13주차 2차시 해 탐색 알고리즘의 이해 II

【학습목표】

1. 분기 한정 기법 알고리즘을 이해할 수 있다.
2. 유전자 알고리즘을 이해할 수 있다.

학습내용1 : 분기 한정 기법

1. 분기 한정 기법의 개요

• 백트래킹 기법은 깊이 우선 탐색수행
• 최적화 문제에 대해서는 최적해가 상태 공간 트리의 어디에 있는지 알 수 없으므로, 트리에서 대부분의노드를 탐색하여야 함
• 입력의 크기가 커지면 해를 찾는 것은 거의 불가능
• 분기 한정(Branch-and-bound) 기법은 이러한 단점을 보완하는 탐색 기법

• 분기 한정 기법은 상태 공간 트리의 각 노드 (상태)에 특정한 값 (한정값)을 부여
• 노드의 한정값을 활용하여 가지치기를 함으로서 백트래킹기법 보다 빠르게 해를 찾음
• 분기 한정 기법에서는 가장 우수한 한정값을 가진 노드를 먼저 탐색하는 최선 우선 탐색 (Best First Search)으로 해를 찾음

* 분기 한정 기법의 효율적인 탐색 원리
① 최적해를 찾은 후에, 탐색하여야 할 나머지 노드의 한정값이 최적해의 값과 같거나 나쁘면 더 이상 탐색하지 않는다.
② 상태 공간 트리의 대부분의 노드가 문제의 조건에 맞지 않아서 해가 되지 못한다.
③ 최적해가 있을만한 영역을 먼저 탐색한다.

* 알고리즘

Branch-and-Bound(S)

1. 상태 S의 한정값을 계산한다.
2. activeNodes = { S } // 탐색되어야 하는 상태의 집합
3. bestValue = ∞ // 현재까지 탐색된 해 중의 최소값
4. while (activeNodes ≠ Ø) {
5. S_{min}= activeNodes의 상태 중에서 한정값이 가장 작은 상태
6. S_{min}을 activeNodes에서 제거한다.
7. S_{min}의 자식 (확장 가능한) 노드 S'_1, S'_2, ..., S'_k를 생성하고, 각각의 한정값을 계산한다.
8. for i=1 to k { // 확장한 각 자식 S'_i에 대해서
9. if (S'_i의 한정값 ≥ bestValue)
10. S'_i를 가지치기한다. // S'i로부터 탐색해도 더 우수한 해가 없다.
11. else if (S'_i가 완전한 해이고 S'i의 값 < bestValue)
12. bestValue = S'_i의 값
13. bestSolution = S'_i
14. else
15. S'_i를 activeNodes에 추가한다. // 나중에 차례가 되면 S'_i로부터 탐색을 수행

}

}

- 각 상태에서는 한정값을 계산하는 방법은 문제에 따라 다르다.
- 하나의 상태에 대해 탐색을 마친 후에는 acitveNodes에서 가장 작은 한정값을 가진 상태를 탐색한다. 즉, 최선 우선 탐색을 한다.
- 여기서 activeNodes는 탐색할 상태의 집합이다.
- Line 1~3: 문제의 초기 상태의 한정값을 계산한 후, 초기 상태만을 원소로 갖는 activeNodes로서 탐색이 시작
- bestValue는 현재까지 탐색된 해 중의 가장 작은 값을 가지는데, bestValue를 가장 큰 수로 초기화

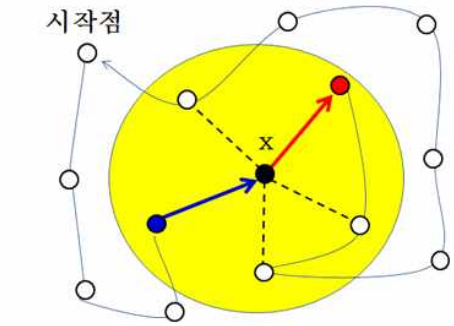
- Line 4~15의 while-루프: activeNodes가 공집합이 되면, 즉, 더 이상 탐색할 상태가 없으므로 탐색을 중단
- activeNodes가 공집합이 아니면, line 5에서는 activeNodes에서 한정값이 가장 작은 상태를 선택하여 이를 S_{min}이라고 하자.
- Line 6: S_{min}을 activeNodes에서 제거
- Line 7: S_{min}으로부터 확장 가능한 상태 (자식 노드)를 생성하고 각각의 상태에 대한 한정값을 계산
- Line 8~15의 for-루프: line 7에서 S_{min}으로부터 생성된 각각의 S'_i에 대하여 루프가 수행

- Line 9~10: S'_i의 한정값이 bestValue보다 같거나 크면 가지치기: S'_i로부터 탐색 되지 않도록 한다.
- Line 11~13: 만일 S'_i가 완전한 해이고 동시에 S'_i의 값이 bestValue보다 작으면, 즉, 더 '우수한' 해이면, bestValue를 S'_i의 값으로 갱신하고, S'_i가 bestSolution이 됨
- Line 15: line 9와 11의 if-조건이 모두 '거짓'이면 S'_i를 나중에 탐색하기위해서activeNodes에 추가함

2. TSP를 분기 한정 기법으로 해결하는 과정

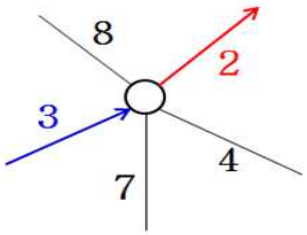
- 한정값 계산을 위해서 여행자 문제의 조건
- 해는 주어진 시작점에서 출발하여 모든 다른 점을 1번씩만 방문하고 시작점으로 돌아와야 한다.
- 이러한 경로 상의 1개의 점 x를 살펴보면, 다른 점에서 점 x로 들어온 후에 점 x를 떠나 또 다른 점으로 나간다.

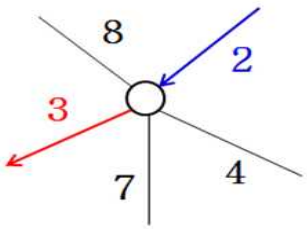
한정값의 계산 방법:
- 점 x로 들어올 때와 나갈 때 사용되는 선분의 가중치를 한정값 계산에활용
- 점 x에 연결된 선분 중에서 가중치가 가장 작은 두 선분의 가중치의 합의 1/2을 한정값으로 이용



- 이 방법은 최적의 여행자 경로를 '근시안적'으로 임의 점 하나에 대해서만 고려한 것임
- 가중치의 합을 1/2로 곱하는 이유: 한 점에서 나가는 선분은 인접한 (다른) 점으로부터 들어오는 선분과 동일하기 때문
- 단, 소수점 이하의 숫자는 올림을 한다.

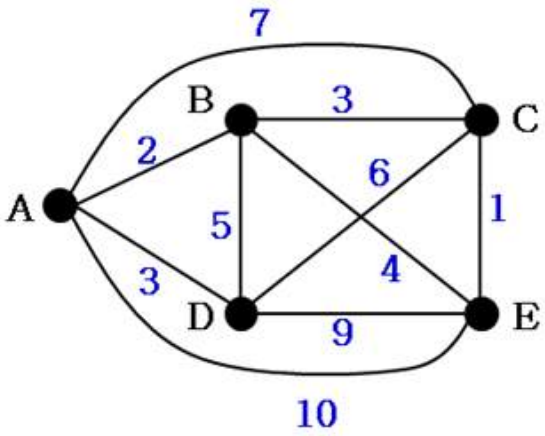
- 아래의 그림은 점에 인접한 선분의 가중치 중에서 2개의 가장 작은 가중치는 3과 2이다.
- 가중치 3인 선분으로 들어와서 가중치 2인 선분으로 나가든지 (왼쪽 그림) 반대로 가중치 2인 선분으로 들어와서 가중치 3인 선분으로 나가든지 (오른쪽 그림), 두 경우 모두 최소의 비용으로 이 점을 방문하는 것이다





Branch-and-Bound 알고리즘 수행과정

- A=시작점
- 초기 상태= [A]
- Branch-and-Bound([A])를 호출하여 탐색 시작



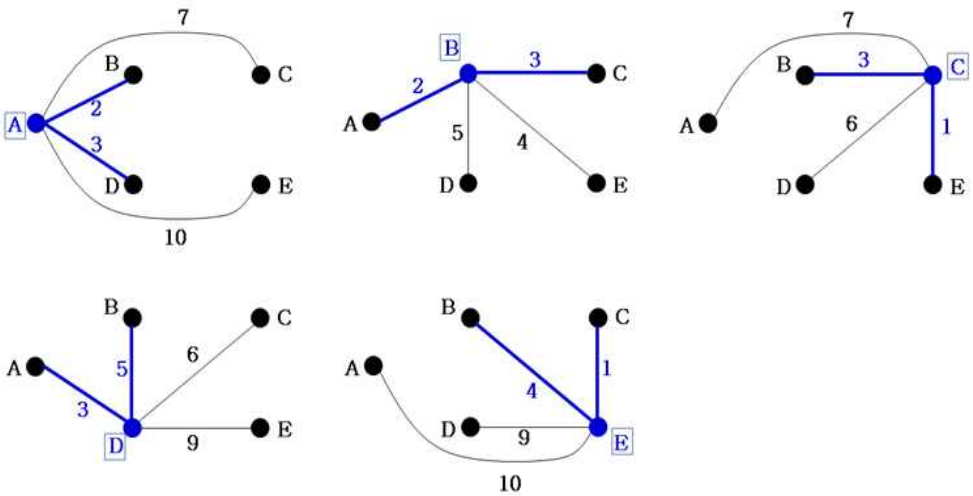
- Line 1: 초기 상태 [A]의 한정값 계산
- 초기 상태는 경로를 시작하기 전이므로, 각 점에 인접한 선분의 가중치 중에서 가장 작은 2개의 가중치의 합을 구한 다음에, 모든 점의 합의 1/2을 한정값으로 정한다.

점 A: 2, 3
점 B: 2, 3
점 C: 1, 3
점 D: 3, 5
점 E: 1, 4

- 따라서 초기 상태의 한정값은 다음과 같이 계산 된다.

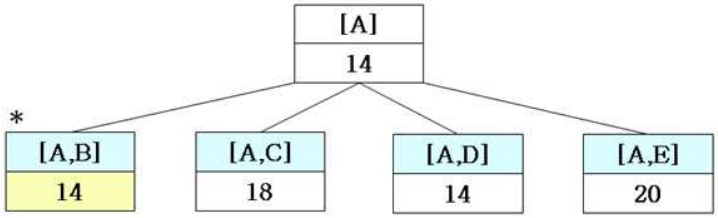
A B C D E

- [(2+3) + (2+3) + (1+3) + (3+5) + (1+4)] x 1/2 = 27/2 = 14

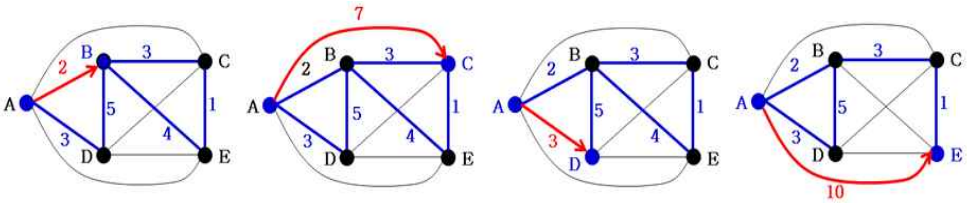


- Line 2~3: activeNodes= {S }, bestValue=∞로 각각 초기화
- Line 4의 while-루프가 activeNodes 집합이 공집합이 될 때까지 수행
- Line 5: activeNodes 집합에 초기 상태 [A]만 있으므로, S_{min}=[A]
- Line 6: [A]가 activeNodes 집합으로부터 제거되어 일시적으로 activeNodes 집합은 공집합.
- Line 7: S_{min} (즉, 상태 [A])의 자식 상태 노드를 아래와 같이 생성하고, 각각 한정값을 구한다.

☞ 여기서 자식 노드는 두 번째 방문 하는 점이 B인 상태 [A,B], C인 상태 [A,C], D인 상태 [A,D], E인 상태 [A,E]이다.

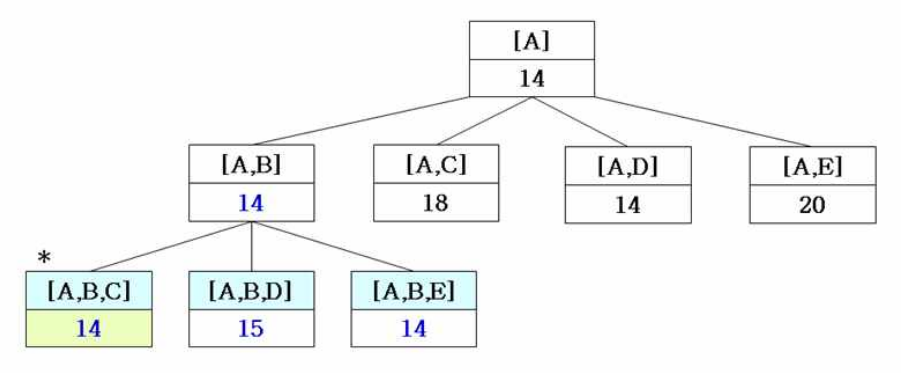


상태 [A,B], [A,C], [A,D], [A,E]의 한정값은 다음과 같이 각각 계산된다.

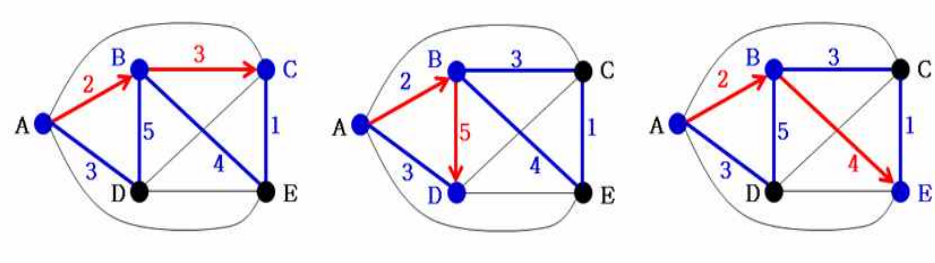


- Line 8의 for-루프: 위와 같이 생성된 4개 (k=4)의 상태 각각에 대하여, (즉, S'_1=[A,B], S'_2=[A,C], S'_3=[A,D], S'_4=[A,E]) line 9~15를 수행한다.
- Line 9: i=1: S'_1의 한정값인 14와 현재의 bestValue인 ∞를 비교하여서 if-조건이 '거짓'이고, line 11에서 상태 [A,B]가 완전한 해가 아니므로, line 14~15에서 S'_1을 activeNodes에 추가
- 이와 유사하게 i=2, 3, 4일 때에도 각각 S'_2, S'_3, S'_4가 activeNodes에 추가
- activeNodes = {[A,B], [A,C], [A,D], [A,E]}

• 다음으로 line 4 while-루프의 조건 검사에서 activeNodes가 공집합이 아니므로, line 5에서 한정값이 가장 작은 상태를 찾는다. 상태 [A,B]와 [A,E]가 동일한 최소의 한정값을 가지므로 이중에서 임의로 S_{min} = [A,B]라고 하자.
• Line 6: activeNodes로부터 [A,B]를 제거하여, activeNodes = { [A,C], [A,D], [A,E] }가 된다.
• Line 7: [A,B]의 자식 상태를 아래와 같이 생성하고, 각각의 한정값을 계산한다.
☞ 여기서 자식 노드는 세번째방문하는점이C인 상태 [A,B,C], D인 상태 [A,B,D], E인 상태 [A,B,E]이다.



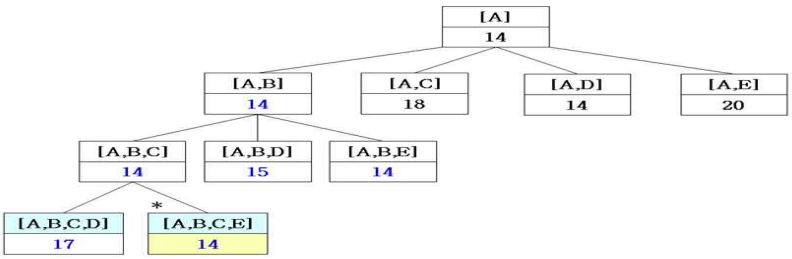
• 상태 [A,B,C], [A,B,D], [A,B,E]의 한정값은 다음과 같이 각각 계산 된다.



• [A,B,C]의 한정값: ([2+3]+[2+3]+[1+3]+[3+5]+[1+4])/2 = 27/2 = 14
• [A,B,D]의 한정값: ([2+3]+[2+5]+[1+3]+[3+5]+[1+4])/2 = 29/2 = 15
• [A,B,E]의 한정값: ([2+3]+[2+4]+[1+3]+[3+5]+[1+4])/2 = 28/2 = 14
• Line 8의 for-루프: 위와 같이 생성된 3개 (k=3)의 상태 각각에 대하여, (즉, S'_1=[A,B,C], S'_2=[A,B,D], S'_3=[A,B,E]) line 9~15를 수행한다.

• Line 9: i=1: S'_1의 한정값인 14와 현재의 bestValue인 ∞를 비교하여서 if-조건이 '거짓'이고,
• Line 11: 상태 [A,B,C]가 완전한 해가 아니므로
• Line 14~15에서 S'_1을 activeNodes에 추가한다.
• 이와 유사하게 i=2, 3일 때에도 각각 S'_2, S'_3이 activeNodes에 추가된다. 따라서activeNodes = {[A,C], [A,D], [A,E], [A,B,C], [A,B,D], [A,B,E] }이다.
• 다음엔 line 4 while-루프의 조건 검사에서 activeNodes가 공집합이 아니므로, line 5에서 한정값이 가장 작은 상태를 찾는다.
☞ 상태 [A,B,C], [A,B,E], [A,D]가 동일한 최소의 한정값을 가지므로 이중에서 임의로 S_{min} = [A,B,C]라고 하자.

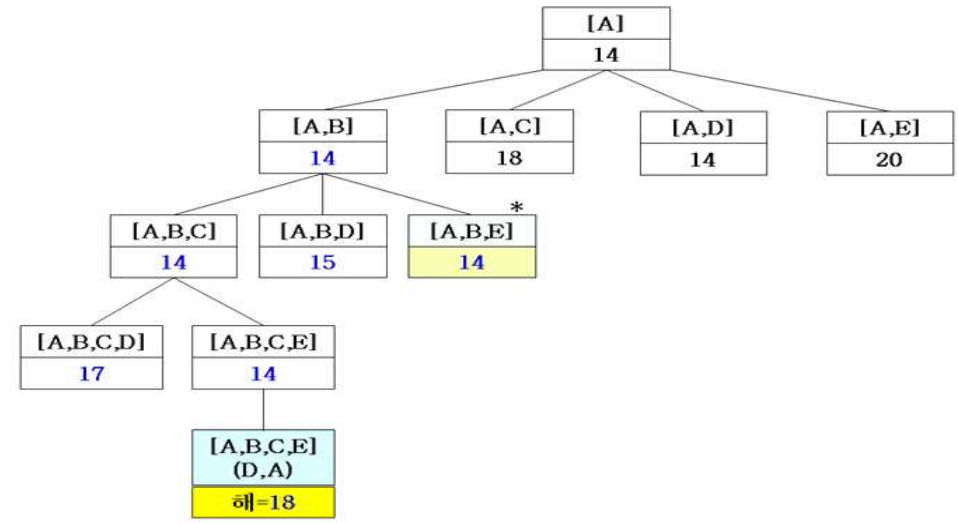
• Line 6: activeNodes로부터 [A,B,C]를 제거하여, activeNodes = {[A,C], [A,D], [A,E], [A,B,D], [A,B,E] }가 된다.
• Line 7: [A,B,C]의 자식 상태를 아래와 같이 생성하고, 각각의 한정값을 구한다.
☞ 여기서 자식 노드들은 네 번째 방문하는 점이D인 상태 [A,B,C,D]와 E인 상태 [A,B,C,E]이다.



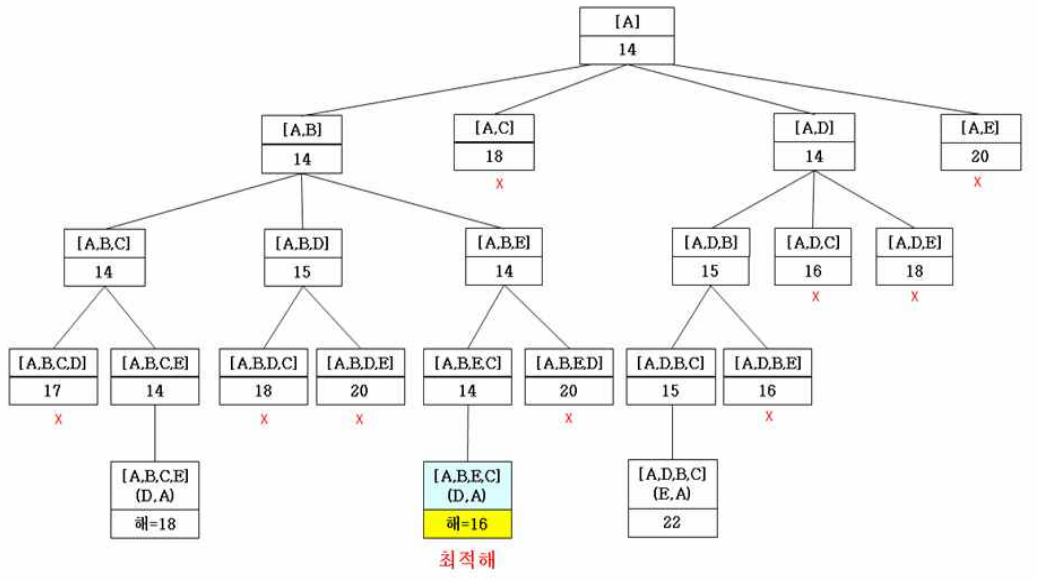
• 상태 [A,B,C,D], [A,B,C,E]의 한정값은 다음과같이각각계산된다.
• [A,B,C,D]의 한정값: ([2+3]+[2+3]+[6+3]+[3+6]+[1+4])/2 = 33/2 = 17
• [A,B,C,E]의 한정값: ([2+3]+[2+3]+[1+3]+[3+5]+[1+4])/2 = 27/2 = 14
• Line 8의 for-루프에서는 위와 같이 생성된 2개 (k=2)의 상태 각각에 대하여, (즉, S'_1=[A,B,C,D], S'_2=[A,B,C,E]) line 9~15를 수행한다.

• Line 9에서 i=1: C_1의 한정값인 17과 현재의 bestValue인 ∞를 비교하여서 if-조건이 '거짓'이고
• Line 11에서 상태 [A,B,C,D]가 완전한 해가 아니므로
• Line 14~15에서 S'_1을 activeNodes에 추가시킨다.
• 이와 유사하게 i=2일 때에도 C_2가 activeNodes에 추가된다. 따라서 activeNodes = {[A,C], [A,D], [A,E], [A,B,D], [A,B,E], [A,B,C,D], [A,B,C,E] }이다.
• 다음엔 line 4 while-루프의 조건 검사에서 activeNodes가 공집합이 아니므로, line 5에서 한정값이 가장작은상태를찾는다.
• 상태 [A,B,C,E], [A,B,E], [A,D]가 동일한 최소 한정값을 가지므로이중에서임의로S_{min} = [A,B,C,E]라고 하자.

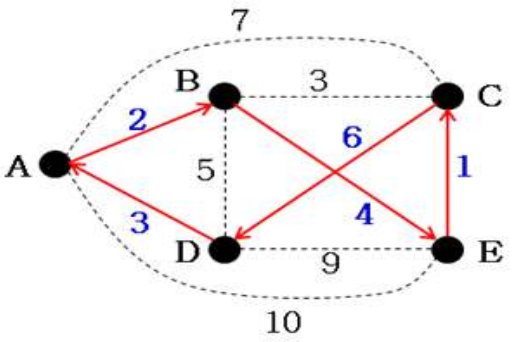
• Line 6: activeNodes로부터 [A,B,C,E]를 제거하여, activeNodes = {[A,C], [A,D], [A,E], [A,B,D], [A,B,E], [A,B,C,D]}가 된다.
• Line 7: [A,B,C,E]의 자식 상태가 1개이므로, 즉, E 다음에 방문할 점인 점 D 하나만 남아 있으므로 D를 방문하는 상태 [A,B,C,E,D]이다. 그런데 D에서 시작점 A로 돌아가야 하므로 하나의 해가 완성된 셈이다. 이 해의 경로 A-B-C-E-D-A의 거리는 2+3+1+9+3 = 18이다.
• Line 11: 해가 발견되었고, 경로 거리가 bestValue =∞보다 작으므로 if-조건이 '참'이 되어서, bestValue=18, bestSolution=[A,B,C,E,D,A]가 된다.



- 다음엔 상태 [A,B,E]로부터 탐색이 시작되며, 그 최종 결과는 다음과 같다.



• 이 예제에서 상태 [A,B,E,C,D,A]가 최적해이고, 경로의 길이는 16이다. 다음 그림은 최적해에 대한 경로를 보이고 있다.



• 백트래킹 알고리즘이 방문한 상태 공간 트리의 노드 수는 총51개
• 분기 한정 알고리즘은 22개
• 이처럼 최적화 문제의 해를 탐색하는 데는 분기 한정 기법이 백트래킹 기법보다 훨씬 우수한 성능을 보임
• 분기 한정 알고리즘은 한정값을 사용하여 최적해가 없다고 판단되는 부분은 탐색을 하지 않고 최선 우선 탐색을 하기 때문이다.

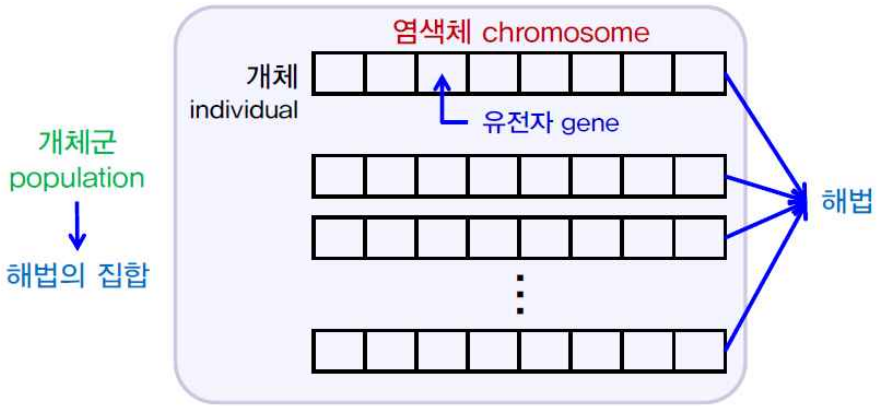
학습내용2 : 유전자 알고리즘

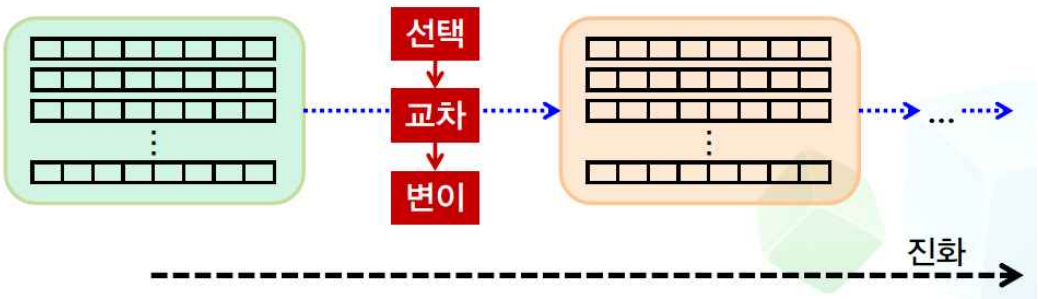
1. 유전알고리즘이란?

유전알고리즘은 1970년 초 John Holland
⦿ 자연계의 진화를 통해 관찰된 메커니즘을 모방하여 최적화 문제를 해결하기 위한 탐색 방법

■ 찰스 다윈의 진화에 대한 원칙을 수용/설계(적자생존의 개념)
☞ 실제 자연에서는 제한된 자원에 대한 개체 간의 경쟁을 통해 환경에 가장 적합한 개체가 형성되고, 이들이 결국은 약자를 다스리게 됨

■ 해가 진화한다.
☞ 주어진 문제 해결을 위해서 연속적인 세대generation를 거치면서 개체individual의 적자생존을 통해 점진적으로 성능을 향상시켜 나감.





2. 유전 알고리즘의 처리 과정

[초기화] 난수를 사용해 n개의 염색체로 이루어진 개체군을 생성
[적합도] 개체군의 각 염색체에 대해 적합도 점수를 계산
[새 개체군] 새로운 개체군이 완성될 때까지 다음 과정을 반복
[선택] 적합도에 따라 개체군에서 두 부모 염색체를 선택
[교차] 교차 확률에 따라 두 부모를 교차시켜 자손을 생성
[변이] 변이 확률에 따라 새 자손의 염색체의 선택된 위치의 값을 변경
[저장] 새 자손을 새로운 개체군에 포함시킴
[대체] 새로운 개체군으로 이전의 개체군을 대체
[종료검사] 종료 조건이 만족되면 종료하고, 현 개체군의 가장 좋은 해를 반환
[반복] 위의 [적합도] 계산 부분부터 다시 수행

3. 처리 과정에 영향을 미치는 요인들

⦿ 염색체를 어떻게 만들 것인가?
▪교차 및 변이 연산에도 영향을 미침

⦿ 부모 염색체를 어떻게 선택할 것인가?
▪ 좋은 부모가 좋은 자손을 생산할 것이라는 희망에서 출발
▪ 엘리티즘 elitism
☞ 적어도 하나의 가장 좋은 해가 아무런 변화도 없이 새로운 개체군으로 복사

⦿ 적합도를 어떻게 계산할 것인가?

⦿ 선택 시 비율을 어떻게 정할 것인가?

⦿ 교차 위치와 교차 확률, 변이 확률?

4. 유전알고리즘의 연산자

⦿ 표현하려는 해에 대한 정보를 염색체의 형태로 표현
- 주어진 문제에 전적으로 의존
- 이진 인코딩, 순열 인코딩, 값 인코딩, 트리 인코딩 등

⦿ 이진 인코딩
- 가장 보편적인 방법
- 각 염색체를 0과 1의 비트열로 표현

☞ 각 비트는 해의 한 특성을 표현

염색체1	1101100100110110
염색체2	1101111000011110

- 종종 교차/변이 연산 후에 별도의 작업이 필요한 경우 발생

⦿ 순열 인코딩
- 외판원 문제나 작업 순서를 결정하는 것과 같이 순서를 결정하는 문제에 적용

염색체1	1 5 3 2 6 4 7 9 8
염색체2	8 5 6 7 2 3 1 4 9

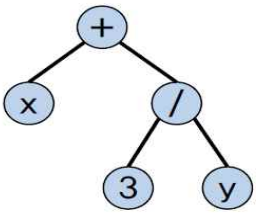
⦿ 값 인코딩
- 직접 값을 사용해서 인코딩하는 방법

염색체1	1.2345 5.3412 0.4987 2.1495 2.5757
염색체2	ABDJEIFJJHDIERJAFDLEFGJB
염색체3	(back), (back), (right), (forward), (left)

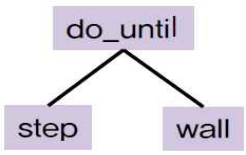
- 각 염색체가 어떤 값(숫자, 실수, 문자 등)들의 스트링이 된다.

⦿ 트리 인코딩

▪ 각 염색체가 프로그래밍 언어에서 함수나 명령과 같은 어떤 객체의 트리로 표현됨



(+ x (/ 3 y))



(do_until step wall)

▪ LISP에서 종종 사용

☞ 트리 형태로 쉽게 파싱 가능 → 상대적으로 용이한 교차/변이 연산

5. 선택 (selection) 연산

⦿ 개체군으로부터 부모가 되어 교차를 수행할 염색체를 선택

▪ 가장 좋은 부모가 생존해서 자손을 생성해야만 한다.

▪ 룰렛 힐 선택, 순위 선택, 토너먼트 선택, 엘리티즘 등

⦿ 룰렛 힐 roulette wheel 선택

▪ 염색체들의 적합도에 비례해서 개체군에서 다음 세대로 넘겨줄 염색체를 선택하기 위한 방법

• 가장 적합한 염색체가 다음 세대로 전달되는 것을 보장하는 것이 아니라, 그것들이 그렇게 될 가능성을 높여주는 방법

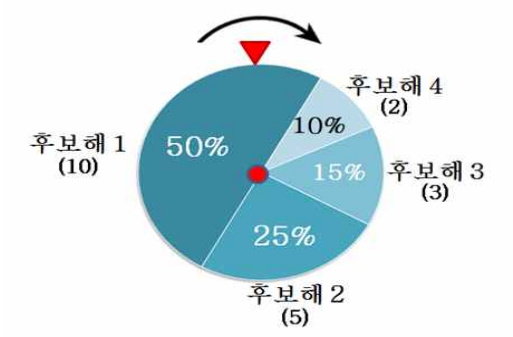
• 선택 연산을 가장 간단히 구현하는 방법은 룰렛 휠 (roulette wheel) 방법이다.

→ 각 후보해의 적합도에 비례하여 원반의 면적을 할당하고, 원반을 회전시켜서

→ 원반이 멈추었을 때 핀이 가리키는 후보해를 선택한다.

→ 면적이 넓은 후보해가 선택될 확률이 높다.

- 후보해 1의 적합도: 10
- 후보해 2의 적합도: 5
- 후보해 3의 적합도: 3
- 후보해 4의 적합도: 2



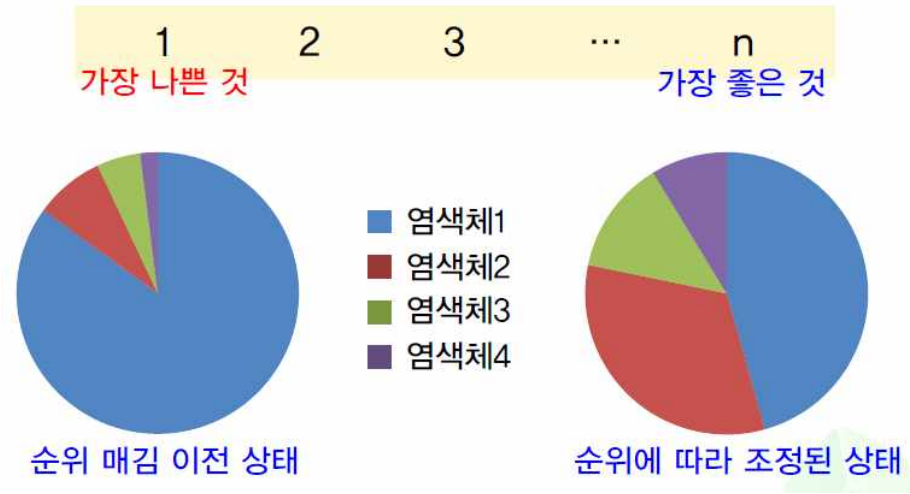
- 각 후보해의 원반 면적은 (후보해의 적합도/ 모든 후보해의 적합도의합)에 비례한다.
- 앞의 예제에서 모든 적합도의 합이 10 + 5 + 3 + 2 = 20이므로,

→ 후보해 1의 면적은 10/20 = 50%
→ 후보해 2의 면적은 5/20 = 25%
→ 후보해 3의 면적은 3/20 = 15%
→ 후보해 4의 면적은 2/20 = 10%

- 현재 4개의 후보해가 있으므로, 4번 원반을 돌리고 회전이 멈추었을 때 핀이 가리키는 후보해를 각각 선택한다.

⦿ 순위 rank 선택

▪ 개체군에 대해 순위를 지정한 다음 각 염색체에게는 이러한 순위에 따라 적합도 점수를 할당.



▪ 모든 염색체가 선택될 기회 제공, 늦은 수렴 속도

⦿ 토너먼트 선택
- 두 개의 염색체를 임의로 선택한 후, 0과 1 사이의 난수를 발생시킨다.
- 이 값이 기준값 보다 작으면 두 염색체 중에서 적합도가 좋은 것을 선택, 그렇지 않으며 적합도가 낮은 것을 선택

⦿ 엘리티즘
- 몇 개의 가장 좋은 염색체를 새로운 개체군으로 그대로 복사한 후, 나머지는 전형적인 방법과 동일하게 동작

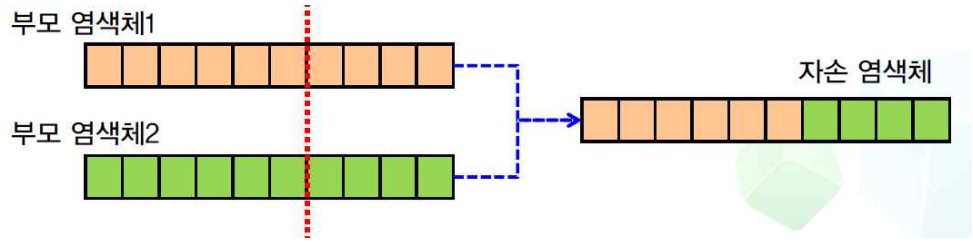
☞ 연산 과정에서 가장 좋은 염색체의 분실 가능성을 방지

☞ 급격한 성능 향상이 가능

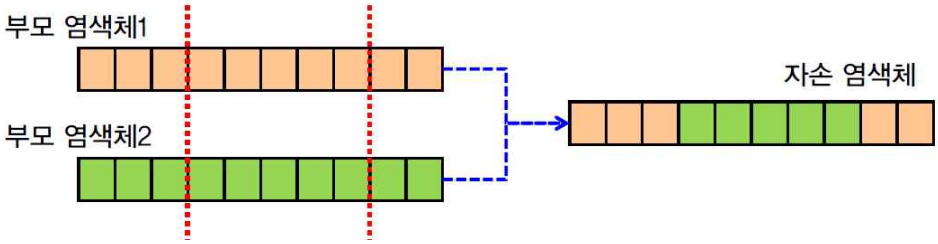
6. 교차 (crossover) 연산

⦿ 두 해의 특징을 부분 결합하여 하나의 새로운 특징을 만들어 내는 GA의 대표적인 연산
- 다른 최적화 방법과 구별짓는 주요 요소
- 연산자의 유형과 구현은 인코딩과 주어진 문제에 의존적
- 단일점 교차, 두점 교차, 균등 교차, 산술 교차 등

⦿ 단일점 single point 교차



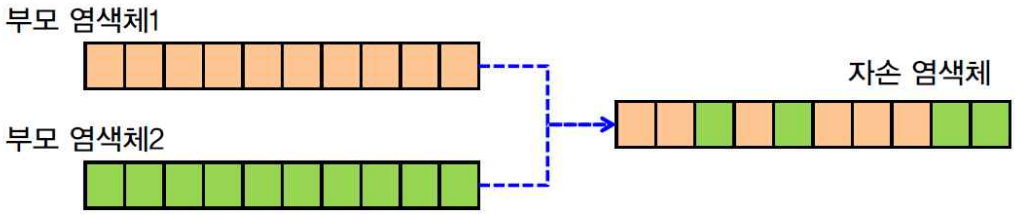
⦿ 두점 two point 교차



⦿ 균등 uniform 교차

```
if 난수(유전자(i)) > Θ then 자손염색체의 유전자(i) ←부모염색체1의 유전자(i)
 else 자손염색체의 유전자(i) ←부모염색체2의 유전자(i)
```

Θ=0.65 난수(i) = {0.75 0.86 0.43 0.67 0.21 0.95 0.70 0.84 0.32 0.59 }

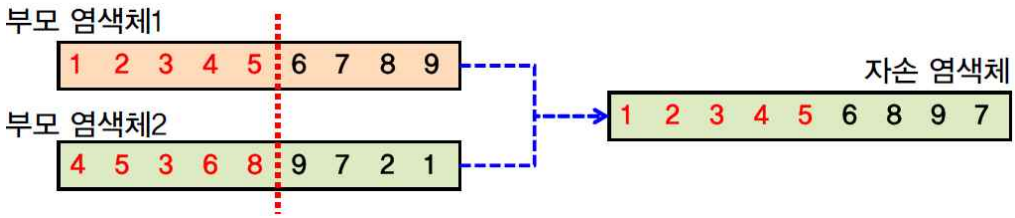


⦿ 산술 arithmetic 교차

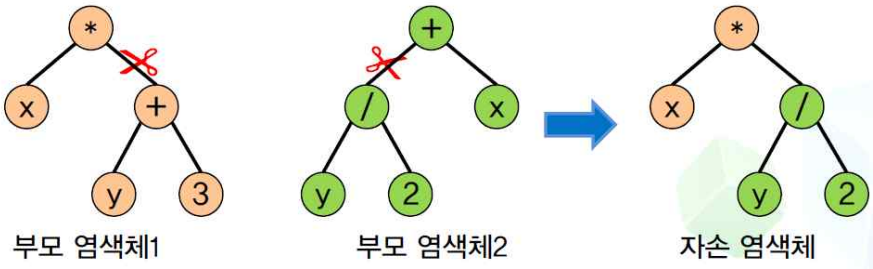
▪ 두 부모 염색체에 특정 산술 연산을 적용하여 자손 염색체를 생성하는 방법

☞ 실수 값을 갖는 염색체를 사용하는 경우에 유용

⦿ 순열 인코딩에서의 단일점 교차 방법



⦿ 트리 인코딩에서의 단일점 교차 방법



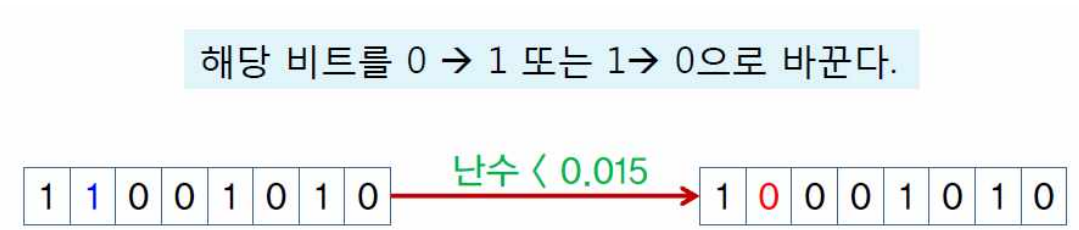
7. 변이 (mutation) 연산

⦿ 낮은 확률을 가지고 새로운 개체의 일부분의 유전자를 변경하는 연산
- 자손 염색체가 부모 염색체에 없는 속성을 갖도록 유도
- 탐색 공간에서 임의적인 이동을 유발하여 개체군 내의 다양성을 유지하며 정상보다 이른 수렴이 일어나지 않도록 억제하는 역할

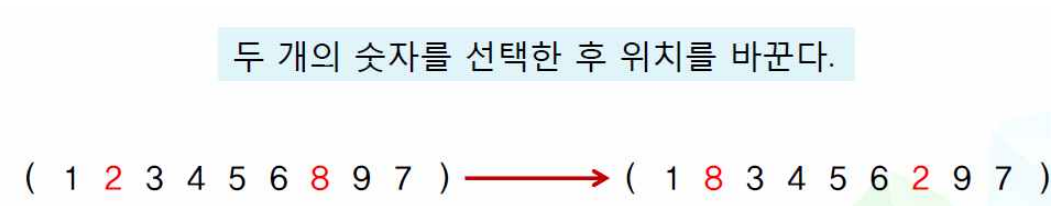
⦿ 전형적인 연산 방법

If 각 유전자에 대한 0~1사이의 난수 < Θ then 해당 유전자를 임의로 변경
else 아무런 변화도 없음

⦿ 이진 인코딩의 경우



⦿ 순열 인코딩의 경우



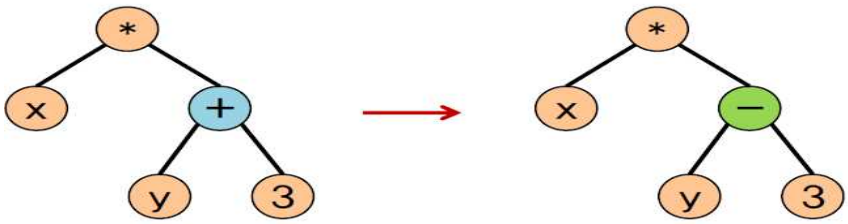
⊙ 실수값 인코딩의 경우

선택된 유전자의 값에 작은 값을 더하거나 빼준다.

(1.29 5.68 2.86 4.11 5.55) ⟶ (1.29 5.68 2.73 4.22 5.55)

⊙ 트리 인코딩의 경우

연산자 또는 숫자에 해당하는 노드를 변경



【학습정리】

1. 분기 한정 기법
- 백트래킹 기법은 깊이 우선 탐색수행
- 최적화 문제에 대해서는 최적해가 상태 공간 트리의 어디에 있는지 알 수 없으므로, 트리에서 대부분의노드를 탐색하여야 함
- 입력의 크기가 커지면 해를 찾는 것은 거의 불가능
- 분기 한정(Branch-and-bound) 기법은 이러한 단점을 보완하는 탐색 기법
- 분기 한정 기법은 상태 공간 트리의 각 노드 (상태)에 특정한 값 (한정값)을 부여
- 노드의 한정값을 활용하여 가지치기를 함으로서 백트래킹기법 보다 빠르게 해를 찾음
- 분기 한정 기법에서는 가장 우수한 한정값을 가진 노드를 먼저 탐색하는 최선 우선 탐색 (Best First Search)으로 해를 찾음

- 분기 한정 기법의 효율적인 탐색 원리
① 최적해를 찾은 후에, 탐색하여야 할 나머지 노드의 한정값이 최적해의 값과 같거나 나쁘면 더 이상 탐색하지 않는다.
② 상태 공간 트리의 대부분의 노드가 문제의 조건에 맞지 않아서 해가 되지 못한다.
③ 최적해가 있을만한 영역을 먼저 탐색한다.

2. 유전자 알고리즘

- 유전자 알고리즘(genetic algorithm)
- 자연계의 진화를 통해 관찰된 메커니즘을 모방하여 최적활 문제를 해결하기 위한 탐색 방법
- 주요 연산: 선택, 교차, 변이

- 기본 처리 과정
① 난수를 사용해 n개 염색체로 구성된 개체군을 형성
② 각 염색체의 적합도 계산
③ while (새로운 개체군이 형성될 동안)
④ 선택 → 교차 → 변이 연산을 수행하여 자손 염색체 생성
⑤ 종료조건 검사 또는 단계2부터 반복

- 처리 시 주요 고려 사항 → 염색체의 인코딩 방법, 선택 방법, 적합도 계산 방법, 교차 위치 및 교차 확률, 변이 확률, 개체군의 크기

- 주요 연산
염색체의 인코딩(표현하려는 해에 대한 정보를 염색체의 형태로 표현하는 방법
→ 인코딩 방법의 선정은 교차 및 변이 연산의 유형에 영향을 미침)
→ 이진 인코딩 : 각 염색체를 0/1의 나열로 표현
→ 순열 인코딩 : 각 염색체의 값들이 순서를 나타내는 값을 가짐
→ 값 인코딩 : 직접 값(숫자, 문자 등)을 사용
→ 트리 인코딩 : 각 염색체가 객체의 트리 형태로 표현

- 선택(개체군으로부터 부모가 되어 교차를 수행할 염색체를 선택)
→ 룰렛 힐 선택 : 염색체의 적합도에 비례하여 선택
→ 순위 선택 : 각 개체에 대해 순위를 지정한 다음 이런 순위를 기준으로 적합도 점수를 할당
→ 토너먼트 선택 : 0~1 사이의 난수와 기준값의 비교를 통해서 부모 염색체를 선정
→ 엘리티즘 : 몇 개의 가장 좋은 염색체에 대해서는 이후의 연산 없이 새로운 개체군으로 그대로 전달하는 방법

- 교차(선택된 두 염색체의 특징을 부분 결합하여 하나의 새로운 특징을 가진 자손 염색체를 생성)
→ 단일점 교차 : 하나의 교차 위치를 기준으로 교차
→ 두점 교차 : 두 개의 교차 위치를 기준으로 교차
→ 균등 교차 : if 각 유전자에 대한 난수 > 기준값 then 첫 번째 부모 염색체의 해당 유전자를 가짐 else 두 번째 부모 염색체의 해당 유전자를 가짐
→ 산술 교차 : AND, 평균과 같은 특정 산술 연산을 적용

- 변이(새로 생성된 자손 염색체의 일부분의 유전자를 임의적으로 변경 → 부모 염색체에는 없는 속성을 부여)