

AI 알고리즘

발견적 해법

메타휴리스틱기법

메타휴리스틱기법

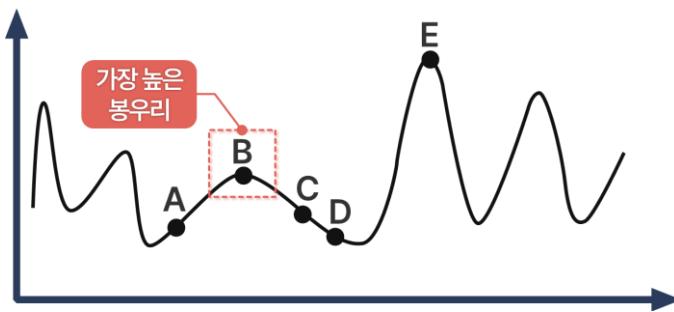
메타휴리스틱기법

❖ 메타휴리스틱해법(Meta-Heuristic Method)

- 블랙박스 접근방법이라고도 함
 - 가이드라인
 - 구성요소들을 정의하고 운영방안을 제설계

❖ 시뮬레이티드 어닐링(Simulated Annealing)

- 상당수의 발견적 해법은 유사한 단계를 반복적으로 수행하는 특징을 가짐
- 반복적인 단계를 수행한다는 것은 해를 찾아나가는데 있어서 점진적인 개선을 한다는 것을 의미
- 현재의 해와 유사한 이웃해의 집합을 정의하고 이 중 가장 성과가 좋은 이웃해로 현재 해를 이동시킴으로써 점진적인 개선을 얻는 방법을 지역 탐색 기법이라고 함
- 지역을 먼저 탐색하는 것



- 당장은 손해를 보더라도, 범위를 더 넓게 갈 수 있는 방법을 허용하는 것
- 지역 탐색 기법이 국소 최적해에 빠지는 문제점 해결 방안
 - 현재 해가 현재보다 열등한 이웃해로 이동하는 것을 확률적으로 허용
- 쇠에 높은 열을 가하여 액체로 만든 다음 냉각시키면서 다시 고체로 변환시키는 과정을 통해 더욱 단단한 쇠를 얻는 것
 - 쇠를 담금질(Annealing) 하는 과정과 유사하여 붙여진 이름

메타휴리스틱기법

❖ 시뮬레이티드 어닐링 해법 알고리즘

- 단계0

 - 초기화

실행 가능한 초기해 X_0 와 초기온도 T_0 를 설정하고, 알고리즘 변수 $k=0, i=0$ 을 초기화 현재까지 찾은 최선해 X^* 는 초기해인 X_0 , 최선해의 목적함수 값인 Z^* 는 초기해의 목적함수 값인 $Z(X_0)$ 로 정의

- 단계1

 - 이웃해 생성

X_k 의 실행 가능한 이웃해를 생성

- 단계2

 - 해이동

2.1 생성된 이웃해 중 Y 를 선택

2.2 만약 $Z(Y) < Z(X_{-K})$ 이거나 랜덤 값 $R \leq e^{-(Z(X_{-k}) - Z(Y))/T}$ 이면 현재해를 이동

즉, $X_{-(k+1)} = Y$ 로 설정하고, 단계3으로 2.3 아니면 현재해를 유지. 즉, $X_{-(k+1)} = X_{-k}$ 로 설정하고, $k=k+1$ 로 변경한 후 단계1로.

- 단계3

 - 해이동 최선해 변경 및 냉각과정

3.1 만약 $Z(X_{-(k+1)}) < Z^*$ 를 만족하면, $X^* = X_{-(k+1)}$ 로, $Z^* = Z(X_{-(k+1)})$ 로 변경

3.2 $i=i+1$ 로 변경한 후 적절한 냉각규칙에 의하여 온도 $T=T_{-i}$ 로 설정

3.3 만약 알고리즘 종료 조건에도 달(온도 T 가 사전에 정의된 온도 이하로 떨어지면) 하면 X^* 를 최종해로 선택하고 알고리즘을 종료 3.4 아니면 $k=k+1$ 로 변경한 후 단계1로

메타휴리스틱기법

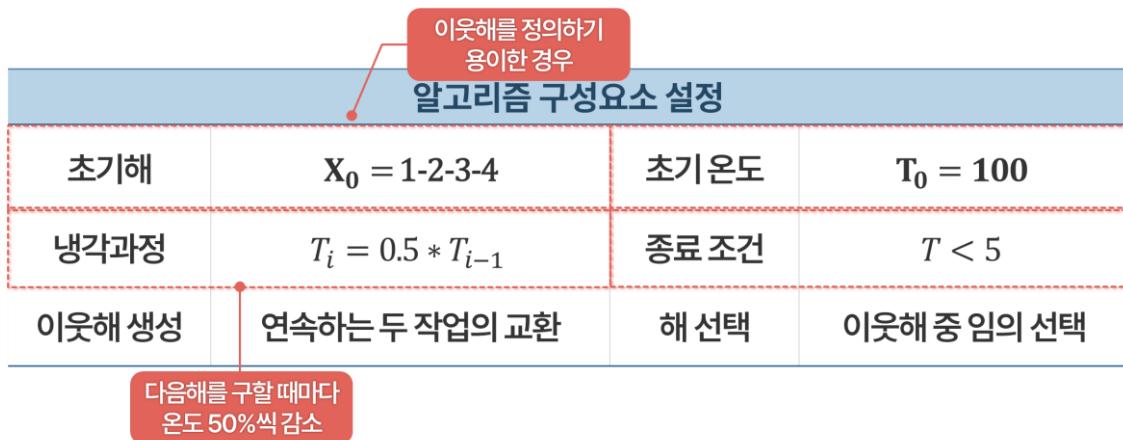
❖ 시뮬레이티드 어닐링 해법

- 이웃해의 목적함수값이 현재의 목적함수값보다 열등한 경우
 - 일정한 확률을 구함
- 일반적으로 많이 사용되는 기준은 메트로폴리스 기준($e^{\frac{Z(x)-Z(Y)}{T}}$)
 - 현재해의 목적함수값=90
 - 선택된 이웃해의 목적함수값=100
 - T의 값=45
- 메트로폴리스 기준에 의하면 현재해가 이웃해로 이동할 확률은 $e^{\frac{90-100}{45}}$ 이므로 0.8
- 만약 냉각과정을 거쳐 T의 값이 45가 아닌 100이되었다면, 동일한 상황에서 이웃해로 이동할 확률은 $e^{\frac{90-100}{100}}$ 이므로 0.368로 감소
- 초기화
 - 발견적 기법이므로 어떤 해를 초기해로 선정하느냐에 따라서 기법의 성능이 크게 변화
- 이웃해
 - 현재해가 한 번의 이동으로 도달할 수 있는 해가 이웃해에 해당
 - 이웃해를 정의하는 것은 이동을 정의하는 것과 동일
 - 동일한 의사결정문제라고 하더라도 이웃해는 다양하게 정의될 수 있음
 - 이웃해의 정의에 따라 해법의 성능이 크게 좌우되므로 적절한 이웃해의 정의가 필수적임
- 냉각과정
 - 파라미터값은 초기에는 높게 잡고 시간이 흐름에 따라 점차 낮추어감
 - 주요 설계요소는 온도를 어떤 속도로 낮추는지, 어떤 기준으로 낮추는지 등
- 의사결정문제마다 새롭게 결정해주어야 하며 의사결정문제의 특성을 고려하여 설계되어야 함
- 예제
 - 목적은 네 가지 작업의 납기지연시간(Tardy Time)의 총합을 최소화하는 것

작업 j	가공시간 p_j	납기 d_j
1	13	30
2	10	35
3	8	20
4	10	25

메타휴리스틱기법

❖ 시뮬레이티드 어닐링 해법



반복 <i>k</i>	탐색해 X_k $Z(X_{k+1})$	냉각 <i>i</i>	온도 T_i	이웃해	선택해 Y $Z(Y)$	랜덤값 <i>R</i>	$e^{\frac{Z(x)-Z(Y)}{T}}$	해 이동	최선해
0	1-2-3-4 27	0	100	2-1-3-4 27 1-3-2-4 17 1-2-4-3 29	1-3-2-4 17			채택	17
1	1-3-2-4 17	1	50	3-1-2-4 16 1-2-3-4 27 1-3-4-2 13	1-3-4-2 13			채택	13
2	1-3-4-2 13	2	25	3-1-4-2 12 1-4-3-2 17 1-3-2-4 17	3-1-4-2 12			채택	12
3	3-1-4-2 12	3	12.5	1-3-4-2 13 3-4-1-2 7 3-1-2-4 16	3-4-1-2 7			채택	7
4	3-4-1-2 7	4	6.25	4-3-1-2 7 3-1-4-2 12 3-4-2-1 11	4-3-1-2 7	0.7	1	채택	7
5	4-3-1-2 7	5	3.13		종료조건 $T < 5$				

메타휴리스틱기법

❖ 시뮬레이티드 어닐링 해법



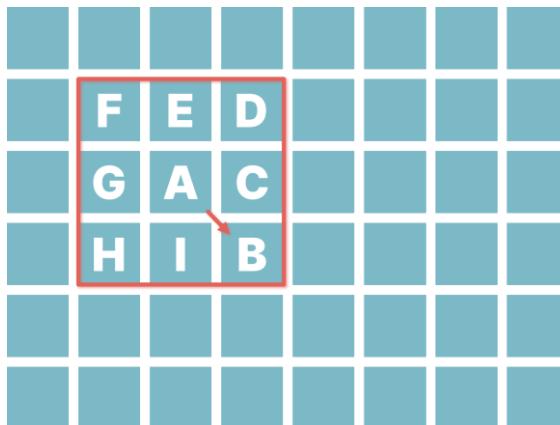
- 네 가지 작업의 납기지연시간(Tardy Time)의 총합을 최소화하는 것

반복 k	탐색해 X_k $Z(X_{k+1})$	냉각 i	온도 T_i	이웃해	선택해 Y $Z(Y)$	랜덤값 R	$e^{\frac{Z(x)-Z(Y)}{T}}$	해 이동	최선해
0	1-2-3-4 27	0	100	2-1-3-4 27 1-3-2-4 17 1-2-4-3 29	1-3-2-4 17			채택	17
1	1-3-2-4 17	1	50	3-1-2-4 16 1-2-3-4 27 1-3-4-2 13	1-3-4-2 13			채택	13
2	1-3-4-2 13	2	25	3-1-4-2 12 1-4-3-2 17 1-3-2-4 17	3-1-4-2 12			채택	12
3	3-1-4-2 12	3	12.5	1-3-4-2 13 3-4-1-2 7 3-1-2-4 16	3-4-1-2 7			기준값 보다 작으면 그 값으로 이동	7
4	3-4-1-2 7	4	6.25	4-3-1-2 7 3-1-4-2 12 3-4-2-1 11	4-3-1-2 7	0.7	1	채택	7
5	4-3-1-2 7	5	3.13		종료조건 $T < 5$				

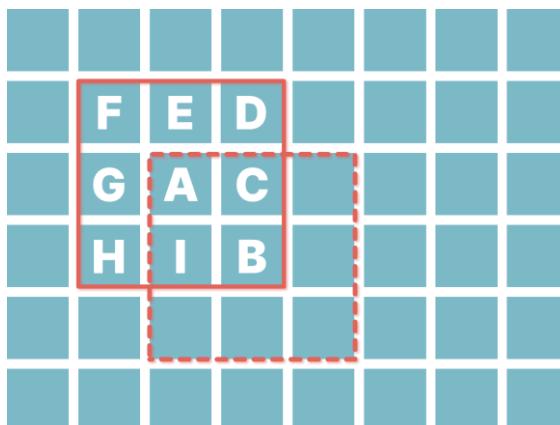
메타휴리스틱기법

❖ 타부 탐색(Tabu Search)

- 지역탐색 기법이 가진 문제점을 해결하기 위해 개발된 메타휴리스틱기법
- 지역탐색 기법에 해당
- 현재해의 모든 이웃해들에 대하여 목적함수값을 계산한 후 이 중 가장 우수한 목적함수값을 제공하는 이웃해로 현재해를 이동
- 현재해의 이동
 - 현재해: A
 - 실행 가능한 이웃해: B, C, D, E, F, G, H, I
 - 이웃해 중 가장 우수한 해: B



- 이웃해 중 더 나은 해를 찾다 보면 A와 B에서 계속해서 반복될 수 있음
- 이러한 방식의 도입은 순환 현상을 발생시킨다는 문제점을 가지고 있음



메타휴리스틱기법

❖ 타부 탐색(Tabu Search) 알고리즘

- 단계 0

- 초기화

실행 가능한 초기해 X_0 와 타부리스트 길이 T 를 설정하고, 타부리스트 $L_0 = \emptyset$, 알고리즘 변수 $k=0$ 을 초기화

현재까지 찾은 최선해 X^* 는 초기해인 X_0 , 최선해의 목적함수 값인 Z^* 는 초기해의 목적함수 값인 $Z(X_0)$ 로 정의

- 단계 1

- 이웃해 생성 및 해 이동 1.1

타부리스트에 포함되지 않는 X_k 의 실행 가능한 이웃해 생성

1.2 생성된 이웃해 중 목적함수 값이 가장 작은 해 Y 를 선택

1.3 현재해를 이동하여 $X_{(k+1)}=Y$ 로 설정하고, 만약 $Z(Y) < Z^*$ 를 만족하면 $X^*=Y$ 로, $Z^*=Z(Y)$ 로 변경

- 단계 2

- 타부리스트 갱신

2.1 현재해 X_k 를 타부리스트에 포함 ($L_{(k+1)}=L_k + \{X_k\}$)

2.2 지금으로부터 T 단계 이전에 타부리스트에 포함되었던 해를 타부리스트에서 제거

- 단계 3

- 종료조건

3.1 만약 알고리즘 종료 조건에도 달하면, X^* 를 최종해로 선택하고 알고리즘을 종료

3.2 아니면 $k=k+1$ 로 변경한 후 단계 1로

메타휴리스틱기법

❖ 타부 탐색 해법

- 예제

작업 j	가공시간 p_j	납기 d_j
1	13	30
2	10	35
3	8	20
4	10	25

알고리즘 구성요소			
초기해	$X_0 = 1-2-3-4$	타부 리스트 길이	$T=2$
이웃해 생성	연속하는 두 작업의 교환	종료 조건	$k \geq 5$

반복 k	탐색해 X_k $Z(X_{k+1})$	타부리스트 L_k	이웃해	선택해 Y $Z(Y)$	최선해
0	1-2-3-4 27	\emptyset	2-1-3-4 27 1-3-2-4 17 1-2-4-3 29	1-3-2-4 17	17
1	1-3-2-4 17	1-2-3-4	3-1-2-4 16 1-2-3-4 - 1-3-4-2 13	1-3-4-2 13	13
2	1-3-4-2 13	1-2-3-4 1-3-2-4	3-1-4-2 12 1-4-3-2 17 1-3-2-4 -	3-1-4-2 12	12
3	3-1-4-2 12	1-3-2-4 1-3-4-2	1-3-4-2 - 3-4-1-2 7 3-1-2-4 16	3-4-1-2 7	7
4	3-4-1-2 7	1-3-4-2 3-1-4-2	4-3-1-2 7 3-1-4-2 - 3-4-2-1 11	4-3-1-2 7	7
5	4-3-1-2 7	3-1-4-2 3-4-1-2	3-4-1-2 - 4-1-3-2 18 4-3-2-1 11	4-3-2-1 11	종료조건 $k \geq 5$

값을 지워버리는 것

메타휴리스틱기법

❖ 효율적인 메타 휴리스틱의 조건

- 기본원칙1
 - 새로운 해의 생성이 쉬워야 함
- 기본원칙2
 - 모든 해는 최적해로의 경로가 존재해야 함
- 기본원칙3
 - 이웃해는 현재해와 가깝게 위치해야 함
- 기본원칙4
 - 목적함수의 위상이 균일하지 않아야 함
 - 우수한 해가 위치한 영역을 치밀하게 탐색한다는 메타휴리스틱기법의 장점을 살리기 어렵게 만듦