

타부 서치(Tabu Search) 이론과 표 채우기

타부 서치는 **로컬 서치(Local Search)**의 일종으로, 로컬 최적해를 피하기 위해 최근 탐색한 해를 금지(Tabu) 리스트에 저장하여 탐색을 제어합니다. 이는 주로 조합 최적화 문제(예: 8-Queens, TSP)에 적합합니다.

타부 서치 작동 원리

1. 초기 해 생성: 임의의 초기 해(예: 8-Queens의 초기 퀸 배치).
2. 이웃 해 생성: 현재 해에서 연산자(예: 퀸 위치 교환)로 이웃 해 생성.
3. 목적 함수 평가: 이웃 해의 비용(예: 충돌 쌍 수) 계산.
4. 타부 리스트 관리:

최근 방문한 해 또는 연산자를 타부 리스트에 추가.

타부 리스트는 고정 크기(FIFO로 오래된 항목 제거).

4 / 6

알고리즘해결설명_이론표_표채우기문제_시험대비용요약.md

2025-10-19

타부 항목은 일정 반복 동안 선택 금지.

5. 최적 이웃 선택: 타부 아닌 이웃 중 목적 함수 값이 가장 좋은 해 선택.
6. Aspiration Criteria: 타부 항목이라도 현재 최적해보다 나으면 선택 가능.
7. 종료 조건: 최대 반복 횟수, 목표 비용 도달 등.

타부 서치 표 채우기 (시험 대비)

시험에서 자주 나오는 타부 서치 관련 표는 알고리즘 비교 또는 탐색 과정 추적 형식입니다. 아래는 예시입니다.

예시 문제: 타부 서치로 8-Queens 문제 해결

초기 해: [3,1,7,4,8,2,6,5], 비용(충돌 쌍 수)=3.

연산자: 두 퀸의 행 위치 교환.

타부 리스트 크기: 2.

이웃 해: [1,3,7,4,8,2,6,5], [3,7,1,4,8,2,6,5], ...

목표: 충돌 쌍 수 = 0.

단

계

1

2

3

현재 해

비

용

[3,1,7,4,8,2,6,5]

3

[3,7,1,4,8,2,6,5]

1

[7,3,1,4,8,2,6,5]

2

이웃 해들 (비용)

선택된 해

타부 리

스트

비고

1,3,7,4,8,2,6,5,

3,7,1,4,8,2,6,5

7,3,1,4,8,2,6,5,

3,1,7,4,8,2,6,5

3,7,1,4,8,2,6,5,

7,1,3,4,8,2,6,5

[3,7,1,4,8,2,6,5]

[(3,1)]

비용 1 선택

[7,3,1,4,8,2,6,5]

[7,1,3,4,8,2,6,5]

[(3,1),

(3,7)]

[(3,7),

(7,3)]

(3,1) 타부, 비용 2

선택

비용 0 달성, 종료

총돌 쌍 계산: 예: [3,1,7,4,8,2,6,5] → 대각선 총돌 (3,1), (7,4), (8,5) = 3쌍.

타부 리스트: 교환된 위치 쌍 (예: (3,1))을 기록, 재교환 방지.

시험용 암기 포인트

타부 서치 특징:

로컬 최적 탈출: 타부 리스트로 최근 해 금지.

유연성: Aspiration Criteria로 타부 무시 가능.

메모리 기반: 단기/장기 메모리 활용.

장점: Hill-Climbing보다 로컬 최적 회피 우수.

단점: 타부 리스트 크기 설정, 계산 비용 증가.

적용 문제: 8-Queens, TSP, 배낭 문제 등.

시험 대비 요약 (객관식/주관식/T-F 대비)

객관식 예시

1. 8-Puzzle에서 A 알고리즘의 휴리스틱으로 적합한 것은?*

(A) Manhattan 거리 (O)

5 / 6

알고리즘해결설명_이론표_표채우기문제_시험대비용요약.md

2025-10-19

(B) 타일 개수 (O, but less effective)

(C) 무작위 값 (X)

(D) 경로 비용 (X)

2. 타부 서치의 주요 특징은?

(A) 최근 방문 해를 금지 (O)

(B) 최적해 보장 (X)

(C) BFS 기반 (X)

(D) 무작위 탐색 (X)

주관식 예시

8-Queens 문제의 순열 기반 모델을 설명하시오.

답: 각 열에 퀸 1개를 배치, 결정 변수 x_i 는 i 열의 행 위치 (1~8). 제약 조건은 각 행에 퀸 1개, 대각선 충돌 없음 ($|x_i - x_j| \neq |i - j|$). 목적 함수는 충돌 쌍 수 최소화(0). 연산자는 퀸의 행 이동 또는 두 퀸의 행 교환.

타부 서치의 작동 원리를 설명하시오.

답: 초기 해에서 이웃 해 생성, 목적 함수로 평가. 타부 리스트로 최근 연산자/해 금지, 최적 이웃 선택. Aspiration Criteria로 타부 무시 가능. 반복하며 최적 해 탐색.

T/F 문제 예시

1. BFS는 최적해를 보장한다. (T, 깊이 기준)
2. A Search는 $h(n)$ 이 과대추정 시 최적해를 보장한다.* (F, admissible 필요)
3. 타부 서치는 로컬 최적을 완전히 피할 수 있다. (F, 개선 가능하나 완전 피하기 어려움)

초심자를 위한 학습 팁

1. 문제 정의 연습: 각 문제의 상태 공간, 연산자, 목적 함수, 제약 조건을 명확히 정리.
2. 알고리즘 비교 암기: 위 표로 uninformed/informed 알고리즘의 성능(완전성, 최적성, 복잡도) 비교.
3. 타부 서치 시뮬레이션: 8-Queens 같은 간단한 문제로 표 채우기 연습.
4. 휴리스틱 설계: 8-Puzzle(Manhattan 거리), 최단 경로(직선 거리) 등 문제별 휴리스틱 이해.
5. 작은 예제 풀이: 4-Queens, 3도시 TSP 등으로 손으로 탐색 트리 그려보기.

이 표와 설명을 바탕으로 시험 준비하시면 객관식, 주관식, T/F 문제에 대비할 수 있습니다. 추가로 특정 문제나 알고리즘에 대해 더 깊이 알고 싶으시면 말씀해주세요!

6 / 6