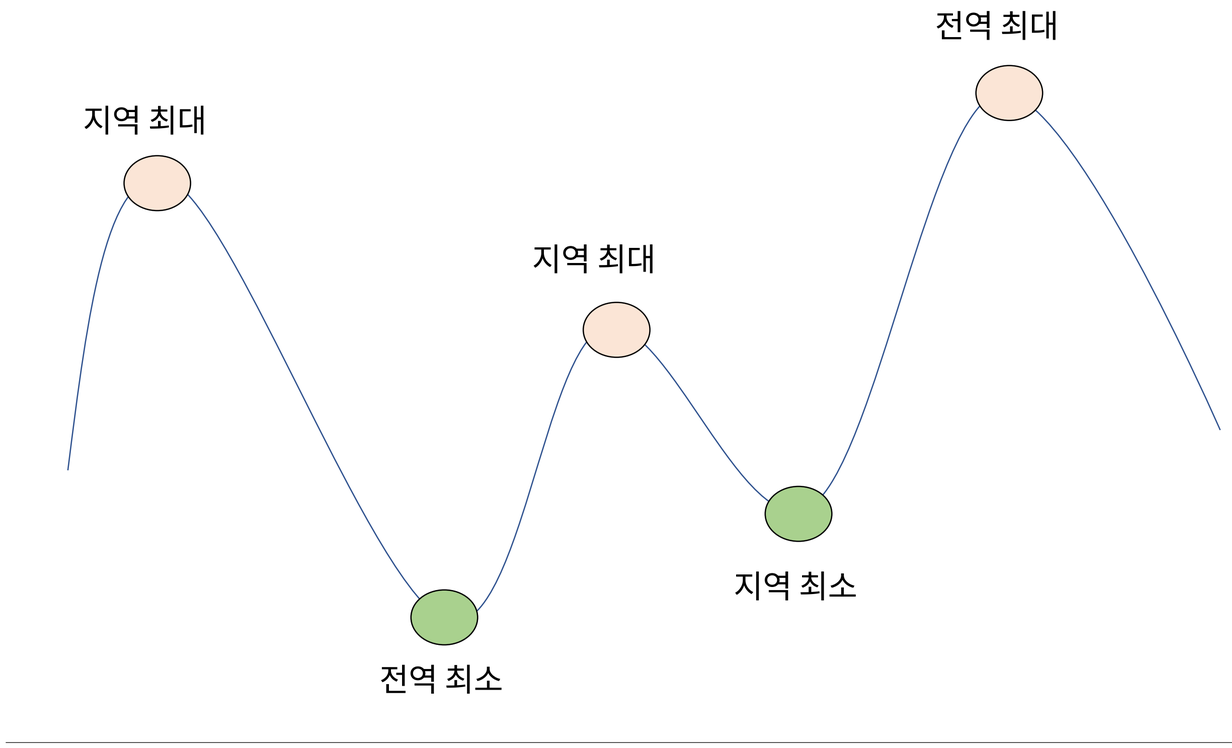
2. 다양한 해법

1 최적해 및 해법의 종류

최적해의 종류

최적해는 지역 최적(local optimum)과 전역 최적(global optimum)으로 구분할 수 있습니다.



- 지역 최적: 일부 영역(지역) 내에서 가장 좋은 해
- 전역 최적: 전체 탐색 공간에서 가장 좋은 해
- 전역 최적이 지역 최적보다 좋지만 찾는 것이 쉽지 않음
- 심지어는 찾은 해가 전역 최적인지 알기도 어려움

해법의 구분

최적화 문제의 해법은 전역 최적해를 찾을 수 있는지 기준으로 구분할 수 있습니다.

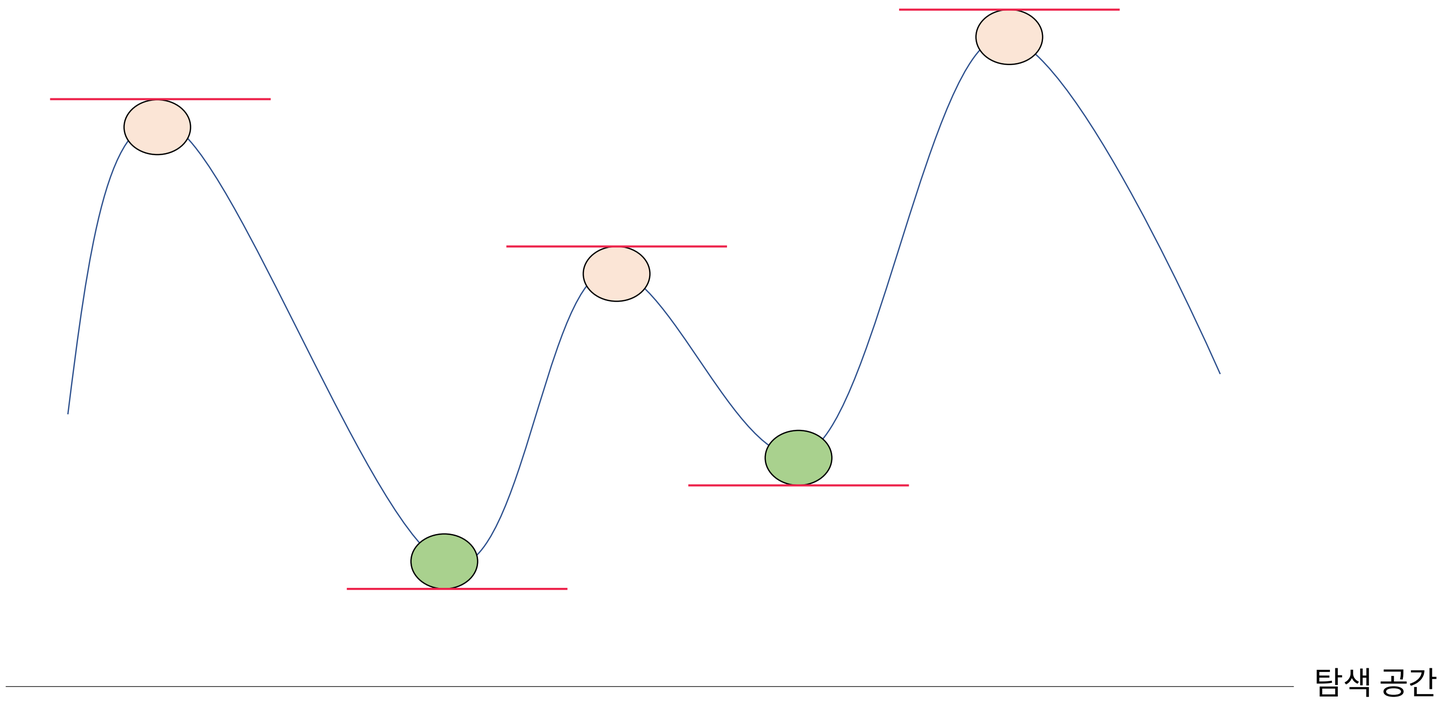
전역 최적해를 찾을 수 있는지 여부	방법	적용 가능한 문제의 특징
O	미분을 이용한 해법	• 제약이 없음 • 목적 함수가 미분 가능함 • 결정 변수로 미분한 값이 모두 0이 되는 연립 방정식을 풀 수 있음
O	라그랑주 승수법 및 KKT 조건	• 제약이 있음 • 목적 함수가 미분 가능함 • 결정 변수로 미분한 값이 모두 0이 되는 연립 방정식을 풀 수 있음
O	전역 탐색	• 제약이 있음 • 탐색 공간의 크기가 유한함
X	탐욕 알고리즘	• 순서를 결정하는 문제
X	내리막 경사법	• 목적 함수가 미분 가능함
X	담금질 기법	• 해 간 유사도를 측정할 수 있음
X	입자 군집 최적화	• 특별한 제약은 없으나, 모든 결정 변수가 연속형일 때 적절
X	유전 알고리즘	• 특별한 제약은 없으나, 모든 결정 변수가 범주형일 때 적절

2. 다양한 해법

2 미분을 이용한 해법

최적해의 특성

지역 최적과 전역 최적 모두 접선의 기울기가 0입니다.



제약이 없는 문제

목적 함수 $f(\boldsymbol{x})$를 결정 변수 $\boldsymbol{x}$로 미분한 값이 0이 되는 결정 변수만 비교하면 전역 최적을 알 수 있습니다

해법: 다음을 만족하는 해를 비교

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \cdots, \frac{\partial f(\boldsymbol{x})}{\partial x_m}\right) = (0, 0, \cdots, 0)$$

(예시) $f(x) = x^2 - 4x + 9$**의 최솟값 구하기**

$$\frac{\partial f(x)}{\partial x} = 2x - 4 = 0$$

→ $f(2) = 5$

- 미분해서 0이 되는 점이 하나라서 별다른 비교를 하지는 않음

등식 형태의 제약식이 있는 문제

등식 형태의 제약식이 있는 최적화 문제는 라그랑주 승수법(Lagrange multiplier)을 이용해 최적해를 구합니다.

문제 구조

- 목적 함수: $f(\boldsymbol{x})$
- i번째 제약식: $g_i(\boldsymbol{x}) = 0$

해법: 다음을 만족하는 해를 비교

$$\frac{\partial h(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \boldsymbol{x}} = \mathbf{0}$$

$$\frac{\partial h(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{0}$$

- $h(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{k} \lambda_i \, g_i(\boldsymbol{x})$
- 라그랑주 승수 λ_j는 0이 아니어야 함

부등식 형태의 제약식이 있는 문제

부등식 형태의 제약식이 있는 최적화 문제는 KKT(Karush-Kuhn-Tucker) 조건을 이용해 구합니다.

문제 구조

- 목적 함수: $f(\boldsymbol{x})$
- i번째 제약식: $g_i(\boldsymbol{x}) \leq 0$

해법: 다음을 만족하는 해를 비교

$$\frac{\partial h(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \boldsymbol{x}} = \mathbf{0}$$

$$\boldsymbol{\lambda} \cdot \frac{\partial h(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{0}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}$$

2. 다양한 해법

3 휴리스틱 방법

휴리스틱 해법이란?

휴리스틱 해법은 수학적으로 완벽한 방법은 아니지만, 경험과 직관 등을 활용하여 그럴싸하게 답을 찾는 방법이라고 할 수 있습니다

분석적 해법

- 함수 형태의 해로 문제를 푸는 접근
- 전역 최적을 구할 수 있음

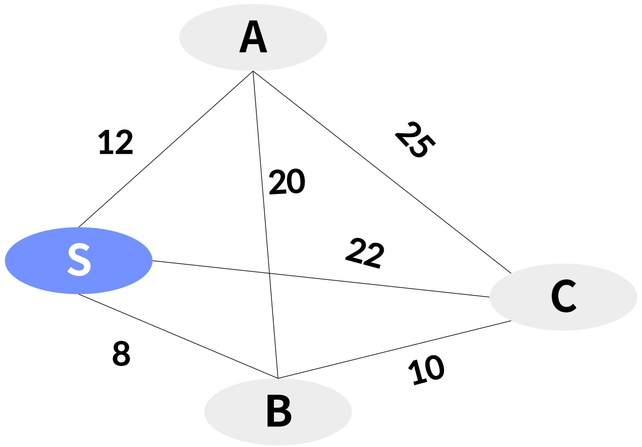
휴리스틱 해법

- 경험과 직관을 활용함
- 전역 최적을 보장할 수 없음
- 문제마다 방법을 설계해야 함

탐욕 알고리즘

탐욕 알고리즘(greedy algorithm)은 미래를 고려하지 않고 각 단계에서 가장 좋은 선택을 하는 방법입니다.

(예시) 외판원 순회 문제: S에서 출발하여 A, B, C를 모두 방문하고 S로 되돌아오는 최소 거리 구하기



(1) 현 위치: S, 아직 방문하지 않은 노드: {A, B, C}, 가장 가까운 노드: B (거리 8)

(2) 현 위치: B, 아직 방문하지 않은 노드: {A, C}, 가장 가까운 노드: C (거리 10)

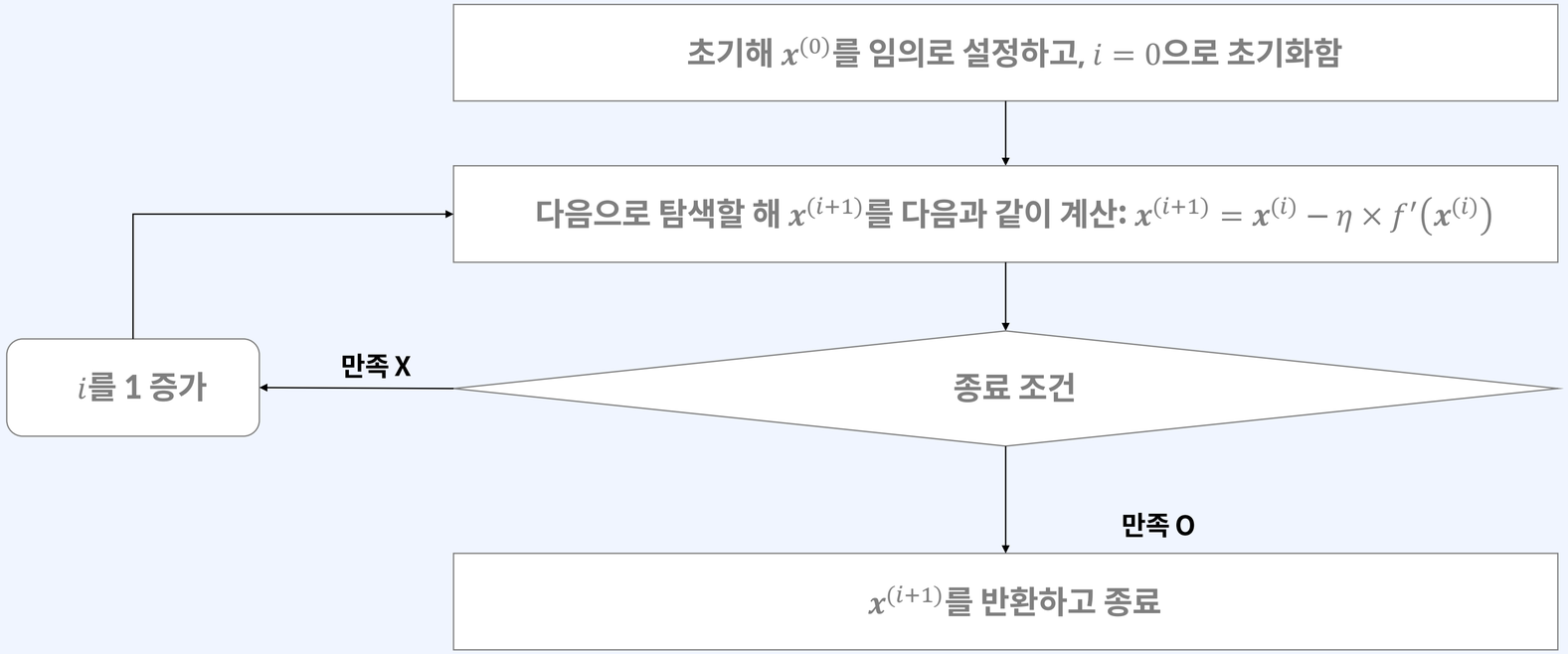
(3) 현 위치: C, 아직 방문하지 않는 노드: {A}, 가장 가까운 노드: A (거리 25)

(4) 현 위치: A, 아직 방문하지 않은 노드: 없음, 가장 가까운 노드: 없음 → S로 복귀

→ 해: S – B – C – A – S (목적 함수: 8 + 10 + 25 + 12 = 55)

내리막 경사법

내리막 경사법은 현재의 위치에서 그은 접선의 기울기 반대 방향으로 이동하는 것을 반복하여 지역 최적에 도달하는 휴리스틱 해법입니다.



- f': 목적 함수 f의 도함수
- η: 학습률

담금질 기법

담금질 기법(simulated annealing)은 지역 최적에 빠지는 것을 방지하기 위한 휴리스틱입니다.

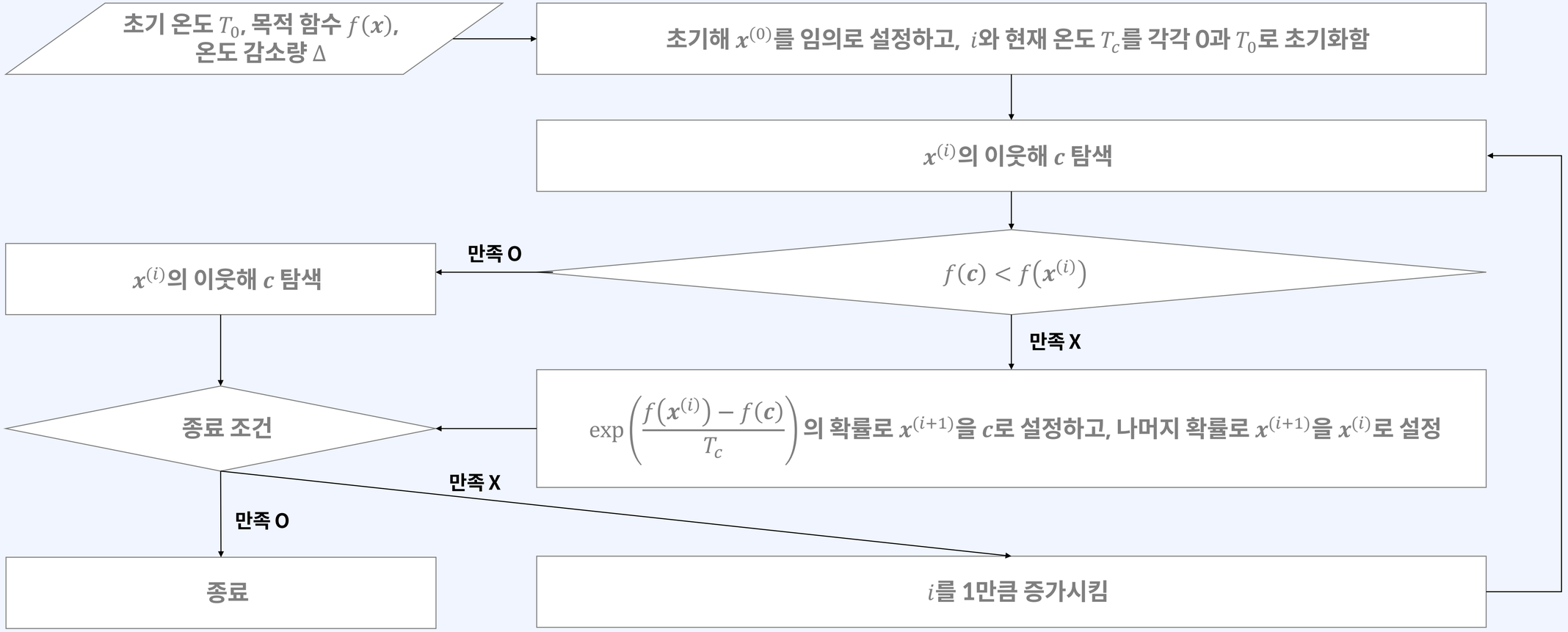
담금질 기법 영상
(Hill_Climbing_with_Simulated_Annealing.gif)

이 해법의 핵심 아이디어는 온도가 높을 때는 해가 개선되지 않는 방향도 높은 확률로 탐색하지만, 온도가 낮을 때는 해가 개선되는 방향으로만 탐색하는 것입니다. 즉, 알고리즘 초기에는 가능한 한 많은 해를 탐색하다가 점점 해가 개선되는 방향으로만 탐색함으로써 지역 최적을 탈피합니다.

담금질 기법
(계속)

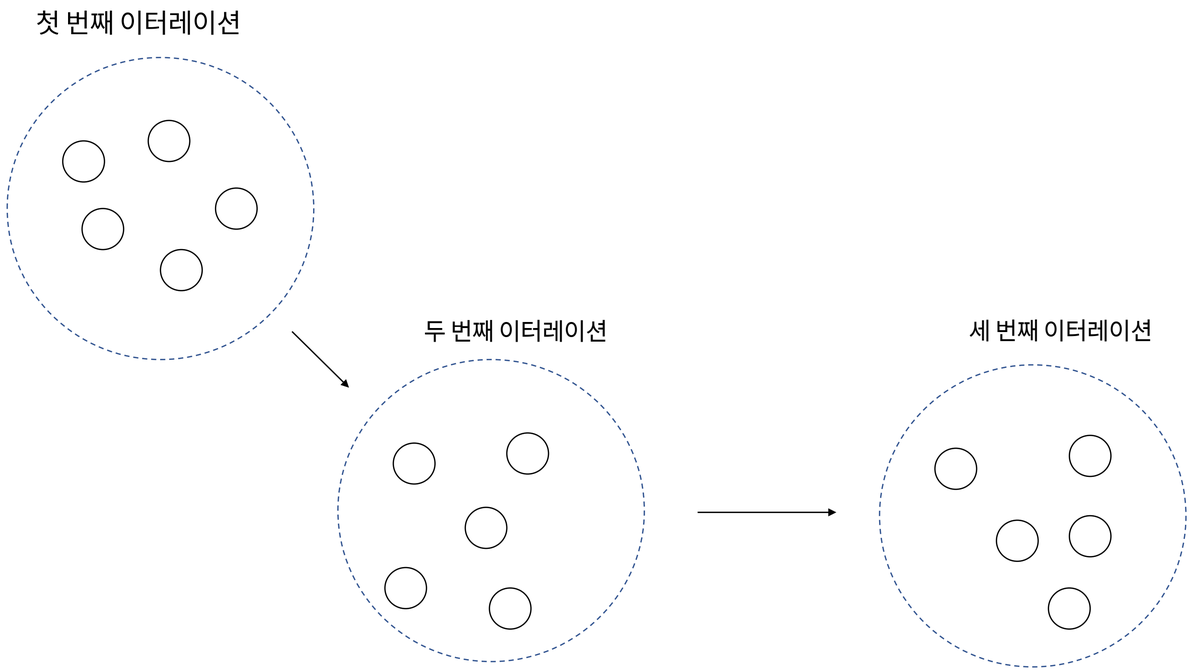
담금질 기법(simulated annealing)은 지역 최적에 빠지는 것을 방지하기 위한 휴리스틱입니다.

최소화 문제를 위한 담금질 기법



입자 군집 최적화

입자 군집 최적화는 새가 무리를 이뤄 나는 것처럼 여러 개의 해가 동시에 최적해를 찾아가는 휴리스틱 해법입니다.



- 작은 원과 큰 원은 각각 입자와 군집을 나타냄
- 여러 개의 입자가 하나의 군집을 이뤄 실행 가능 공간을 탐색함
- 이터레이션마다 각 입자의 위치와 속도가 정의됨
- 입자의 위치는 해를 나타내며 속도는 다음 이터레이션에서 입자의 위치를 계산하는 데 사용함

입자 군집 최적화 (계속)

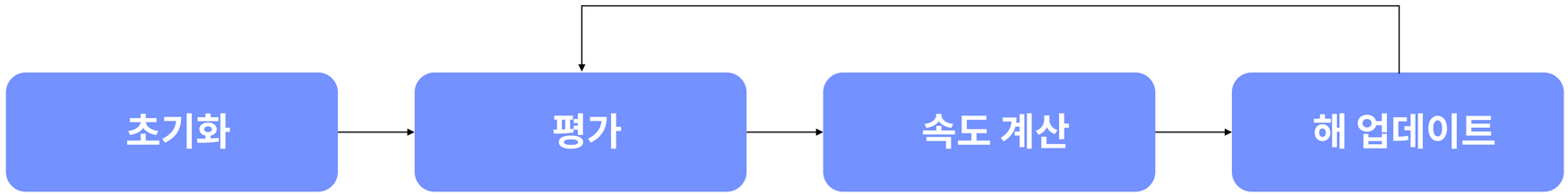
입자 군집 최적화는 새가 무리를 이뤄 나는 것처럼 여러 개의 해가 동시에 최적해를 찾아가는 휴리스틱 해법입니다.

해 업데이트

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}$$

- $\boldsymbol{x}_i^{(k)}$: k번째 이터레이션에서 $i(i = 1, 2, \cdots, n)$번째 입자의 위치
- $\boldsymbol{v}_i^{(k)}$: k번째 이터레이션에서 i번째 입자의 속도
- 각 입자의 위치와 속도는 서로 영향을 끼침

알고리즘의 구성



입자 군집 최적화 (계속)

입자 군집 최적화는 새가 무리를 이뤄 나는 것처럼 여러 개의 해가 동시에 최적해를 찾아가는 휴리스틱 해법입니다.

초기화: 모든 i에 대해 $\boldsymbol{x}_i^{(0)}$과 $\boldsymbol{v}_i^{(0)}$을 임의로 초기화

- $\boldsymbol{x}_i^{(0)}$는 실행 가능 공간에서 임의로 선택함
- $\boldsymbol{v}_i^{(0)}$는 사용자가 정한 하한 $\boldsymbol{v}_L$과 상한 $\boldsymbol{v}_U$ 사이에서 임의로 선택: $\boldsymbol{v}_i^{(0)} \sim U(\boldsymbol{v}_L, \boldsymbol{v}_U)$

평가: 모든 해를 목적 함수로 평가하고 최적 해 업데이트

- 입자 i가 찾은 최고 해 업데이트: $\boldsymbol{p}_i = \min_k f\left(\boldsymbol{x}_i^{(k)}\right)$
- 전체 최적 해 업데이트: $\boldsymbol{g} = \min_{i,k} f\left(\boldsymbol{x}_i^{(k)}\right)$

속도 계산

- $\boldsymbol{v}_i^{(k+1)} = \boldsymbol{v}_i^{(k)} + r_1\left(\boldsymbol{p}_i - \boldsymbol{x}_i^{(k)}\right) + r_2\left(\boldsymbol{g} - \boldsymbol{x}_i^{(k)}\right)$
- r_1과 r_2는 각각 개별 입자의 중요도와 군집의 중요도로 사용자가 설정한 상한인 ϕ_1와 ϕ_2를 바탕으로 샘플링: $r_1 \sim U(0, \phi_1), r_2 \sim U(0, \phi_2)$

해 업데이트

- $\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}$