

2024 Probabilistic Model Class

Modern Optimization with R

- Chapter 4. Local Search



순천향대학교 미래융합기술학과

Senseable AI Lab

석사과정 김병훈

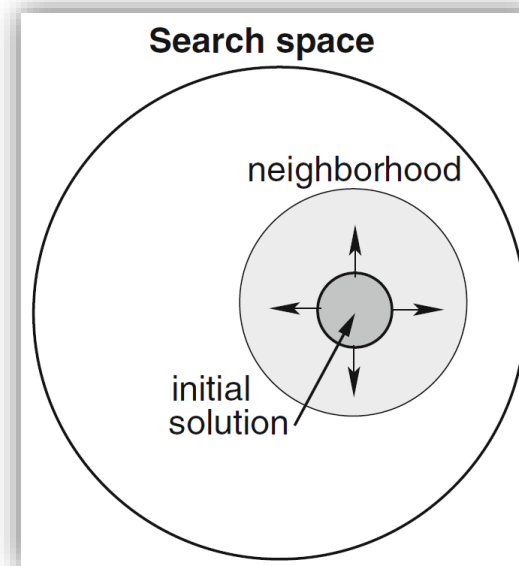
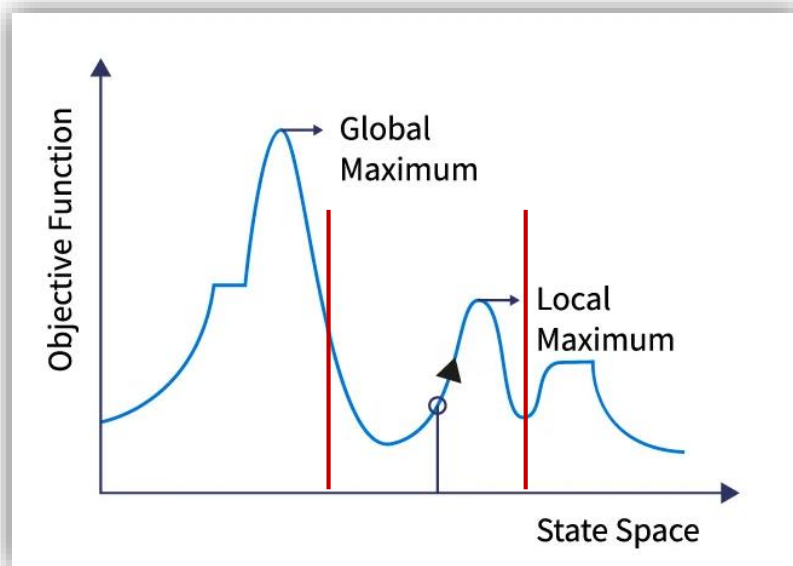
- 1 Introduction
- 2 Hill Climbing
- 3 Simulated Annealing
- 4 Tabu Search

1. Introduction

Local Search

Local Search

- 주변을 탐색하여 점진적으로 해를 개선하는 방식
 - 1) 특정 영역 내에서 최적의 해를 찾음
 - 2) 전역 최적 해를 찾는 것은 보장하지 않음
- 구동 원리
 - 초기 솔루션에서 시작하여 이웃 솔루션을 평가 -> 향상된 솔루션을 새로운 현재 솔루션으로 선택하는 과정을 반복

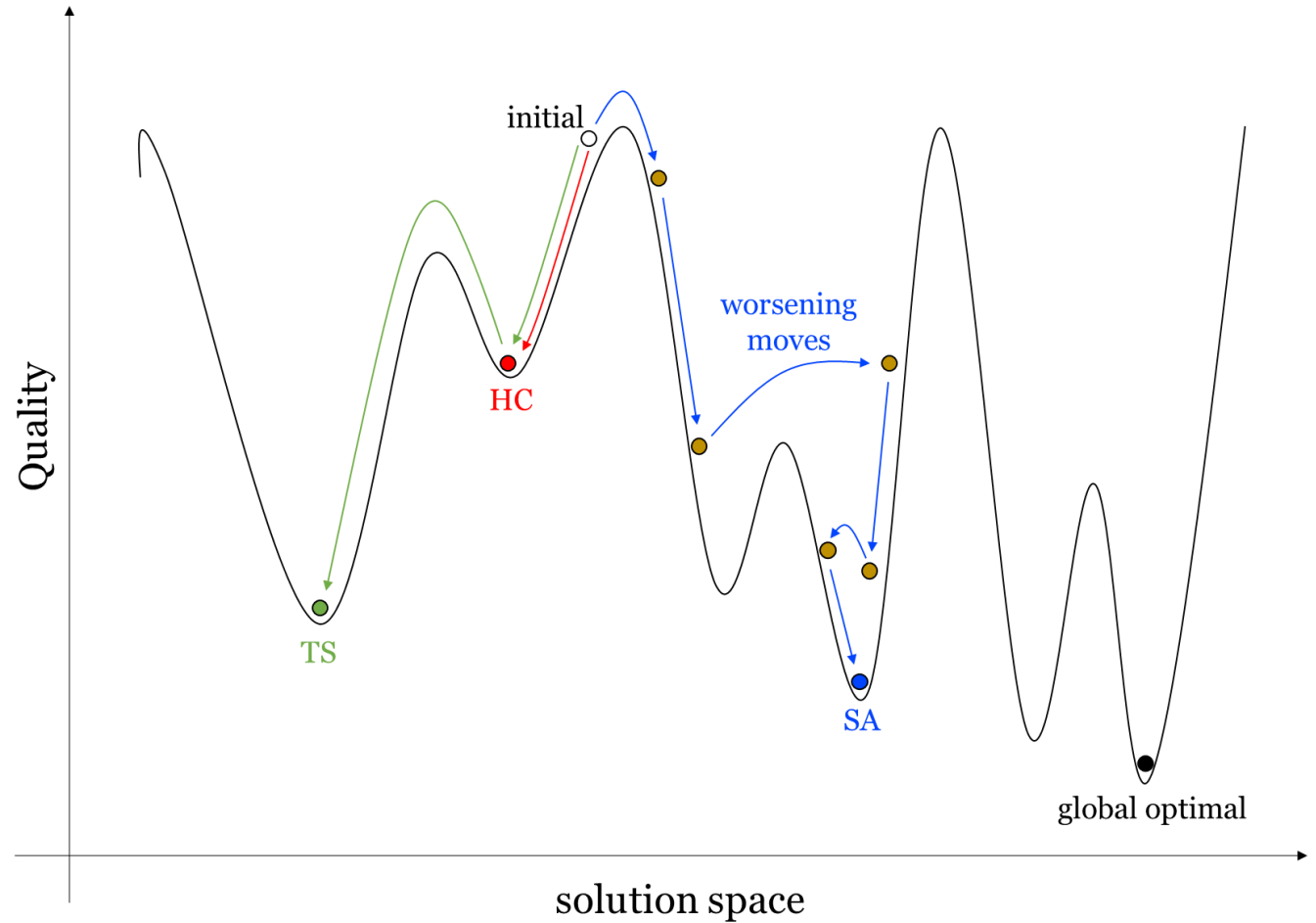


1. Introduction

Local Search

Local Search Algorithm 종류

- Hill-Climbing Search
- Simulated Annealing
- Tabu Search



2. Hill Climbing

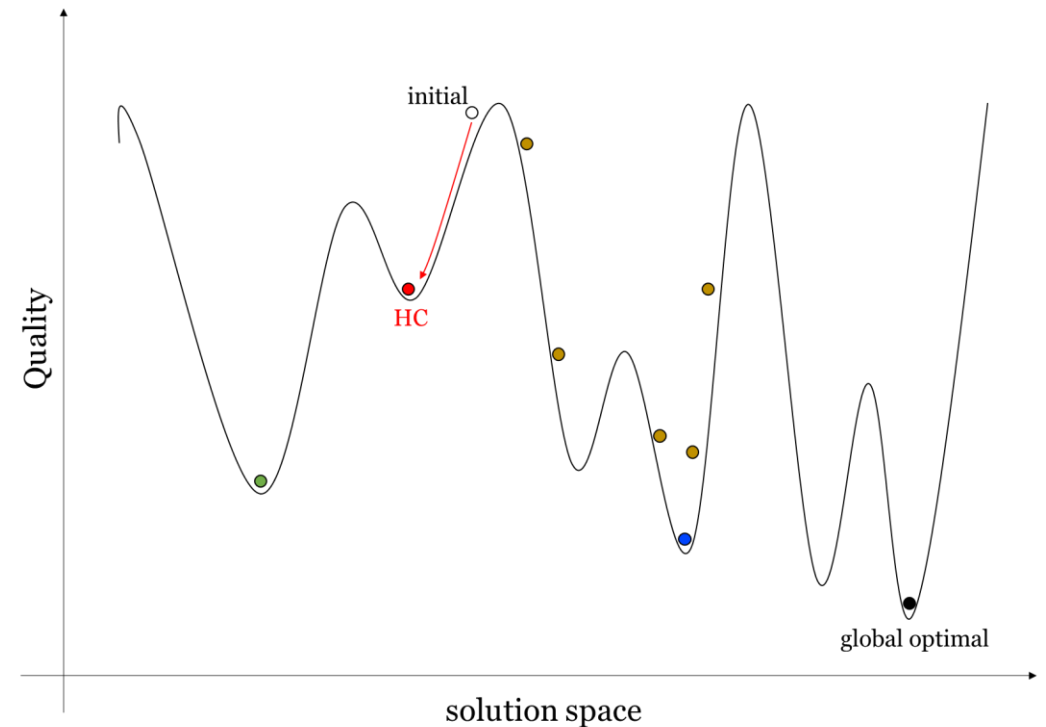
Define

Hill Climbing

- 주변 값을 탐색하여 점차적으로 최적의 해를 찾는 기법
- 현재 위치에서 주변 값을 탐색하여 후보 값들을 선정한 뒤, 그 중에서 하나를 고름

Algorithm 2 Pure hill climbing optimization method

```
1: Inputs:  $S, f, C$       ▷  $S$  is the initial solution,  $f$  is the evaluation function,  $C$  includes control parameters
2:  $i \leftarrow 0$                 ▷  $i$  is the number of iterations of the method
3: while not termination_criteria( $S, f, C, i$ ) do
4:    $S' \leftarrow \text{change}(S, C)$                                 ▷ new solution
5:    $B \leftarrow \text{best}(S, S', f)$                                 ▷ best solution for next iteration
6:    $S \leftarrow B$                                               ▷ deterministic select function
7:    $i \leftarrow i + 1$ 
8: end while
9: Output:  $B$                                               ▷ the best solution
```

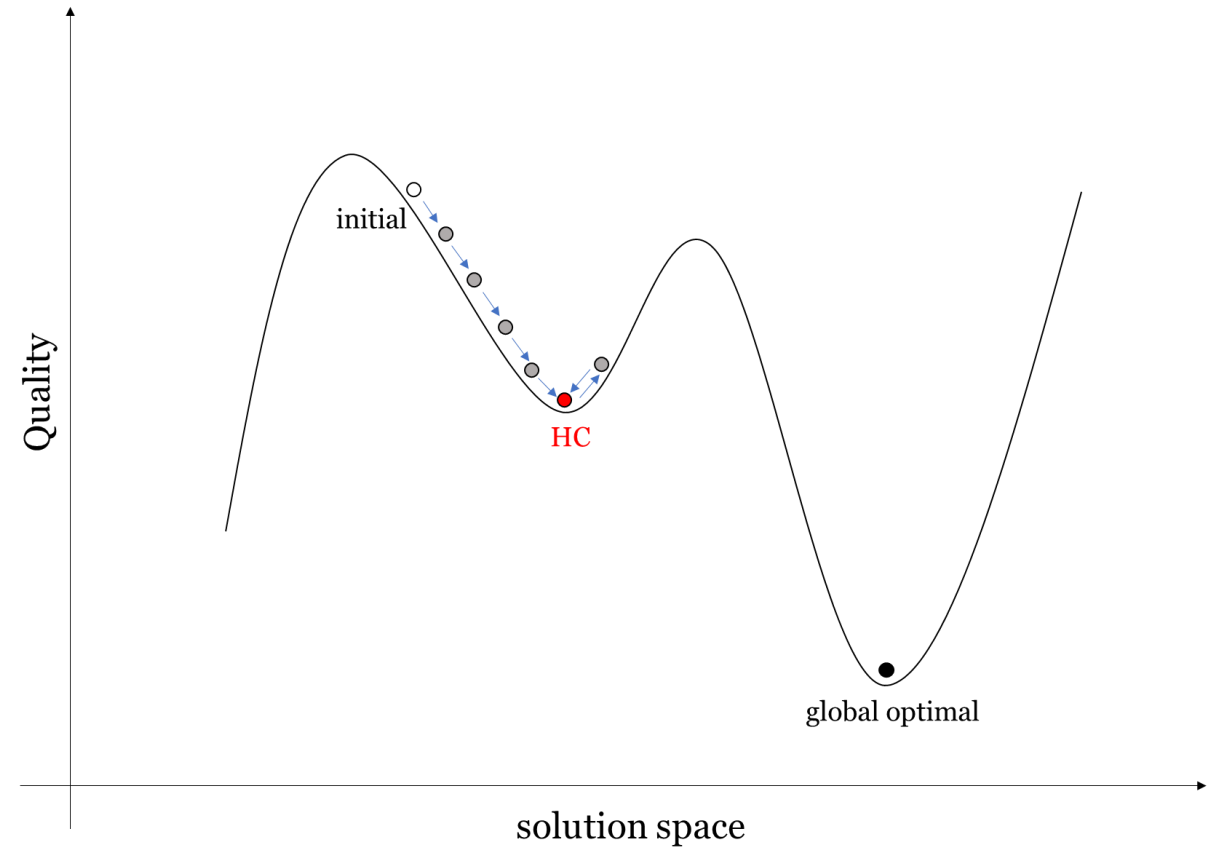


2. Hill Climbing

구동방식

Hill Climbing

1. Initial 설정
2. 주변 값 Search -> 목적함수 최소/최대화 값 선택
3. 이전 값과 비교
4. 업데이트
5. Iteration 만큼 반복



2. Hill Climbing

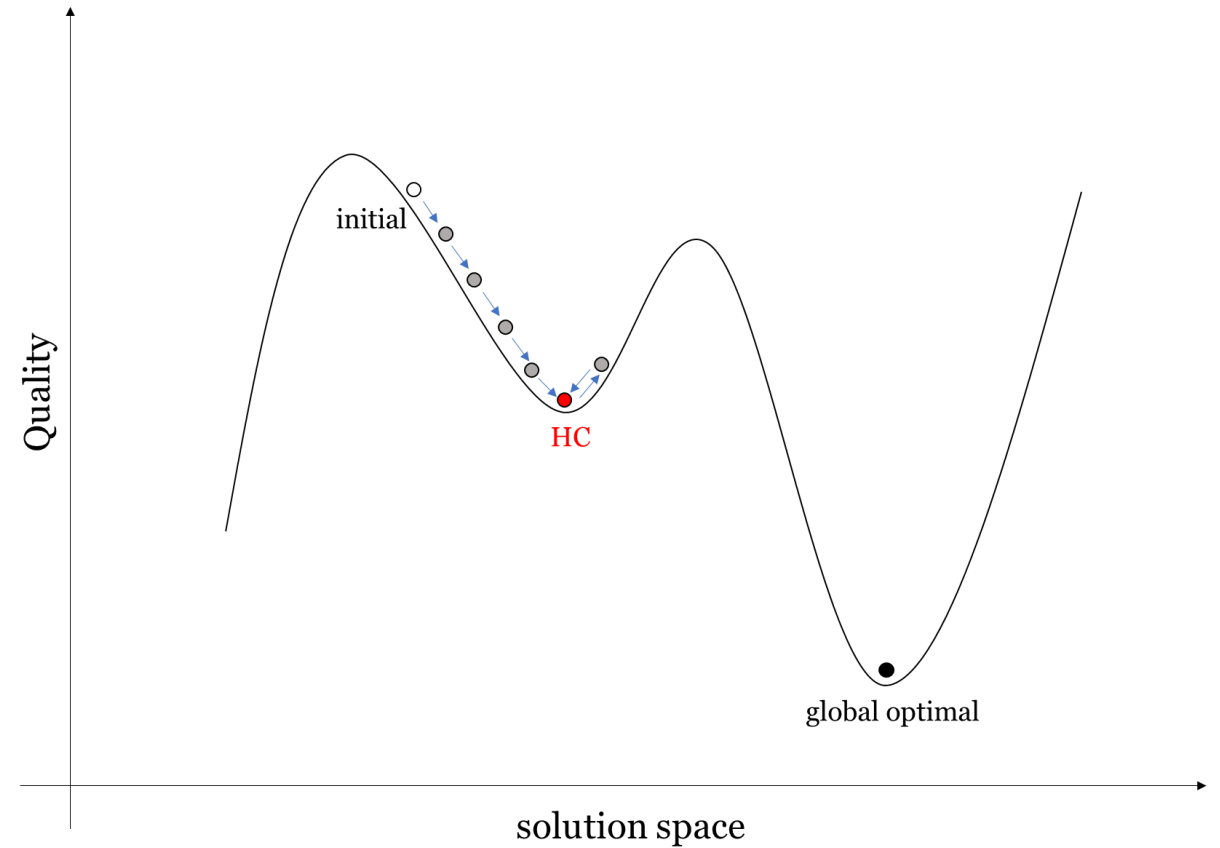
장단점

Hill Climbing 장점

- 메모리 사용량이 적음
- 합리적인 해 찾을 수 있음

Hill Climbing 단점

- Local optimum
- 주변 값과 현재 값이 동일하면 이동 멈춤

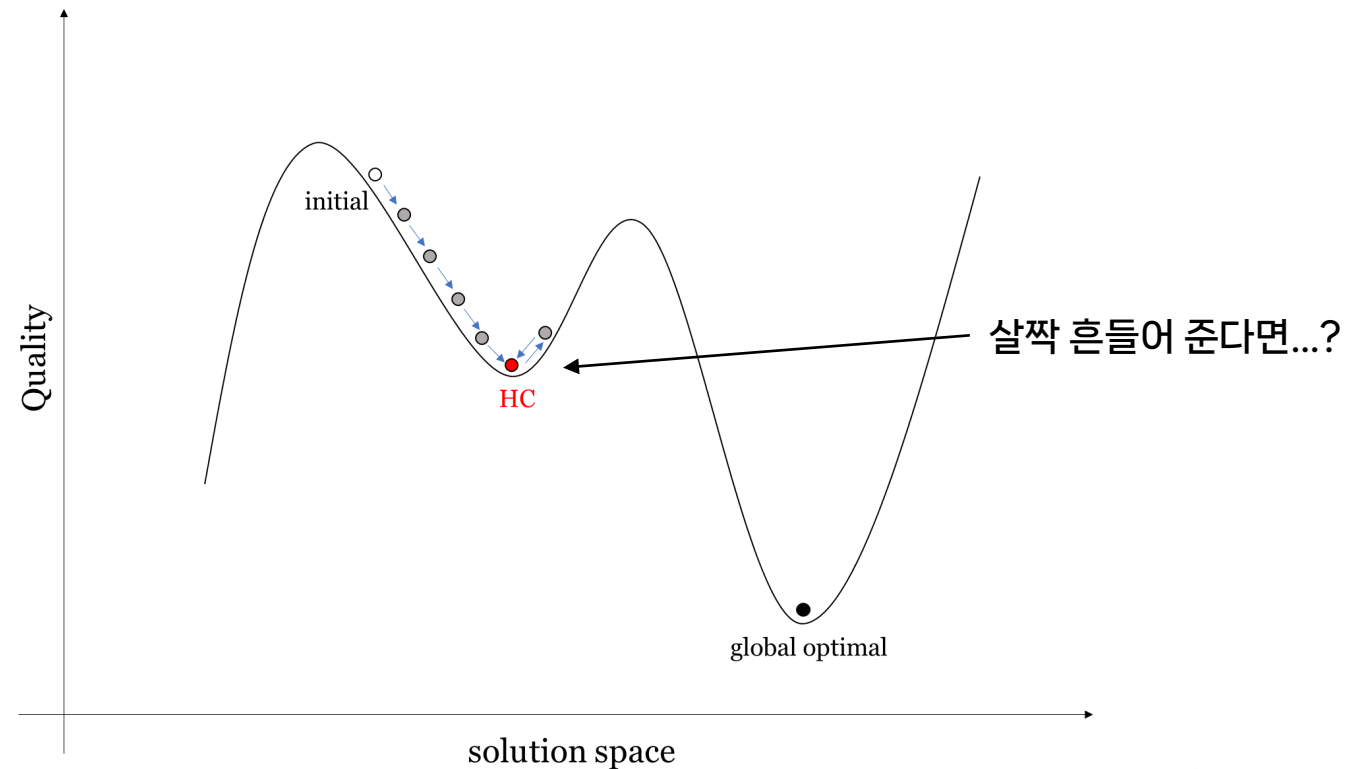


3. Simulated Annealing

Define

Simulated Annealing

- 확률적 휴리스틱 접근법 (휴리스틱 구현 부분 -> 언덕 흔들기)
- Hill-Climbing 기법은 현재 위치보다 높은 곳으로 이동하지 않기 때문에 이를 해결하기 위해 등장

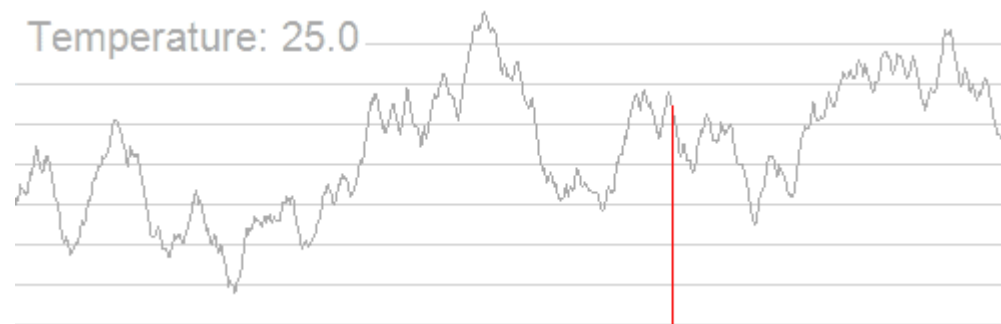


3. Simulated Annealing

구동방식

Simulated Annealing

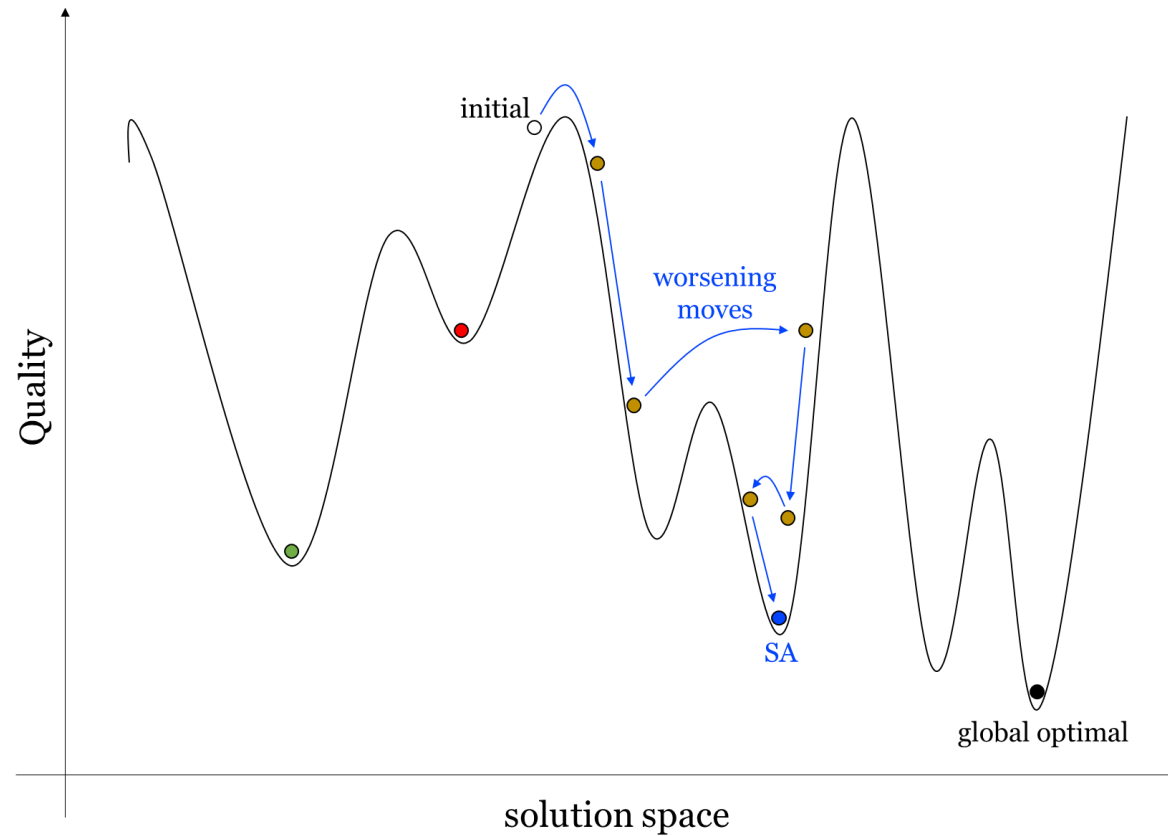
1. Initial 설정
2. 주변 값 Search -> 목적함수 최소/최대화 값 선택
3. 이전 값과 비교
 - 새로운 값이 원래 값보다 나쁘지만 acceptance probability 보다 적으면 새로운 값으로 바꿈
 - 처음에는 높은 확률로 이를 허가하지만, 점차 확률을 낮춤
4. 업데이트
5. Iteration 만큼 반복



3. Simulated Annealing

구동방식

Simulated Annealing



3. Simulated Annealing

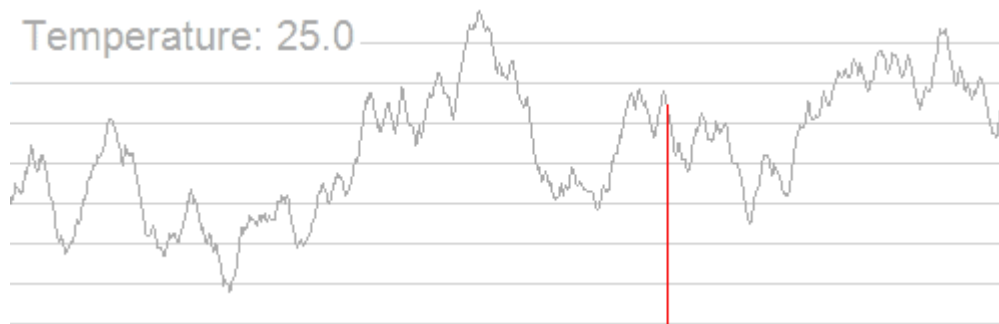
장단점

Simulated Annealing 장점

- 응용범위가 넓음
- 최적의 해를 찾을 수 있음

Simulated Annealing 단점

- 여전히 Local optimum
- 좋은 결과를 얻는데 걸리는 계산 시간 높음 -> 병렬 처리로 개선 가능

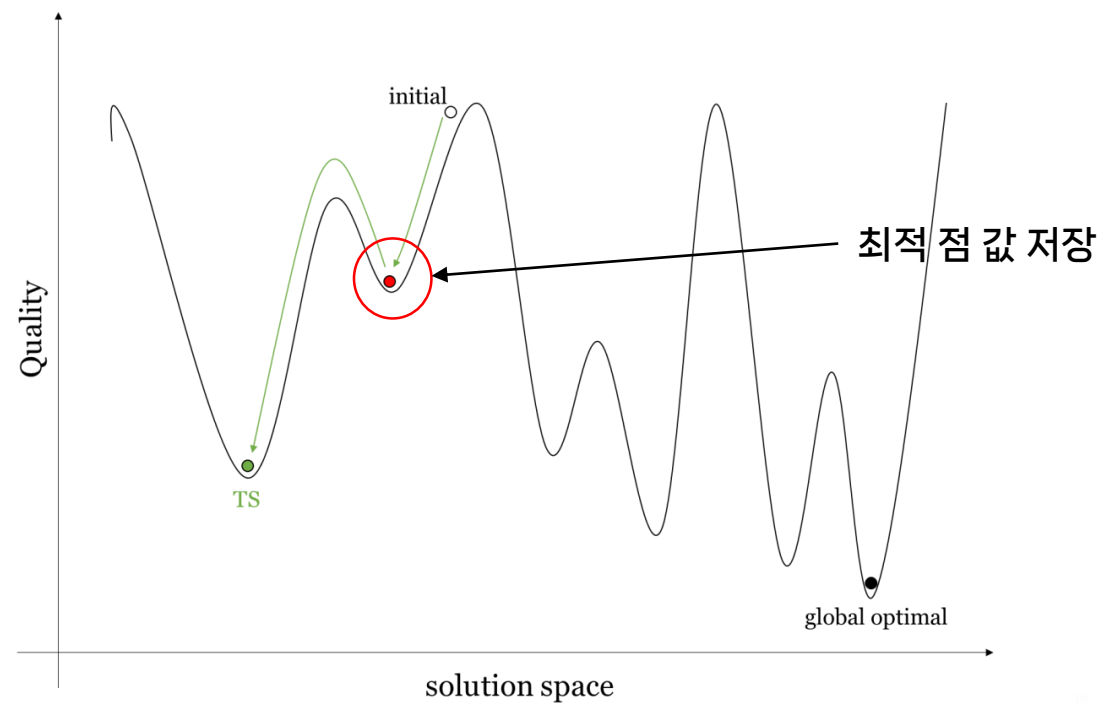


4. Tabu Search

Define

Tabu Search

- Tabu : 지역 최적 점에 대한 정보
- 메타 휴리스틱 접근법 (최적화 문제의 형태에 상관없이 주어진 최적화 문제를 풀 수 있음)
- 지역 최적 점에 대한 정보를 저장하여 이 정보를 기반으로 지역 최적점을 회피

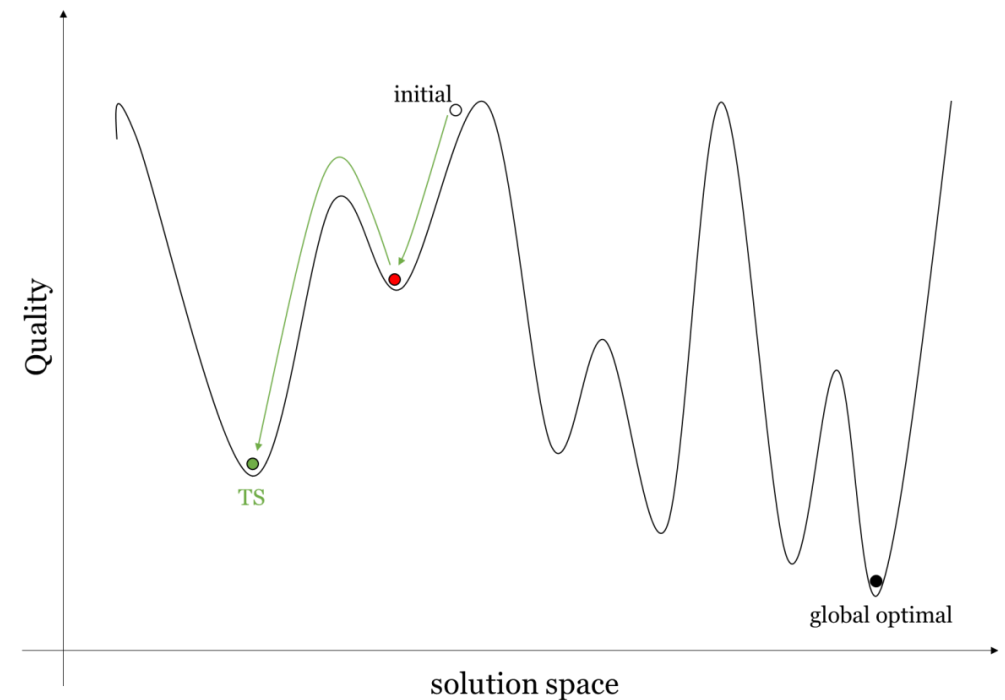


4. Tabu Search

Define

Tabu Search

1. Initial 설정
2. 주변 값 Search -> 목적함수 최소/최대화 값 선택
3. 이전 값과 비교
 - 새로운 값이 Tabu 이동인 지 확인
 - Tabu가 아니면 목적 함수 개선 여부 확인
 - 개선되면 값을 타부 목록에 추가, 아니면 이동
 - Tabu면 그대로 이동
4. 업데이트
5. Iteration 만큼 반복

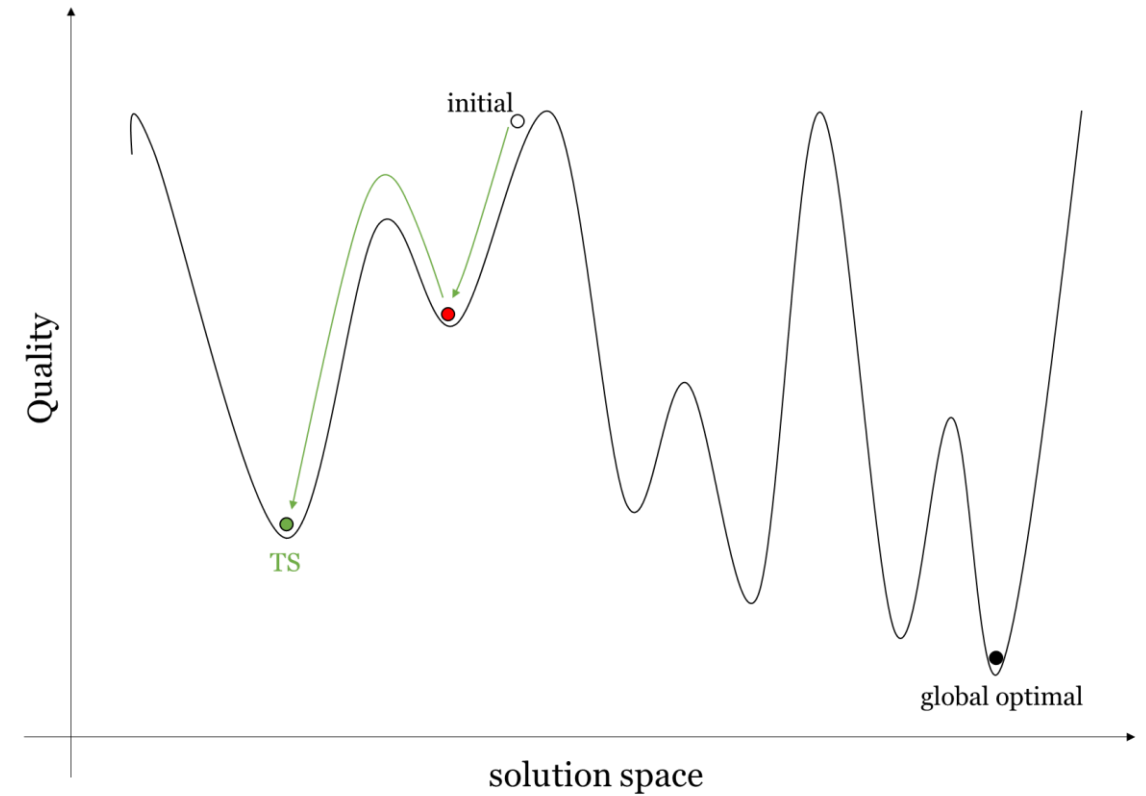


4. Tabu Search

장단점

Tabu Search 장점

- Local optimum에 빠질 가능성이 적음



Thanks

