

국지적 탐색 (Local Search)

한경수
성결대학교 컴퓨터공학과

Introduction



- 지금까지의 탐색은 탐색 공간을 체계적으로 탐색하도록 설계
 - 하나 이상의 경로를 메모리에 유지
 - 경로 상의 각 위치에서 어떤 대안이 탐색됐고 어떤 것이 미탐사 상태인지 기록
 - 목표가 발견되면 그 목표까지의 경로가 해가 됨
- 그러나, 목표까지의 경로(순서)는 중요치 않은 문제도 많음
 - 8-퀸, 통신망 최적화, 포트폴리오 관리

한경수

2

학습 목표



- 이론지식 응용 역량
 - 다음 국지적 탐색 알고리즘을 비교 설명할 수 있다.
 - 언덕 등반 탐색
 - Simulated Annealing
 - 국지적 빔 탐색
 - 유전 알고리즘
- 공학기술 및 도구 활용 역량
 - 국지적 탐색 알고리즘을 테스트 할 수 있다.

한경수

3

국지적 탐색

한경수

4

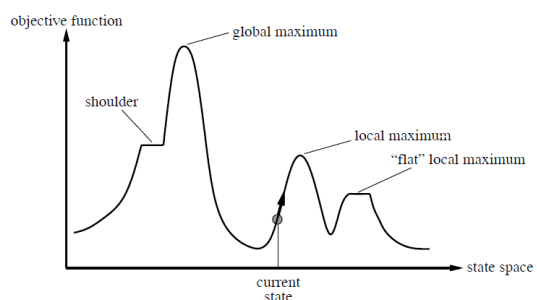
국지적 탐색(Local Search)

- 시작 상태에서 시작하여 인접한(neighboring) 상태로만 탐색
 - 경로들이나 도달 상태들을 기록하지 않음
- 장점
 - 극소량의 메모리 사용
 - 체계적 알고리즘이 적합하지 않은 대규모 혹은 무한 상태 공간에서도 국지적 탐색은 적절한 해를 찾을 수 있음
- 최적화 문제(optimization problem) 해결에 유용함
 - 최적화 문제: 목적 함수(objective function)에 따라 **최고 상태**를 찾는 게 목표

한경수

5

상태 공간 풍경(landscape)

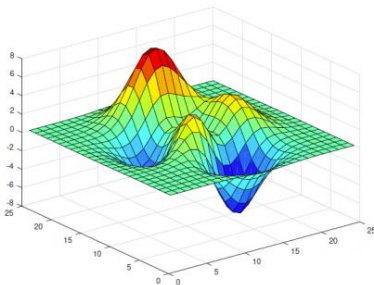


- Global maximum을 찾는 과정: **hill climbing**
- Global minimum을 찾는 과정: **gradient descent**

한경수

6

상태 공간 풍경



한경수

7

7



언덕 등반 탐색

단기 기억 상실증에 걸린 사람이 짝은 안개가 끼인 상태에서 에베레스트 산을 오르는 상황

한경수

8

8

언덕 등반 탐색

- hill-climbing search
- 하나의 현재 상태만 기억하고, **가장 높은 값을 갖는 인접 상태로 이동**
 - 최급상승(steepest ascent)하는 방향으로
- 더 높은 값을 갖는 이웃이 없는 봉우리에 도착했을 때 종료
- 탐색 트리를 유지하지 않음
- 현 노드의 자료구조
 - 상태와 목적 함수 값으로만 구성
- 현 상태의 인접한 이웃까지만 보고 **그 너머는 미리 보지 않음**
- 최급상승(steepest ascent)/ **최급강하(steepest descent) 언덕 등반**

한경수

9

9

언덕 등반 탐색

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
  current ← problem.INITIAL
  while true do
    neighbor ← current의 후속자 상태 중 목적 함수 값이 가장 큰 후속자
    if VALUE(neighbor) ≤ VALUE(current) then return current
    current ← neighbor
```

최급상승(steepest ascent) 언덕 등반

한경수

10

10

언덕 등반 탐색

```
function HILL-CLIMBING(problem) returns a state that is a local minimum
  current ← problem.INITIAL
  while true do
    neighbor ← current의 후속자 중 휴리스틱 함수 값이 가장 작은 후속자
    if h(neighbor) ≥ h(current) then return current
    current ← neighbor
```

최급강하(steepest descent) 언덕 등반

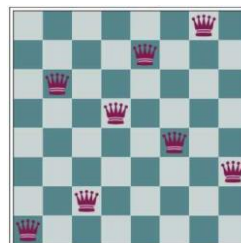
한경수

11

11

예: 8-퀸 문제

- 동일한 행, 열, 대각선에는 하나의 Queen 만 존재하도록 8개의 Queen을 배치



- 상태: 8개의 모든 퀸이 보드의 각 열마다 하나씩 위치한 퀸의 배치
- 초기 상태: 랜덤하게 선택된 하나
- 각 상태의 후속자는 하나의 퀸을 동일 칼럼에서 다른 위치로 이동시켜 생성할 수 있는 모든 가능한 상태
- 각 상태에는 $8 \times 7 = 56$ 개의 후속자가 있음

한경수

12

12

예: 8-퀸

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	13	13	16	13	16
17	16	18	15	14	16	16	16
18	14	13	15	15	14	16	16
14	14	13	17	12	14	12	18

h : 가로,세로,대각선에 마주치는 퀸 쌍의 개수

현 상태의 h 값은?

$h = 17$

최고 후속자: $h = 12$

최고 후속자 중 무작위로 하나 선택

한경수

13

언덕 등반 탐색

- 탐욕적 국지 탐색
 - 다음에 어디로 갈지에 대해 미리 고민하지 않고 **최고의 이웃** 상태로 이동
- 해결책 방향으로 **매우 빠르게 전진**
 - 나쁜 상태를 개선하기가 매우 용이하기 때문
- 그러나...

한경수

14

예: 8-퀸

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	13	13	16	13	16
17	16	18	15	14	16	16	16
18	14	13	15	15	14	16	16
14	14	13	17	12	14	12	18

$h = 17$

5번 움직임

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	13	13	16	13	16
17	16	18	15	14	16	16	16
18	14	13	15	15	14	16	16
14	14	13	17	12	14	12	18

$h = 1$

극소(local minimum)

언덕 등반 탐색은 **정체 상태에 빠지기 쉬움!**

한경수

15

정체 예: 8-퀸

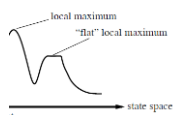
- 무작위로 생성된 8-Queens 상태에서 출발했을 때,
 - 문제의 14%만 해결 가능
 - 86%는 정체**
- 매우 빨리 동작함
 - 성공 시 평균 4번의 움직임
 - 실패 시 평균 3번
 - 총 상태 공간: $8^8 \approx 1,700$ 만 개의 상태

한경수

16

정체 상태 탈출

- 수평 이동 허용**
 - 평평한 극대라면 무한 루프!
 - 연속적인 수평 이동 횟수 제한
 - 예: 8-퀸에서 연속적인 수평 이동 횟수를 100회로 제한한 경우
 - 문제 해결 비율 14% \rightarrow 94%
 - 성공 시 평균 21번 움직임, 실패 시 64번



한경수

17

언덕 등반의 변종

- 확률적 언덕 등반**
 - stochastic hill climbing
 - 오르막 이동의 **경사도**에 따라 확률적으로 오르막 이동을 선택
 - 최급상승 언덕 등반**보다 일반적으로 **더 느리게 수렴**
 - 더 좋은 해를 발견하는 경우도 있음**

한경수

18

언덕 등반의 변종

- 제1선택 언덕 등반
 - first-choice hill climbing
 - 현 상태보다 더 좋은 후속자가 생성될 때까지 무작위로 후속자를 생성
 - 상태의 후속자가 많을 때 유용한 방법

한경수

19

언덕 등반

- 지금까지의 언덕 등반은 불완전 (incomplete)
 - 극대(극소)에서 정체될 수 있기 때문
- 완전한 언덕 등반은 불가능할까?



한경수

20

무작위 재시작 언덕 등반

- random-restart hill climbing
- 무작위로 생성된 초기 상태에서의 언덕 등반을 목표가 발견될 때까지 연속적으로 수행
- 1에 근접하는 확률로 **완전(complete)**
 - 목표 상태가 초기 상태로 생성될 때가 있음
- 각 언덕 등반 탐색의 성공 확률이 p 이면, 재시작 횟수의 기대값은 $1/p$

한경수

21

21

- 수평이동 허용한 경우:
- 문제의 14%만 해결 가능
 - 86%는 정체
 - 성공 시 평균 4번의 움직임
 - 실패 시 평균 3번
- 예: 8-퀸**
- 수평이동을 불허한 경우; $p \approx 0.14$
 - $1/0.14 = 7.14$; 약 7회 반복 필요(실패 6 + 성공 1)
 - 움직임 횟수의 기대값

$$= \frac{(\text{성공 횟수}) \times (\text{성공 시 움직임})}{(\text{성공 횟수}) \times (\text{성공 시 움직임}) + (\text{실패 횟수}) \times (\text{실패 시 움직임})}$$

$$= \frac{(\text{성공 시 움직임})}{(\text{성공 시 움직임}) + ((1-p)/p) \times (\text{실패 시 움직임})}$$

$$= \frac{4}{4 + (0.86/0.14) \times 3} \approx 22$$
 - 수평이동 허용한 경우; $p \approx 0.94$
 - 약 $1/0.94 \approx 1.06$ 회 반복 필요
 - 움직임 횟수 = $21 + (0.06/0.94) \times 64 \approx 25$
 - 8-퀸 문제에서 무작위 재시작 언덕 등반은 매우 효과적임; 300만 쿼 수초 내에 해결

한경수

22

22

무작위 재시작 언덕 등반

- 언덕 등반의 성공 여부는 상태 공간 지형에 매우 의존적임
- 매우 어려운 문제에서도 흔히 소수의 재시작만으로 상당히 좋은 극대(극소)를 찾을 수 있음

한경수

23

23

Simulated Annealing

모의 풀림; 모의 담금질; 담금질 기법

한경수

24

24

무작위 보행

- 무작위 보행(random walk)은 어떤가?



한경수

25

25

언덕 등반 vs. 무작위 보행

- 언덕 등반
 - 내리막 이동을 허용하지 않음 → 불완전
- 무작위 보행(random walk)
 - 후속자들 중 무작위로 하나를 선택하여 이동
 - 완전하지만 극히 비효율적
- 언덕 등반과 무작위 보행을 결합하여 **효율적이면서 완전한 방법은 없을까?**

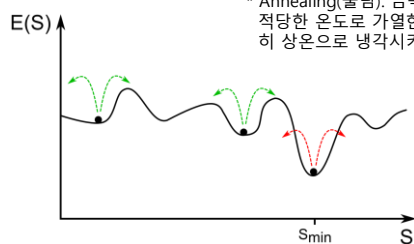
한경수

26

26

Simulated Annealing

- simulated annealing



* Annealing(풀림): 금속 재료를 적당한 온도로 가열한 다음 서서히 상온으로 냉각시키는 조작

- 초기에는 강하게, 서서히 강도 감소

한경수

27

27

Simulated Annealing

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  current ← problem.INITIAL
  for t = 1 to ∞ do
    T ← schedule(t)           schedule(t): 시간에 따라 온도 T값을 결정하는 함수
    if T = 0 then return current
    next ← current의 후속자 중 무작위로 선택
    ΔE ← VALUE(current) - VALUE(next)
    if ΔE > 0 then current ← next
    else current ← next only with probability eΔE/T
```

한경수

28

28

Simulated Annealing

- 무작위 이동
 - 그 이동이 상황을 개선시키면, 이동!
 - 개선시키지 않더라도, 1미만의 확률로 이동
 - 이 확률은 이동의 부적절성($|\Delta E|$)이 커짐에 따라, 또 온도(T)가 낮아짐에 따라 지수적으로 감소
 - 온도가 높은 초기에 부적절한 이동의 허용 가능성이 높았다가, 온도가 낮아지면서 가능성이 낮아짐
- 충분히 천천히 온도를 낮춘다면, **1에 근접한 확률로 최적(global optimum)을 찾을**

한경수

29

29

국지적 빔 탐색 (Local Beam Search)

메모리를 조금만 더 사용하자!

한경수

30

30

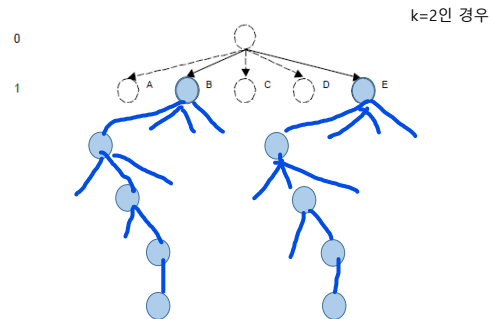
국지적 빔 탐색

- local beam search
- k 개의 상태를 유지
- 무작위로 k 개의 상태 생성
- **반복**
 - 모든 k 상태의 후속자들을 모두 생성
 - 그 중 목표가 존재하면 종료
 - 전체 리스트에서 k 개의 최고 후속자 선택/이동

한경수

31

국지적 빔 탐색 예



한경수

32

국지적 빔 탐색 vs. 무작위 재시작

- 국지적 빔 탐색은 무작위 재시작 탐색과 다른 것인가?



국지적 빔
X

한경수

33

국지적 빔 탐색

- k 개 상태의 **다양성이 부족함**
- 탐색 과정에서 상태 공간의 작은 영역으로 금방 집중되어 버림

→ 동결 → 정보공유 0

한경수

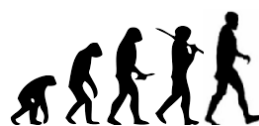
34

확률적 빔 탐색

- k 개 상태의 **다양성을 높이기** 위함
- k 개의 후속자를 **확률적**으로 선택
 - 선택 확률: 각 후속자의 값에 따라 증가하는 함수

한경수

35



유전 알고리즘 (Genetic Algorithm)

유전자 알고리즘

생물의 진화를 모방

- 환경에 잘 적응하는 개체들은 튼튼한 자식을 남길 확률이 높음
- 세대교체가 진행됨에 따라 전체적으로 나은 개체 생성

한경수

36

유전 알고리즘

- genetic algorithm; GA
- 각 상태(개체; individual)는 유한 문자(예: 0, 1) 스트링으로 표현
- 2개의 부모 상태를 결합하여 후속자 상태를 생성함; 확률적 빔 탐색의 변종
- k 개 초기 상태(개체군; population)를 무작위로 생성
- 상태들의 다음 세대 생성
 - 각 상태는 목적 함수 혹은 적응도 함수(fitness function)로 평가함
 - 적응도 점수에 따라 확률적으로 선택됨

한경수

37

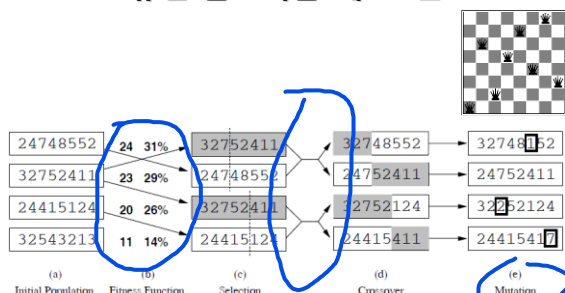
유전 알고리즘

- 선택(selection)
 - 확률에 따라 생식을 행할 개체 쌍을 선택
- 교차(crossover)
 - 각 개체 쌍에 대해 교차점이 무작위로 선택됨
 - 부모 스트링을 교차점에서 교차시켜 후손 생성
 - 개체군이 초기에는 매우 다양함
 - 교차는 탐색 초기과정에 상태공간에서 큰 움직임을 취함
 - 대부분 개체가 매우 유사해지게 되면 움직임이 작아짐
- 돌연변이(mutation)
 - 작은 확률로 후손 스트링의 일부 값을 변경

한경수

38

유전 알고리즘 예: 8-퀸



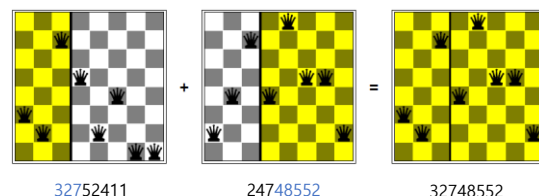
- 스트링 표현: 각 칼럼에서의 위치 (1..8)
- 적응도 함수: 공격받지 않는 퀸 쌍 개수; 해결책은 적응도 함수 값이 28

한경수

39

유전 알고리즘 예: 8-퀸

- 교차



한경수

40

유전 알고리즘

- 오르막 이동, 무작위 탐사, 병렬적인 탐색 쓰레드 사이의 정보 교환 등을 결합
- 주요 장점은 교차 연산으로부터 나옴
 - 초기 스트링 코드가 무작위로 나열되면 장점 없음
 - 독자적으로 유용한 기능을 수행토록 진화한 스트링 블록들을 결합하여 빠른 탐색이 가능토록 함
 - 예: 246*****
- 부분 스트링이 해결책의 의미 있는 구성요소에 대응될 때 유전 알고리즘은 최상의 성능을 발휘함
 - 각 개체의 표현 방법에 주의해야 함

한경수

41

정리



- 국지적 탐색
 - 언덕 등반 탐색
 - Simulated Annealing
 - 국지적 빔 탐색
 - 유전 알고리즘

한경수

42



정리 문제 풀이

한경수

43

정리 문제 1: 국지적 탐색

- 다음은 각각 어떤 탐색 알고리즘에 해당하는가?
 - 국지적 빔 탐색($k = 1$)
 - 국지적 빔 탐색
 - 초기 상태는 하나
 - 유지하는 상태 개수는 무제한
 - Simulated Annealing($T = \infty$ 로 고정)
 - 유전 알고리즘(개체군 크기 = 1)

한경수

44