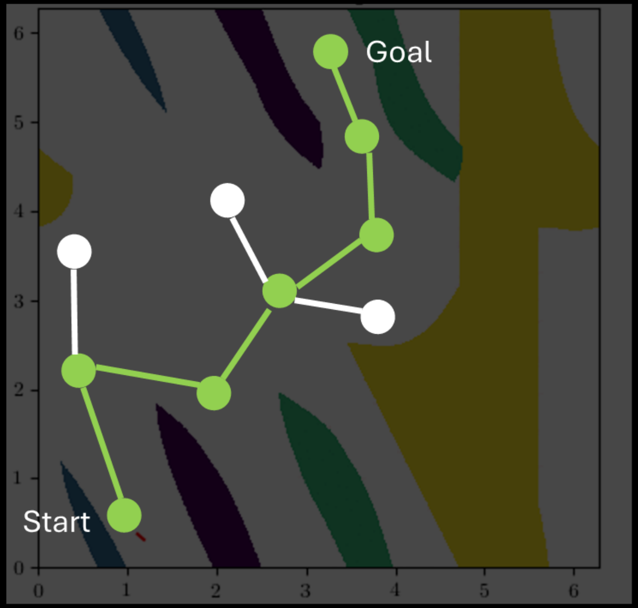
모션 퀄리티 향상 기법

- 솔루션 퀄리티가 떨어지는 이유

PRM



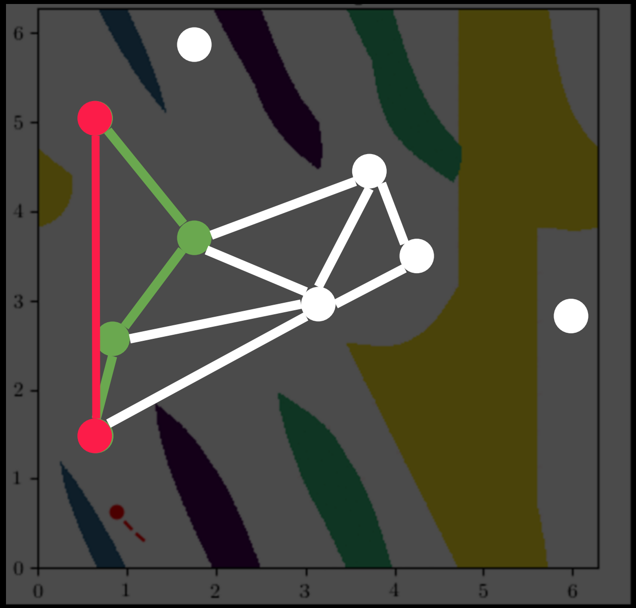
RRT



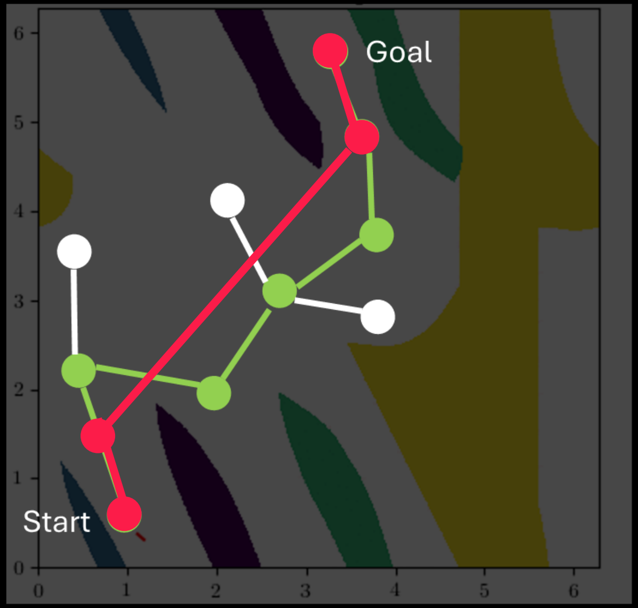
모션 퀄리티 향상 기법

- Path Shortcutting 기법
- 좀 더 부드러운 경로를 만들어낼 수는 없을까?

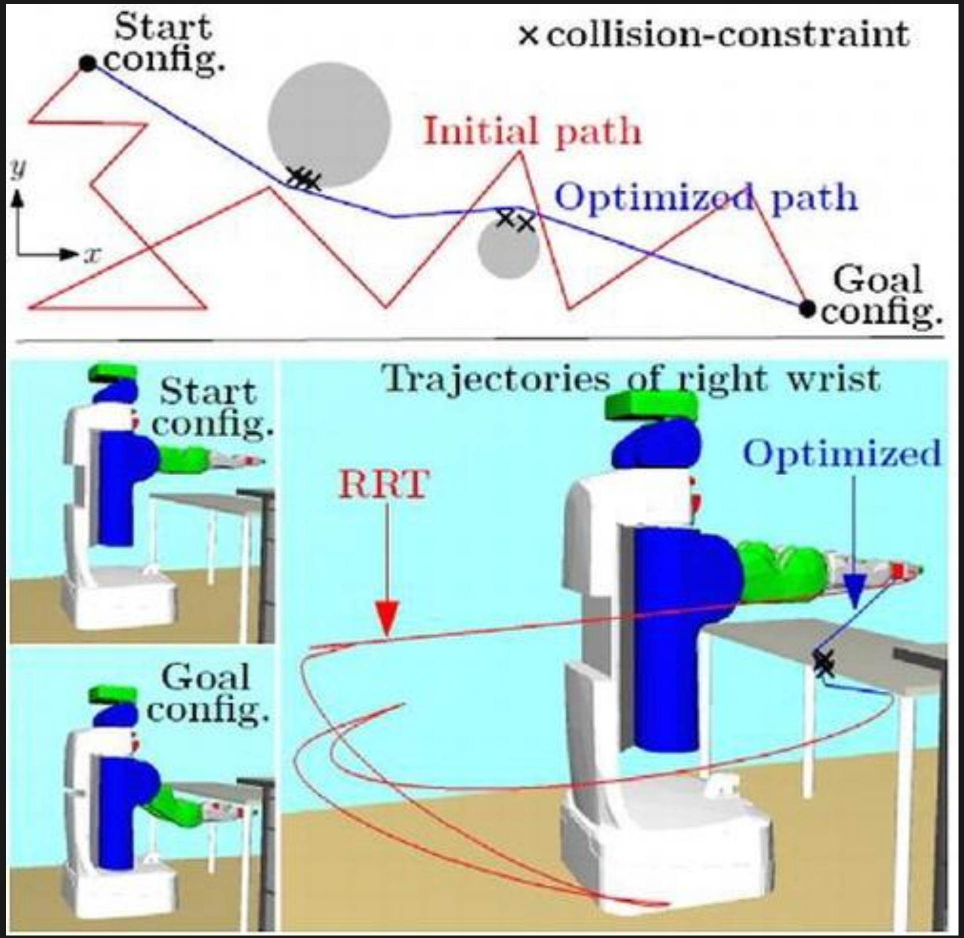
PRM



RRT



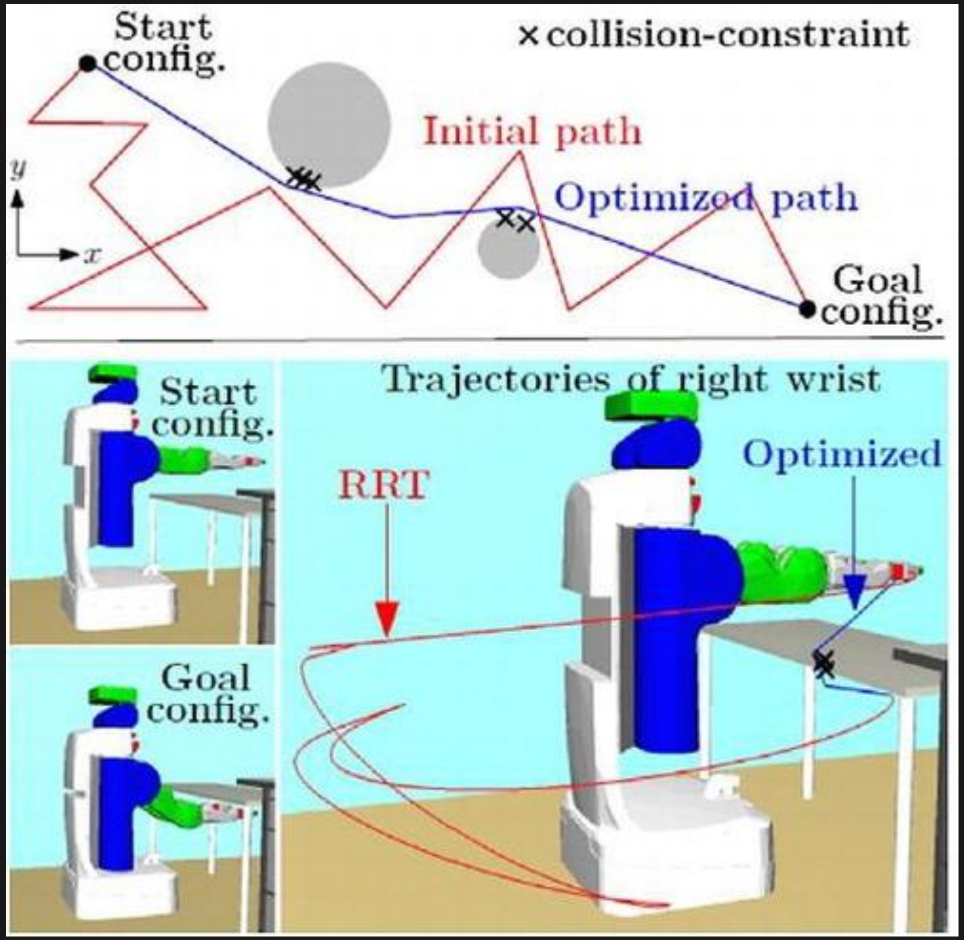
최적화 기법 vs. 샘플링 기법



Campna et al. A gradient-based path optimization method for motion planning

최적화 기법 vs. 샘플링 기법

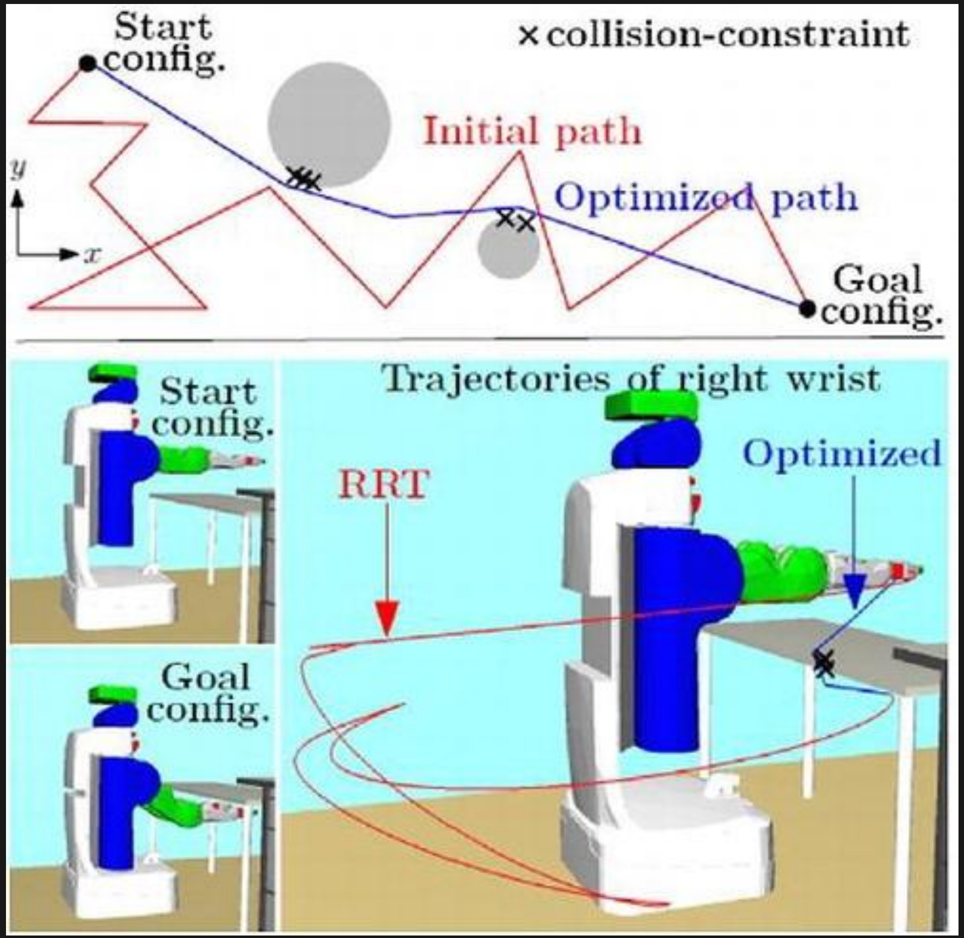
- 최적화란,
  - 예산 안에서 최대 효율로 물건 구매
  - 여행 경로에서 이동 거리 최소화
  - ...



Campna et al. A gradient-based path optimization method for motion planning

최적화 기법 vs. 샘플링 기법

- 그림 최적화 기법이 샘플링 기법보다 더 좋은가?



Campna et al. A gradient-based path optimization method for motion planning

최적화 기법 vs. 샘플링 기법

샘플링 기법

- 고차원의 C-Space 를 효과적으로 근사
- Probabilistic Completeness

- 최적의 경로를 보장하지 않음
- 솔루션 퀄리티가 낮음

최단 경로

알고리즘

최적화 기법

최적화 기법 vs. 샘플링 기법

샘플링 기법

- 고차원의 C-Space 를 효과적으로 근사
- Probabilistic Completeness

- 최적의 경로를 보장하지 않음
- 솔루션 퀄리티가 낮음

최단 경로 알고리즘

최적화 기법

- 최적의 경로를 보장
- 솔루션 퀄리티가 높음

최적화 기법 vs. 샘플링 기법

샘플링 기법

- 고차원의 C-Space 를 효과적으로 근사
- Probabilistic Completeness

- 최적의 경로를 보장하지 않음
- 솔루션 퀄리티가 낮음

최단 경로 알고리즘

최적화 기법

- 최적의 경로를 보장
- 솔루션 퀄리티가 높음

- 솔루션을 찾을 수 없는 경우도 있음
- 초기 예측값에 따라 성능이 달라짐

최적화 (optimization)

$$\min$$

최적화 (optimization)

$$\min$$

$$\max$$

최적화 (optimization)

$$\min$$

최적화 (optimization)

- 목적 함수 (objective function)

$$\min f(x)$$

최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)

$$\min_{x \in \mathcal{X}} f(x)$$

최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

최적화 (optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \ h_j(x) = 0$$

20 m 길이의 울타리를 이용해서, 가능한 넓이가 최대가 되는 직사각형 농장을 만드는 상황

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

20 m 길이의 울타리를 이용해서, 가능한 넓이가 최대가 되는 직사각형 농장을 만드는 상황

- 목적 함수 (objective function) $A(w, h) = w\,h$
- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

20 m 길이의 울타리를 이용해서, 가능한 넓이가 최대가 되는 직사각형 농장을 만드는 상황

- 목적 함수 (objective function) $A(w,h) = w\,h$
- 설계 변수 (design variables) w, h
- 제약 조건 (constraints)

최적화 (optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

20 m 길이의 울타리를 이용해서, 가능한 넓이가 최대가 되는 직사각형 농장을 만드는 상황

- 목적 함수 (objective function) $A(w, h) = w\,h$
- 설계 변수 (design variables) w, h
- 제약 조건 (constraints) $2(w + h) = 20$, $w \geq 0, \quad h \geq 0$

최적화 (optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

20 m 길이의 울타리를 이용해서, 가능한 넓이가 최대가 되는 직사각형 농장을 만드는 상황

- 목적 함수 (objective function) $A(w, h) = w\,h$
- 설계 변수 (design variables) w, h
- 제약 조건 (constraints) $2(w+h) = 20$, $w \geq 0, \quad h \geq 0$

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

  $$\int_0^1 \|\xi'(t)\| \, dt$$

- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

  $$\int_0^1 \|\xi'(t)\|\, dt$$

- 설계 변수 (design variables)

  $$\xi : [0,1] \rightarrow \mathcal{C}_{\text{free}}$$

- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

$$\int_0^1 \|\xi'(t)\| \, dt$$

- 설계 변수 (design variables)

$$\xi : [0,1] \to \mathcal{C}_{\text{free}}$$

- 제약 조건 (constraints)

$$\xi(0) = q_{\text{start}}, \quad \xi(1) = q_{\text{goal}},$$
$$\xi(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0,1].$$

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

$$\int_0^T \|u(t)\|^2 \, dt$$

- 설계 변수 (design variables)
- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

$$\int_0^T \|u(t)\|^2 \, dt$$

- 설계 변수 (design variables)

궤적 위치 $q(t)$

궤적 속도 $v(t)$

제어 입력(가속도) $u(t)$

총 궤적 실행 시간 T

- 제약 조건 (constraints)

최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

$$\int_0^T \|u(t)\|^2\, dt$$

- 설계 변수 (design variables)

궤적 위치 $q(t)$

궤적 속도 $v(t)$

제어 입력(가속도) $u(t)$

총 궤적 실행 시간 T

- 제약 조건 (constraints)

$$\begin{aligned}
&\dot{q}(t) = v(t),\\
&\dot{v}(t) = u(t),\\
&q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}},\\
&q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}},\\
&q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T],\\
&u_{\min} \leq u(t) \leq u_{\max} \quad \forall t \in [0, T]
\end{aligned}$$

최적화 (optimization) 예시 - 모션 플래닝

최적화 기법의 특징

- 문제 정의의 자유도가 높음

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \ h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 목적 함수 (objective function)

$$\int_0^T \|u(t)\|^2 \, dt$$

- 설계 변수 (design variables)

궤적 위치 $q(t)$

궤적 속도 $v(t)$

제어 입력(가속도) $u(t)$

총 궤적 실행 시간 T

- 제약 조건 (constraints)

$$\begin{aligned}
&\dot{q}(t) = v(t), \\
&\dot{v}(t) = u(t), \\
&q(0) = q_{\text{start}}, \quad v(0) = v_{\text{start}}, \\
&q(T) = q_{\text{goal}}, \quad v(T) = v_{\text{goal}}, \\
&q(t) \in \mathcal{C}_{\text{free}} \quad \forall t \in [0, T], \\
&u_{\min} \leq u(t) \leq u_{\max} \quad \forall t \in [0, T]
\end{aligned}$$

최적화 (optimization)

최적화 기법의 특징

- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

최적화 (optimization)

최적화 기법의 특징
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

- 어떻게 솔루션을 찾을까?

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

최적화 (optimization)

최적화 기법의 특징

- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

최적화 (optimization)

최적화 기법의 특징

- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신

$$x_{k+1} = x_k + \alpha_k\, d_k$$

d_k: 하강 방향
α_k: 스텝 크기

최적화 (optimization)

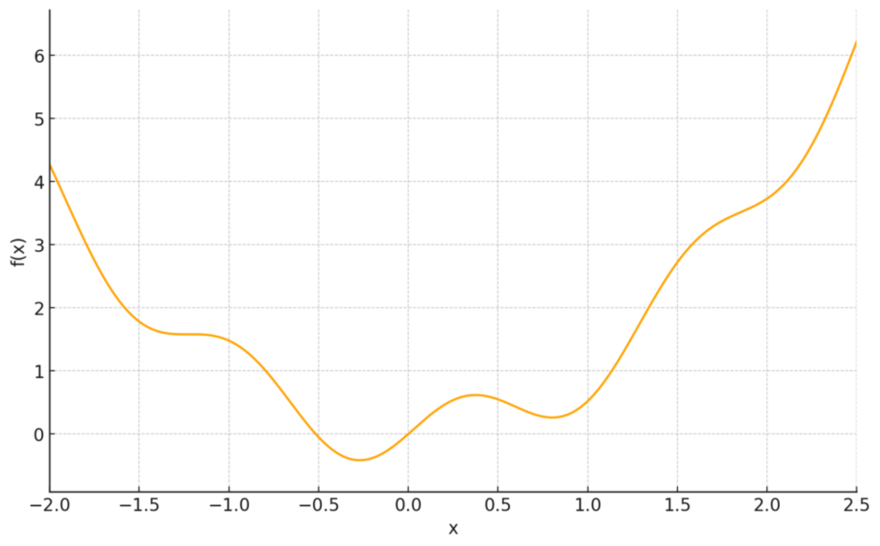
최적화 기법의 특징
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 (optimization)

최적화 기법의 특징

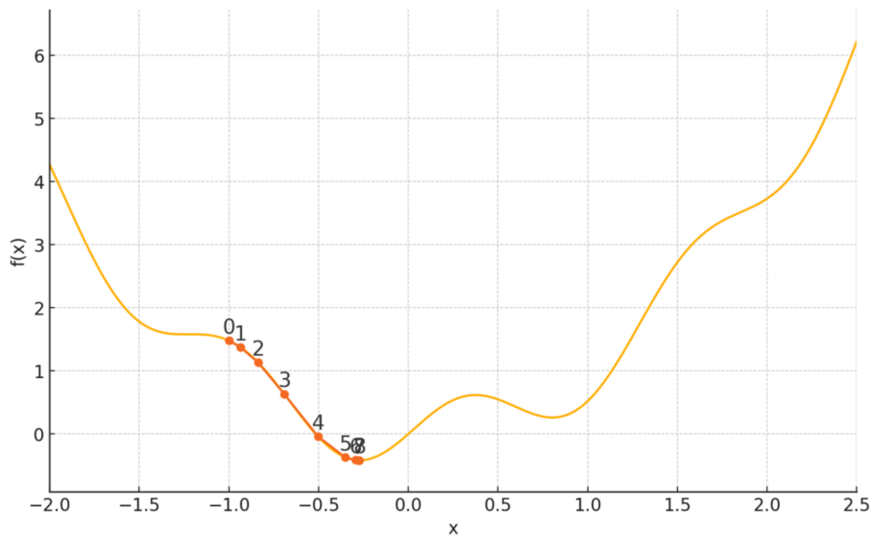
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k\, d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 (optimization)

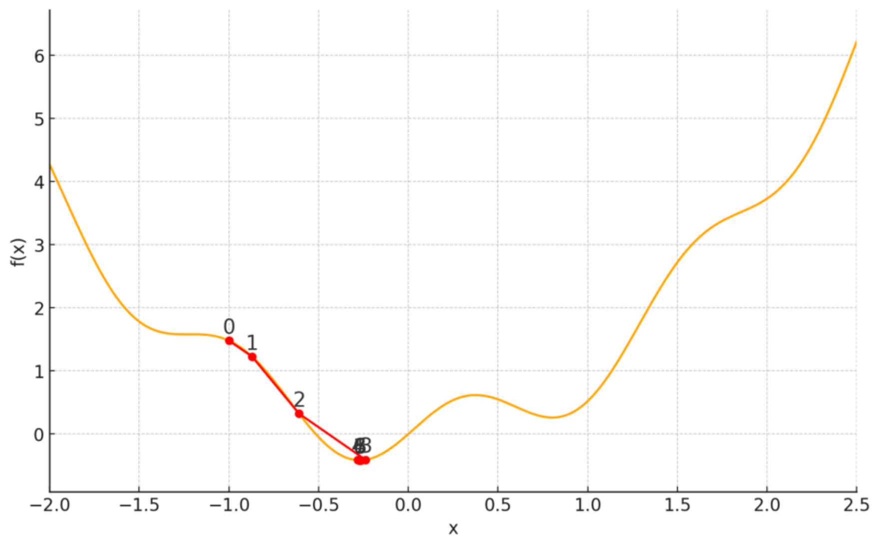
최적화 기법의 특징
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 (optimization)

최적화 기법의 특징

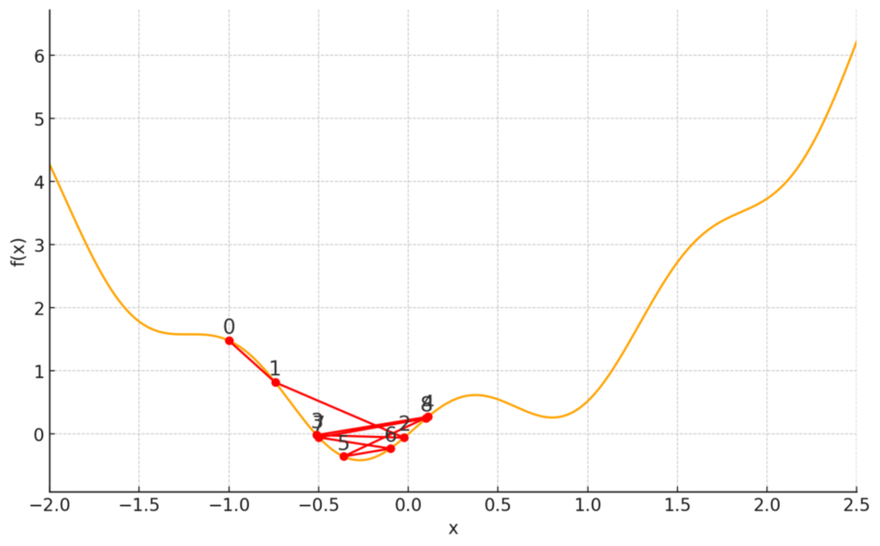
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k \, d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 (optimization)

최적화 기법의 특징

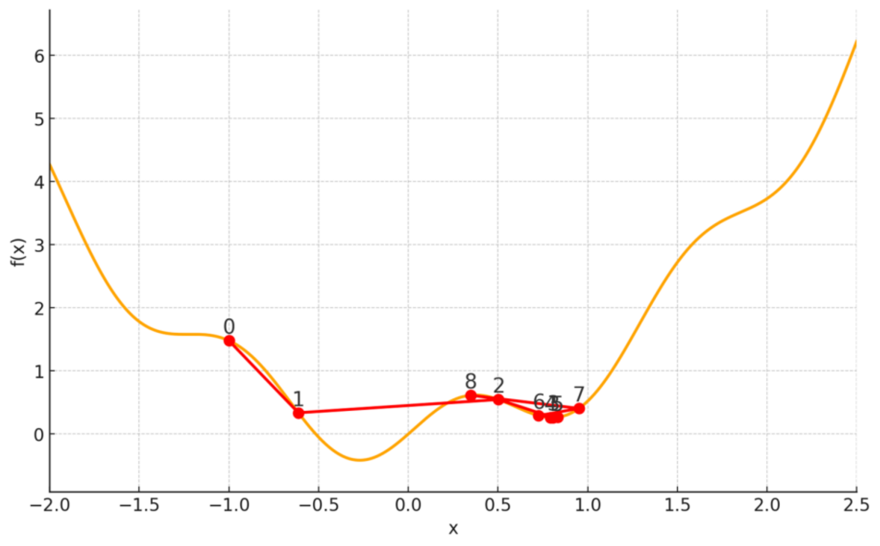
- 문제 정의의 자유도가 높음

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



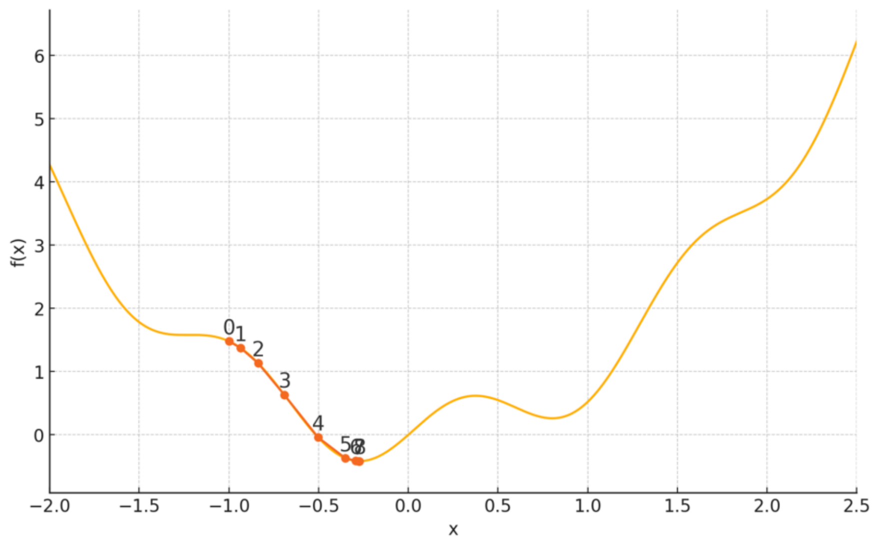
최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$



최적화 기법의 특징
- 문제 정의의 자유도가 높음
- Local minimum problem

최적화 (optimization)

최적화 기법의 특징

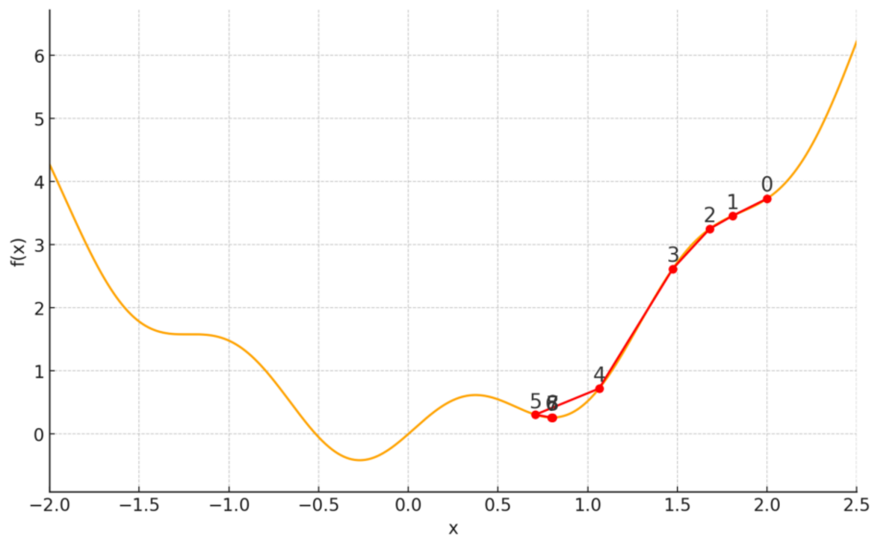
- 문제 정의의 자유도가 높음
- Local minimum problem

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



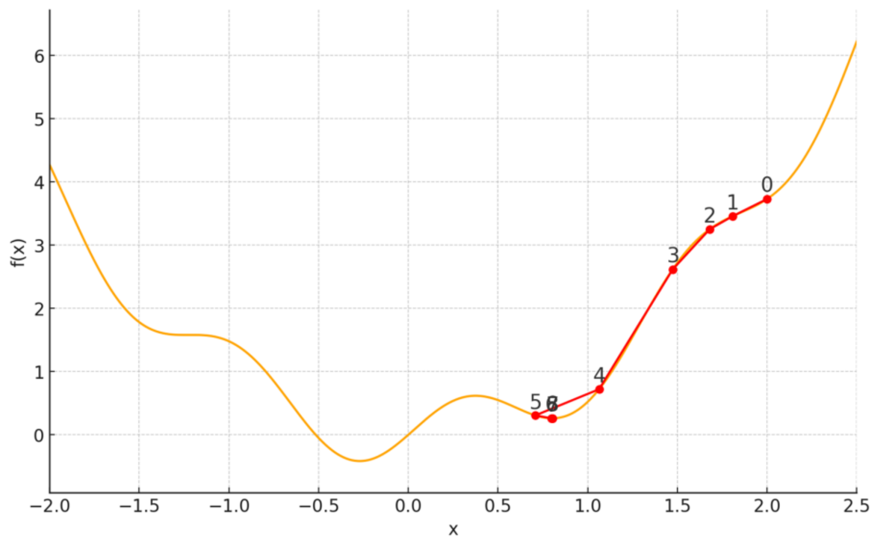
최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 기법의 특징

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향

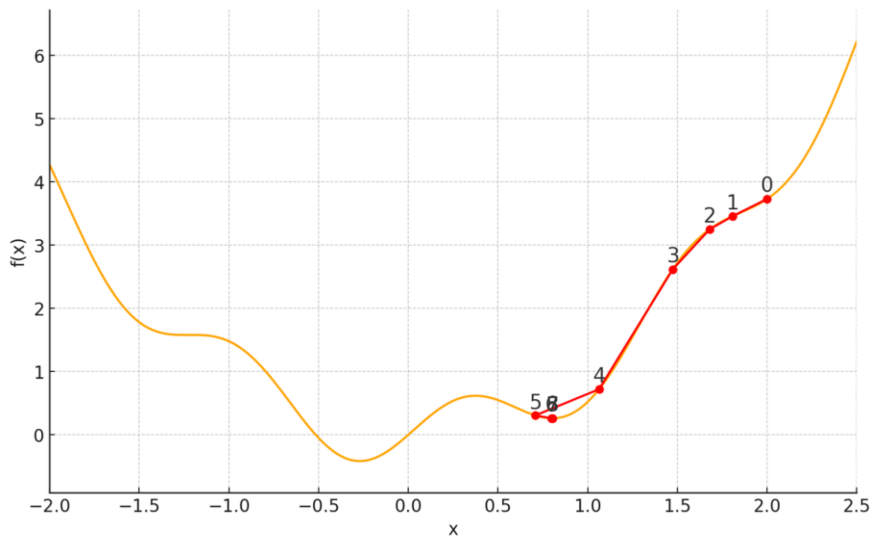
최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복



최적화 기법의 특징

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향
- 경우에 따라서는 솔루션을 찾을 수 없기도 함

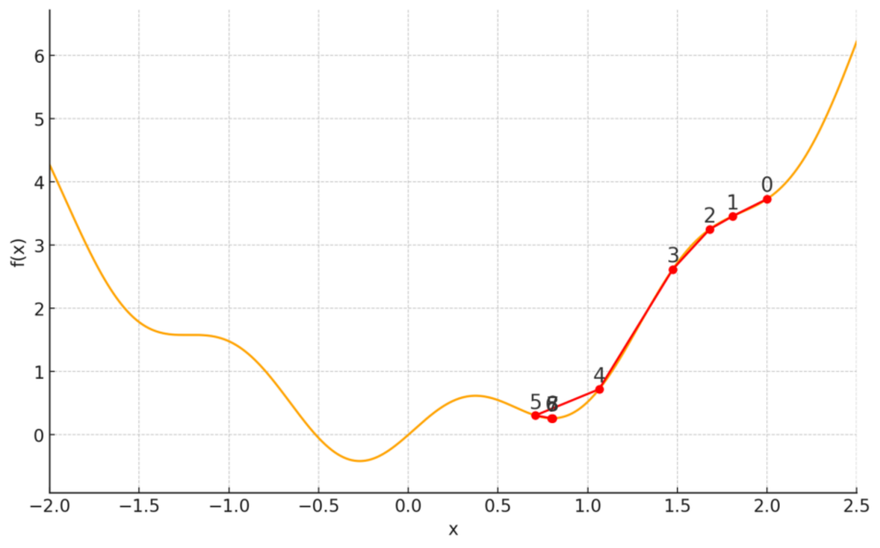
최적화 (optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

- 어떻게 솔루션을 찾을까?
  → 경사 하강법 (gradient descent)
  → 시작점 설정 x_0
  → 변수 갱신
  $x_{k+1} = x_k + \alpha_k \, d_k$
  d_k: 하강 방향
  α_k: 스텝 크기
  → 반복
- Deterministic vs. Stochastic

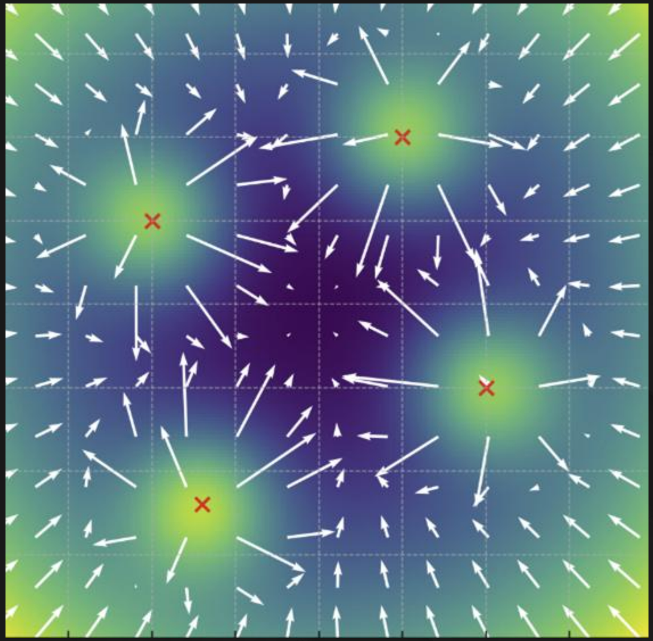
$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$



최적화 기법의 특징

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향
- 경우에 따라서는 솔루션을 찾을 수 없기도 함



최적화 (optimization) 예시 - 모션 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

충돌이 일어나지 않는 최단 경로를 찾기

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향
- 경우에 따라서는 솔루션을 찾을 수 없기도 함

최적화 (optimization)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

변수가 실수가 아니라 정수라면?

- 예시: 마트에서 물품을 구매

최적화 (optimization)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

변수가 실수가 아니라 정수라면?

- 예시: 마트에서 물품을 구매
- Gradient Descent 기법을 바로 적용하기 어려움
- 이산 최적화

이산 최적화 (discrete optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables) → {0,1,2}
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

이산 최적화 (discrete optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

각각 무게와 가치가 정해진 여러 개의 물건이 있을 때, 가치의 총 합이 가장 높게 가방 안에 물건을 담는 방법은?

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

각각 무게와 가치가 정해진 여러 개의 물건이 있을 때, 가치의 총 합이 가장 높게 가방 안에 물건을 담는 방법은?

- 목적 함수 (objective function)
- 설계 변수 (design variables)

$$x_i = \begin{cases} 1, & \text{물건 } i\text{를 배낭에 넣음,} \\ 0, & \text{물건 } i\text{를 배낭에 넣지 않음.} \end{cases} \qquad i = 1, 2, \ldots, n.$$

- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

각각 무게와 가치가 정해진 여러 개의 물건이 있을 때, 가치의 총 합이 가장 높게 가방 안에 물건을 담는 방법은?

- 목적 함수 (objective function) $\max_{x_1,\ldots,x_n} \quad \sum_{i=1}^{n} v_i\, x_i$
- 설계 변수 (design variables) $x_i \;=\; \begin{cases} 1, & \text{물건 } i\text{를 배낭에 넣음,} \\ 0, & \text{물건 } i\text{를 배낭에 넣지 않음.} \end{cases} \qquad i = 1, 2, \ldots, n.$
- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

각각 무게와 가치가 정해진 여러 개의 물건이 있을 때, 가치의 총 합이 가장 높게 가방 안에 물건을 담는 방법은?

- 목적 함수 (objective function) $\max_{x_1,\ldots,x_n} \quad \sum_{i=1}^{n} v_i\, x_i$
- 설계 변수 (design variables) $x_i \;=\; \begin{cases} 1, & \text{물건 } i\text{를 배낭에 넣음}, \\ 0, & \text{물건 } i\text{를 배낭에 넣지 않음}. \end{cases} \qquad i = 1, 2, \ldots, n.$
- 제약 조건 (constraints) $\sum_{i=1}^{n} w_i\, x_i \;\leq\; W$

$x_i \in \{0,\, 1\}, \quad \forall\, i = 1, \ldots, n.$

이산 최적화 (discrete optimization) 예시 - 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

4대의 모바일 로봇 (A, B, C, D) 이 4개의 픽업 (Pick-up) 작업(Task 1, 2, 3, 4)을 수행하는데 가장 짧은 방법은?

- 목적 함수 (objective function)
- 설계 변수 (design variables)
- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시 - 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

4대의 모바일 로봇 (A, B, C, D) 이 4개의 픽업 (Pick-up) 작업(Task 1, 2, 3, 4)을 수행하는데 가장 짧은 방법은?

- 목적 함수 (objective function)
- 설계 변수 (design variables)

$$x_{i,j} = \begin{cases} 1, & \text{로봇 } i\text{가 Task } j\text{를 수행한다,} \\ 0, & \text{그렇지 않다,} \end{cases} \quad i \in \{A, B, C, D\},\ j \in \{1, 2, 3, 4\}$$

- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시 - 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

4대의 모바일 로봇 (A, B, C, D) 이 4개의 픽업 (Pick-up) 작업(Task 1, 2, 3, 4)을 수행하는데 가장 짧은 방법은?

- 목적 함수 (objective function) $\min \sum_{i \in \{A,B,C,D\}} \sum_{j=1}^{4} d_{i,j}\, x_{i,j}$
- 설계 변수 (design variables) $x_{i,j} = \begin{cases} 1, & \text{로봇 } i\text{가 Task } j\text{를 수행한다,} \\ 0, & \text{그렇지 않다,} \end{cases} \quad i \in \{A, B, C, D\},\ j \in \{1, 2, 3, 4\}$
- 제약 조건 (constraints)

이산 최적화 (discrete optimization) 예시 - 플래닝

$$\min_{x \in \mathcal{X}} f(x)$$
$$g_i(x) \leq 0,\ h_j(x) = 0$$

4대의 모바일 로봇 (A, B, C, D) 이 4개의 픽업 (Pick-up) 작업(Task 1, 2, 3, 4)을 수행하는데 가장 짧은 방법은?

- 목적 함수 (objective function) $\min \sum_{i \in \{A,B,C,D\}} \sum_{j=1}^{4} d_{i,j}\, x_{i,j}$
- 설계 변수 (design variables) $x_{i,j} = \begin{cases} 1, & \text{로봇 } i\text{가 Task } j\text{를 수행한다,} \\ 0, & \text{그렇지 않다,} \end{cases} \quad i \in \{A,B,C,D\},\ j \in \{1,2,3,4\}$
- 제약 조건 (constraints) $x_{i,j} \in \{0,1\}, \quad \forall i \in \{A,B,C,D\},\ j \in \{1,2,3,4\}$

이산 최적화 (discrete optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables) → {0,1,2}
- 제약 조건 (constraints)

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

이산 최적화 (discrete optimization)

- 목적 함수 (objective function)
- 설계 변수 (design variables) → {0,1,2}
- 제약 조건 (constraints)

- 어떻게 솔루션을 찾을까?
  → Linear Programming (LP)
    → Simplex Method
  → Branch and Bound

- 추가 예시: Job Scheduling, Resource Allocation, etc.

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

최적화 (optimization) 라이브러리

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0, \; h_j(x) = 0$$

실제 코드를 구현할 때는 라이브러리를 활용

- High-level Library:
- Low-level Library:

최적화 (optimization) 라이브러리

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

실제 코드를 구현할 때는 라이브러리를 활용

- High-level Library: MoveIt (OMPL), MPLib 등
  → 최적화 문제를 정의해주지 않아도 (로봇, 환경, 시작점, 도착점) 만 알려주면 모션플래닝 문제를 알아서 풀어줌.
- Low-level Library:

최적화 (optimization) 라이브러리

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ h_j(x) = 0$$

실제 코드를 구현할 때는 라이브러리를 활용

- High-level Library: MoveIt (OMPL), MPLib 등
  → 최적화 문제를 정의해주지 않아도 (로봇, 환경, 시작점, 도착점) 만 알려주면 모션플래닝 문제를 알아서 풀어줌.
- Low-level Library: Scipy, Gurobi 등
  → 최적화 문제를 직접 정의해 주어야 함. 연구 목적에 가까움

최적화 (optimization) 라이브러리

$$\min_{x \in \mathcal{X}} f(x)$$

$$g_i(x) \leq 0,\ \ h_j(x) = 0$$

실제 코드를 구현할 때는 라이브러리를 활용

- High-level Library: MoveIt (OMPL), MPLib 등
  → 최적화 문제를 정의해주지 않아도 (로봇, 환경, 시작점, 도착점) 만 알려주면 모션플래닝 문제를 알아서 풀어줌.
- Low-level Library: Scipy, Gurobi 등
  → 최적화 문제를 직접 정의해 주어야 함. 연구 목적에 가까움
- 실제 산업 현장에서는 두 가지를 함께 활용

최적화 기법 vs. 샘플링 기법 - 모션 플래닝 관점

샘플링 기법

- 고차원의 C-Space 를 효과적으로 근사
- Probabilistic Completeness

- 최적의 경로를 보장하지 않음
- 솔루션 퀄리티가 낮음
- Narrow Passage Problem

최단 경로 알고리즘

최적화 기법

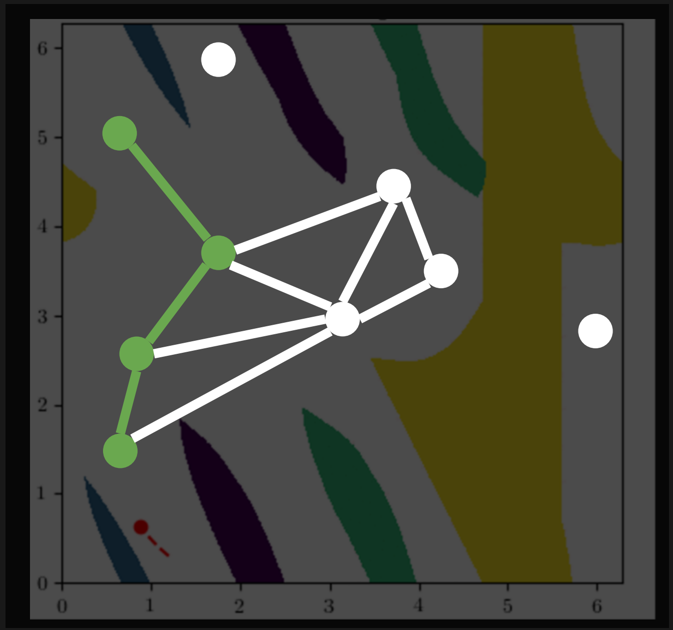
- 최적의 경로를 보장
- 솔루션 퀄리티가 높음
- 문제 정의의 자유도가 높음

- 솔루션을 찾을 수 없는 경우도 있음
- 초기 예측값에 따라 성능이 달라짐
- Local minimum problem

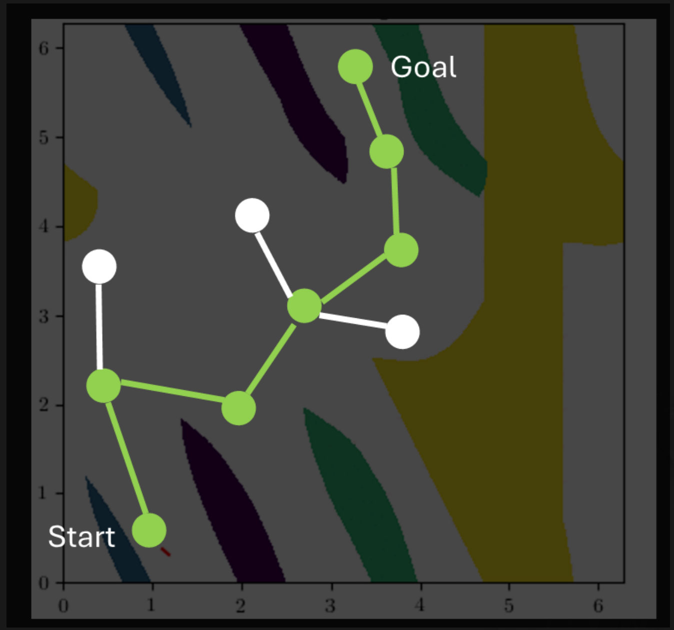
샘플링 기법과의 비교

- 솔루션 퀄리티가 떨어지는 이유

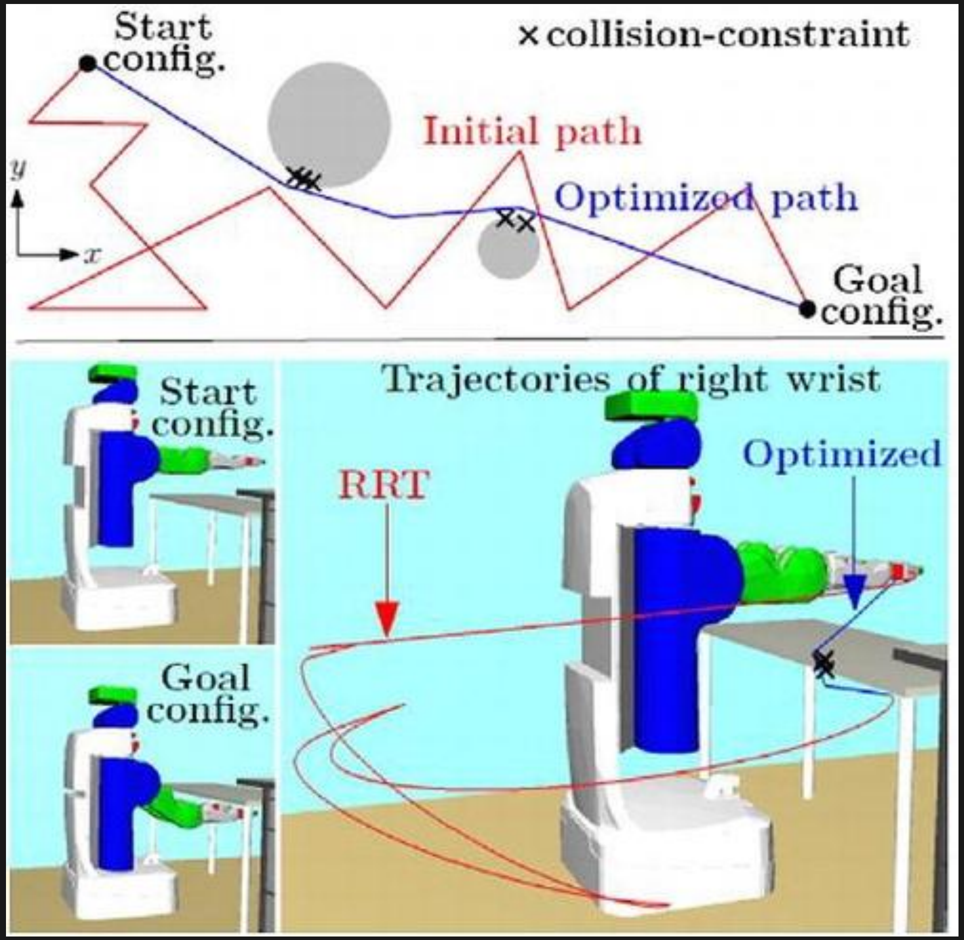
PRM



RRT

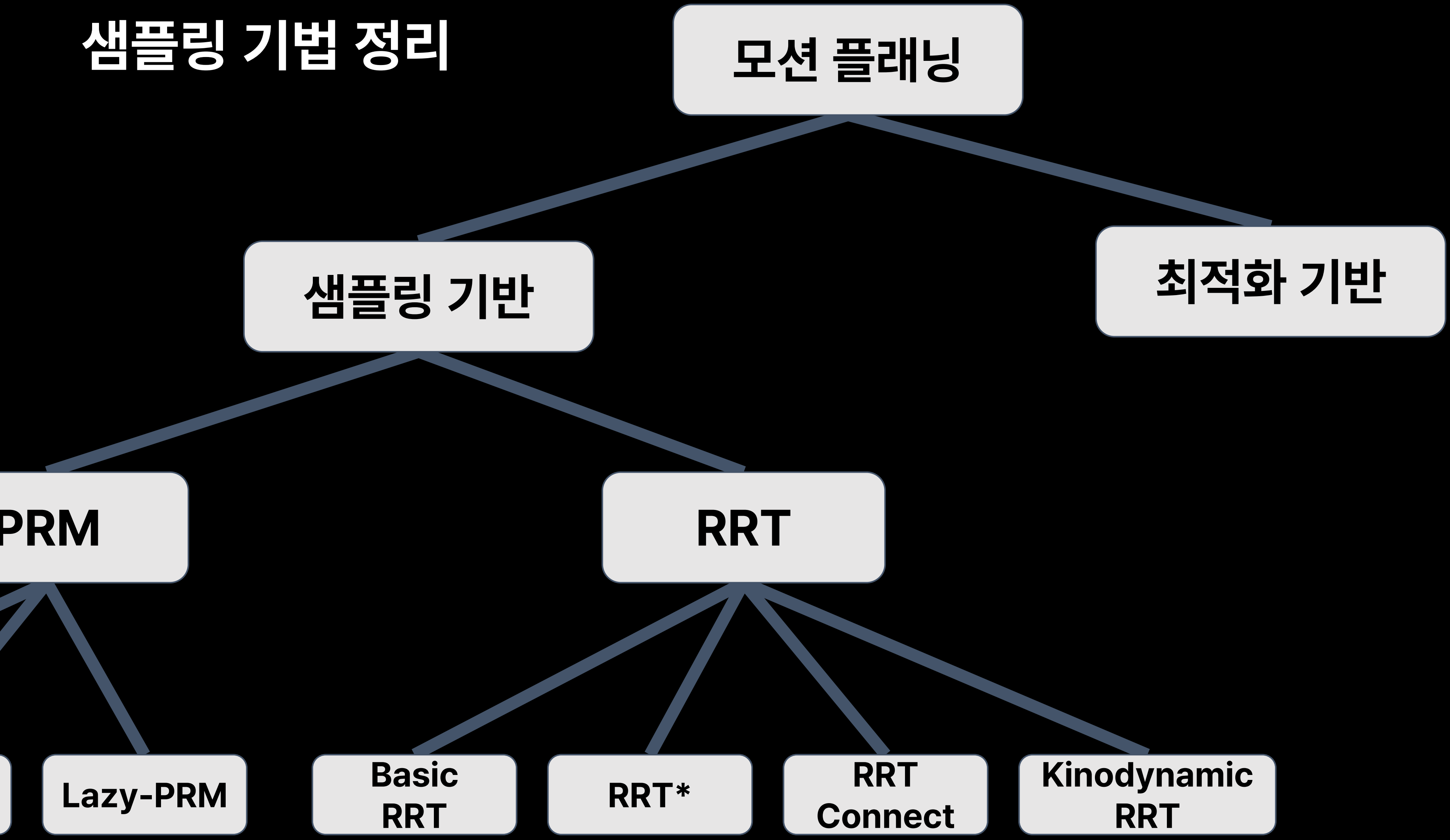


샘플링 기법과의 비교

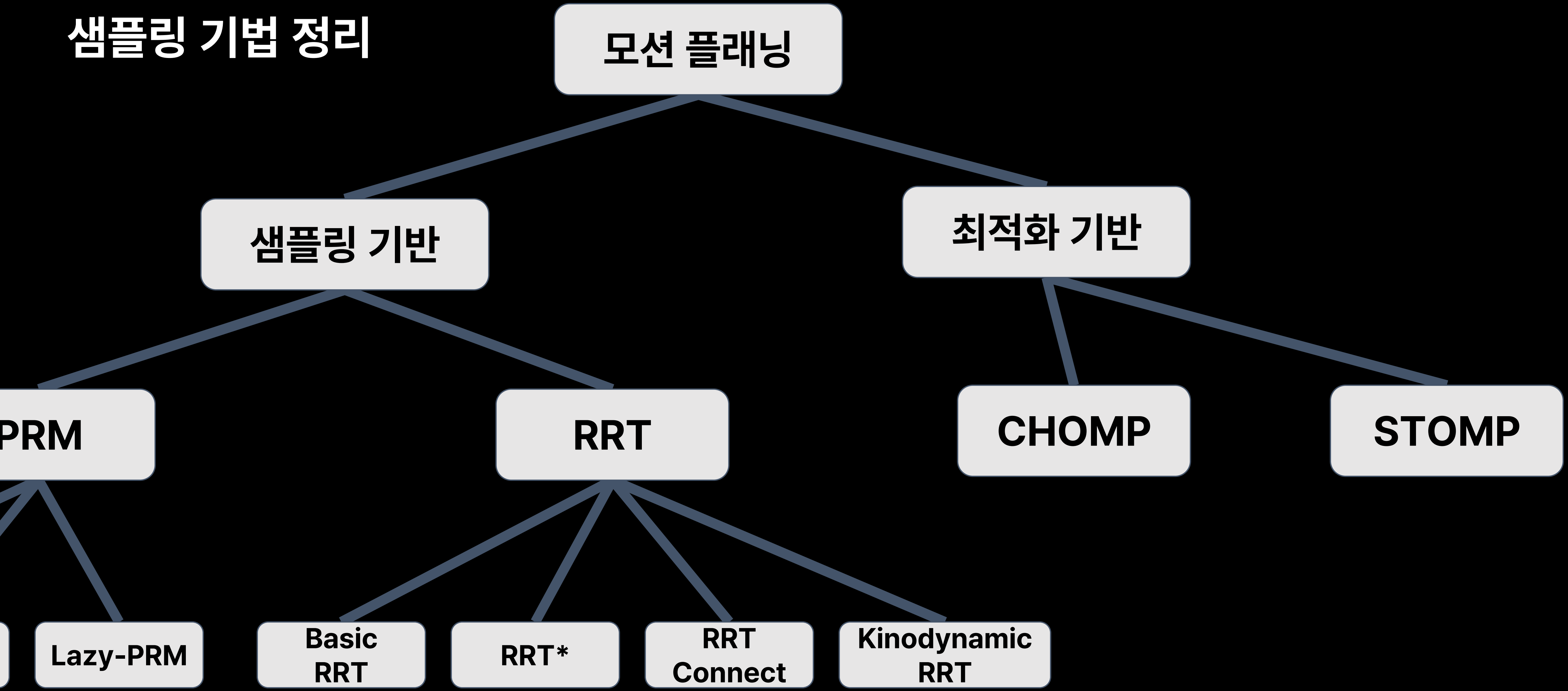


Campna et al. A gradient-based path optimization method for motion planning

샘플링 기법 정리


모션 플래닝
샘플링 기반
최적화 기반
PRM
RRT
Lazy-PRM
Basic RRT
RRT*
RRT Connect
Kinodynamic RRT

샘플링 기법 정리


모션 플래닝
샘플링 기반
최적화 기반
PRM
RRT
CHOMP
STOMP
Lazy-PRM
Basic RRT
RRT*
RRT Connect
Kinodynamic RRT

강의 요약

01

최적화 기법 vs. 샘플링 기법

02

최적화 문제 정의

- 목적 함수
- 설계 변수
- 제약 조건

03

최적화 문제 특징

- 문제 정의의 자유도가 높음
- Local minimum problem
- 시작점 위치의 영향
- 경우에 따라서는 솔루션을 찾을 수 없기도 함
- Deterministic & Stochastic Gradient Descent

04

최적화 라이브러리

- High-level: 실제 적용 목적
- Low-level: 연구 목적

$$a \cdot x \le b + M \cdot (1 - y)$$

y	의미	결과
1	조건이 **참** → 제약 적용	$a \cdot x \le b$
0	조건이 **거짓** → 제약 제거	$a \cdot x \le b + M$ (사실상 항상 참)

- $y = 1$: A → B
- $y = 0$: B → A

$$\text{A 먼저} \quad \Rightarrow \quad \texttt{start[A] + dur[A]} \le \texttt{start[B] + M(1 - y)}$$