

강의 개요

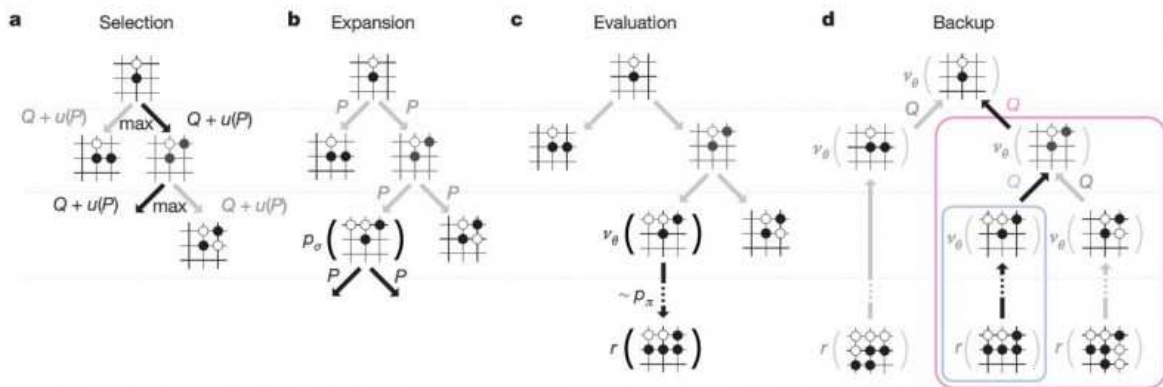
본 강의에서는 인공지능에서 탐색이 의미하는 바를 살펴보고, 8-퍼즐 문제를 해결하는 데 활용할 수 있는 탐색 방법에 대해 학습한다. 특히 최상 우선 탐색 방법을 활용하여 문제 해결을 위한 최적의 경로를 찾는 것을 중점적으로 학습한다.

학습목표

- 인공지능에서 탐색이 의미하는 바를 이해할 수 있다.
- 8-퍼즐 문제를 해결하는 데 활용할 수 있는 탐색 방법에 대해 설명할 수 있다.
- 최상 우선 탐색 방법을 활용하여 문제를 해결하는 최적의 경로를 찾는 과정을 실행할 수 있다.

1. 인공지능에서 탐색의 의미

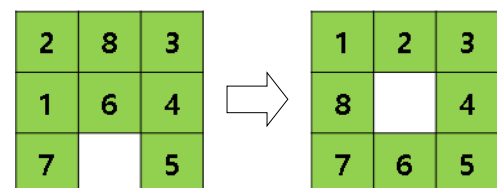
- 탐색은 인공지능에서 기본적으로 중요한 도구이다. 탐색은 컴퓨터가 문제를 해결하기 위하여 스스로 해답에 이르는 경로를 찾아가는 과정으로 탐색만 사용해도 많은 문제들을 해결할 수 있다. 알파고가 그 대표적인 사례이다. 알파고가 바둑을 제패하는 데 핵심적인 역할을 수행한 것이 바로 탐색 알고리즘이다. 바둑에서 상대방이 첫 번째 수를 두면 컴퓨터는 가능한 수에 대하여 탐색한 후에 가장 최적의 수를 선택한다. 문제는 가능한 수가 너무 많아서 제한된 시간 내에 완료할 수 없다는 것이다. 알파고는 딥러닝을 사용하여 가장 가능성 있는 몇 개의 수만을 추려낸 후에 이들 수에 대해서만 탐색한다.



<몬테카를로 트리 탐색 알고리즘>

2. 8-퍼즐 문제의 탐색 방법

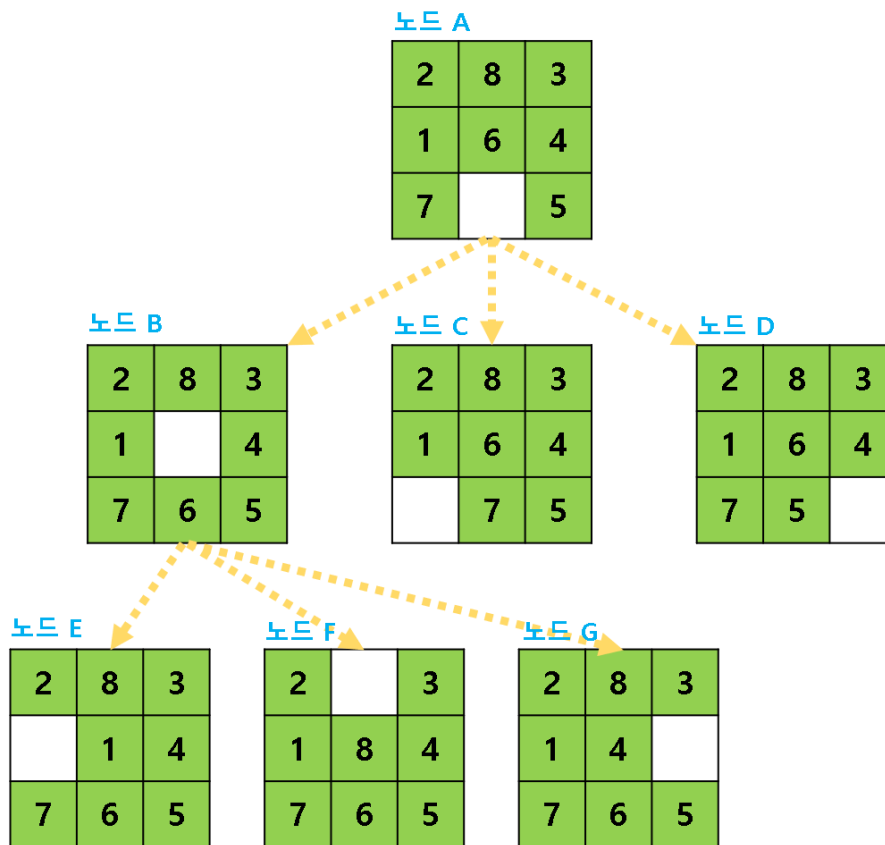
- 8-퍼즐은 8개의 숫자가 적혀진 타일들을 적절히 움직여서 목표 상태에 도달하는 게임이다.
- 현재 상태에서 제자리에 있지 않은 타일의 개수: 4개
- 모든 타일의 목표 위치까지의 거리 총합
: $1 + 1 + 0 + 0 + 0 + 1 + 0 + 2 = 5$



<현재 상태>

<목표 상태>

- 현재 상태에서 목표 상태로 가기 위한 탐색 트리 구조를 만들면 다음과 같다.



- 위의 트리 구조를 바탕으로 깊이 우선 탐색과 너비 우선 탐색을 적용하면 다음과 같다.

| 탐색 방법 | 탐색 경로 |
|----------|---------------------------|
| 깊이 우선 탐색 | A - B - E - F - G - C - D |
| 너비 우선 탐색 | A - B - C - D - E - F - G |

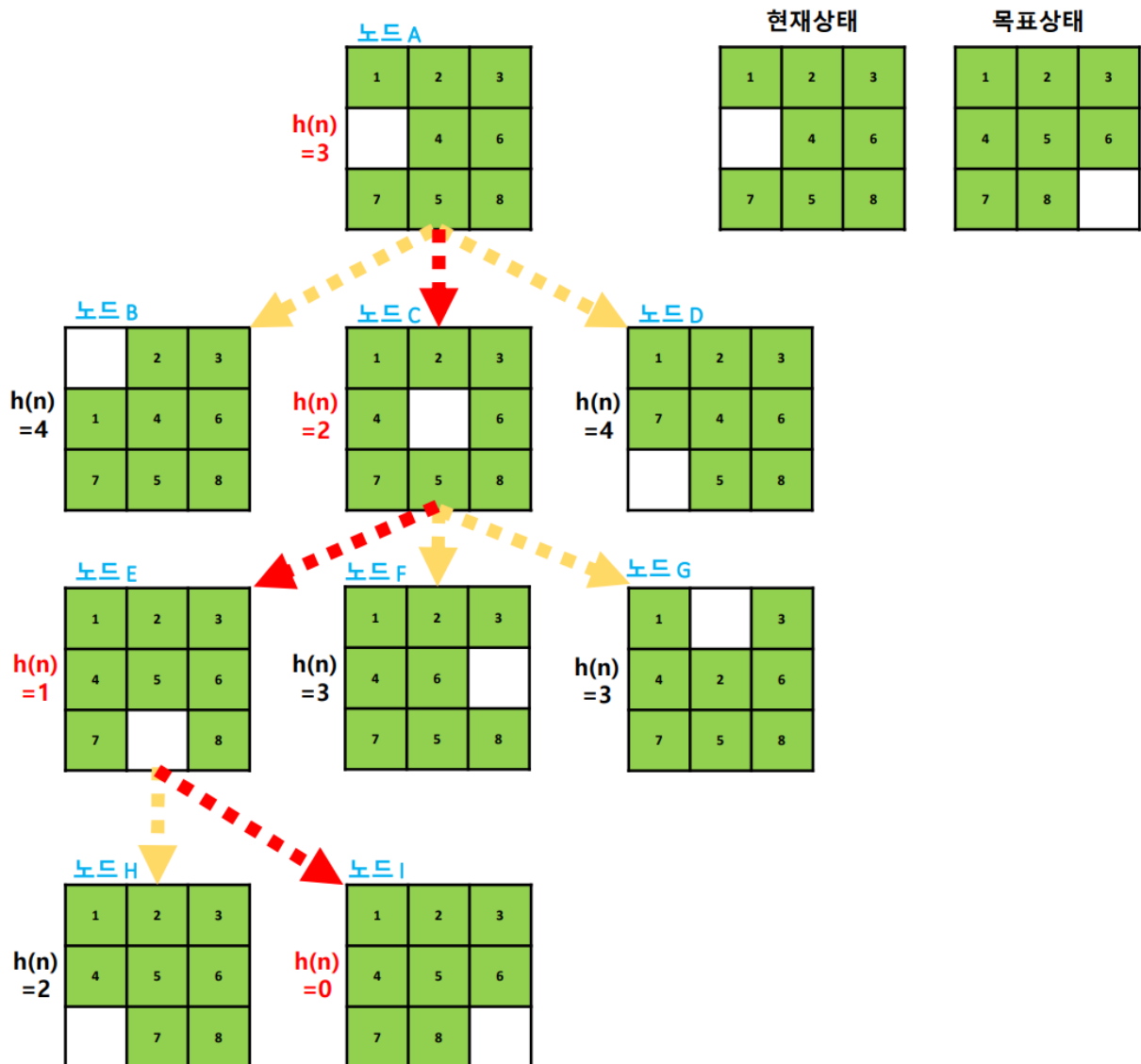
- 깊이 우선 탐색이나 너비 우선 탐색과 같은 탐색 방법을 맹목적 탐색이라고 한다. 그 이유는 모든 노드를 이미 정해져 있는 순서대로 탐색하기 때문이다. 이러한 맹목적 탐색 방법은 시작 노드에서 목표 노드까지의 경로를 찾기 위해 상당히 많은 시간을 소모해야 한다. 그런데 만약, 우리가 문제 영역에 대한 정보나 지식을 사용할 수 있다면 탐색 작업을 훨씬 빠르게 할 수 있지 않을까? 이것을 정보이용 탐색 방법이라고 하며, 경험적 탐색 방법 또는 휴리스틱 탐색 방법이라고도 한다. 이때 사용하는 정보를 휴리스틱 정보(평가함수)라고 한다.
- 휴리스틱(heuristic)이란, 시간이나 정보가 불충분하여 합리적인 판단을 할 수 없거나, 굳이 체계적이고 합리적인 판단을 할 필요가 없는 상황에서 신속하게 어림짐작하는 것을 뜻한다. 다르게 말해서 휴리스틱은 최적의 해를 구하는 것이 아니라, 적절한 해를 빠르게 찾는 방법인 것이다. 우리 인간들은 때때로 상대적으로 중요하지 않은 것을 판단할 때, 논리적 분석이나 정확한 사실에 의거하여 판단하기보다는 경험이나 직관에 의존하는 경우가 있는데 이것이 바로 휴리스틱이다. 이러한 휴리스틱은 인공지능을 구현하는 대부분의 알고리즘에 적용되어있는 만큼 인공지능 분야에서 매우 중요하게 작용한다.

- 그렇다면, 8-퍼즐 문제의 경우에 휴리스틱을 적용하여 문제를 해결하려면 어떻게 해야 할까?
현재 노드에서 다음 노드로 나아갈 때, 휴리스틱 정보를 활용하는 방법이 있다. 이때 휴리스틱 정보를 평가함수라고도 하는데 다음과 같은 것들이 있을 수 있다.

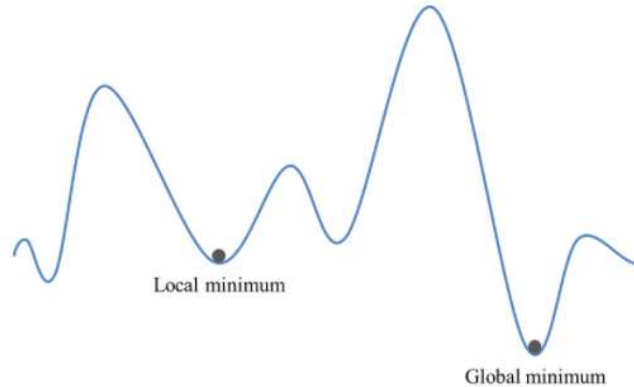
- $h_1(N)$ = 현재 제 위치에 있지 않은 타일의 개수
- $h_2(N)$ = 각 타일의 목표 위치까지의 거리 합

- 위와 같은 휴리스틱 정보(평가함수)는 값이 낮을수록 좋은 것이며, 이러한 정보를 활용하면 문제를 해결하는 과정에서 목표 노드로의 탐색이 더욱 빨라질 수 있다. 물론 이러한 방식의 탐색은 최적의 해를 보장하지는 않는다.

- 휴리스틱 정보를 활용하는 탐색 방법 중에 언덕 등반 기법(Hill-Climbing)이 있다. 언덕 등반 기법은 휴리스틱 정보를 바탕으로 현재 노드보다 더 유망한 노드를 선택하여 탐색하는 방법이다. 아래의 그림은 '현재 제 위치에 있지 않은 타일의 개수'를 평가함수로 적용하여 언덕 등반 기법으로 탐색을 진행한 결과이다.

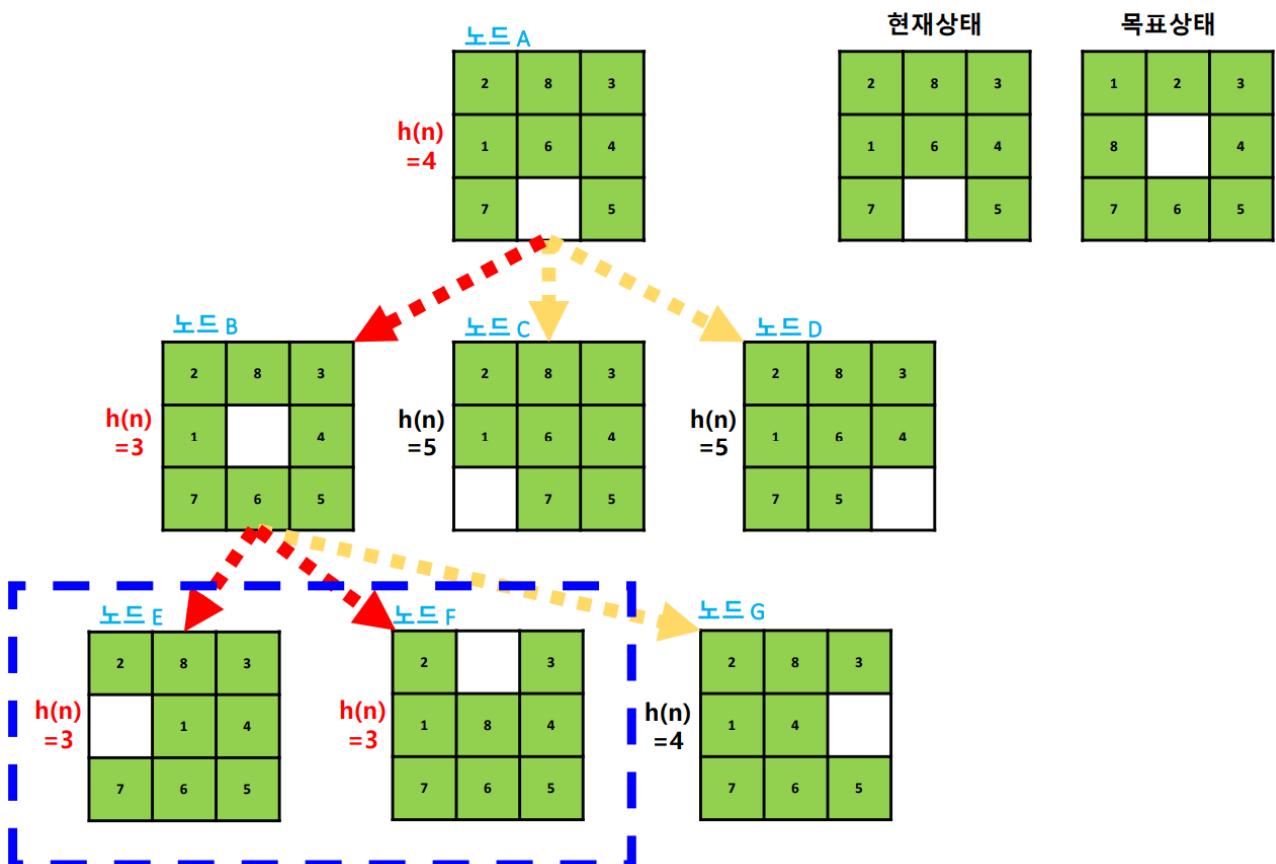


- 언덕 등반 기법은 빠르게 탐색을 진행한다는 장점이 있지만, 한 가지 치명적인 문제점이 있다. 바로 '지역 최적해 수렴'이라는 것인데 현재의 탐색 경로가 최적의 경로가 아님에도 국소적으로는 최적으로 판단할 수 있기에 탐색을 진행하게 되고, 그 결과 최적의 결과를 얻을 수 없게 되는 것을 뜻한다.



<지역 최적해 수렴 현상>

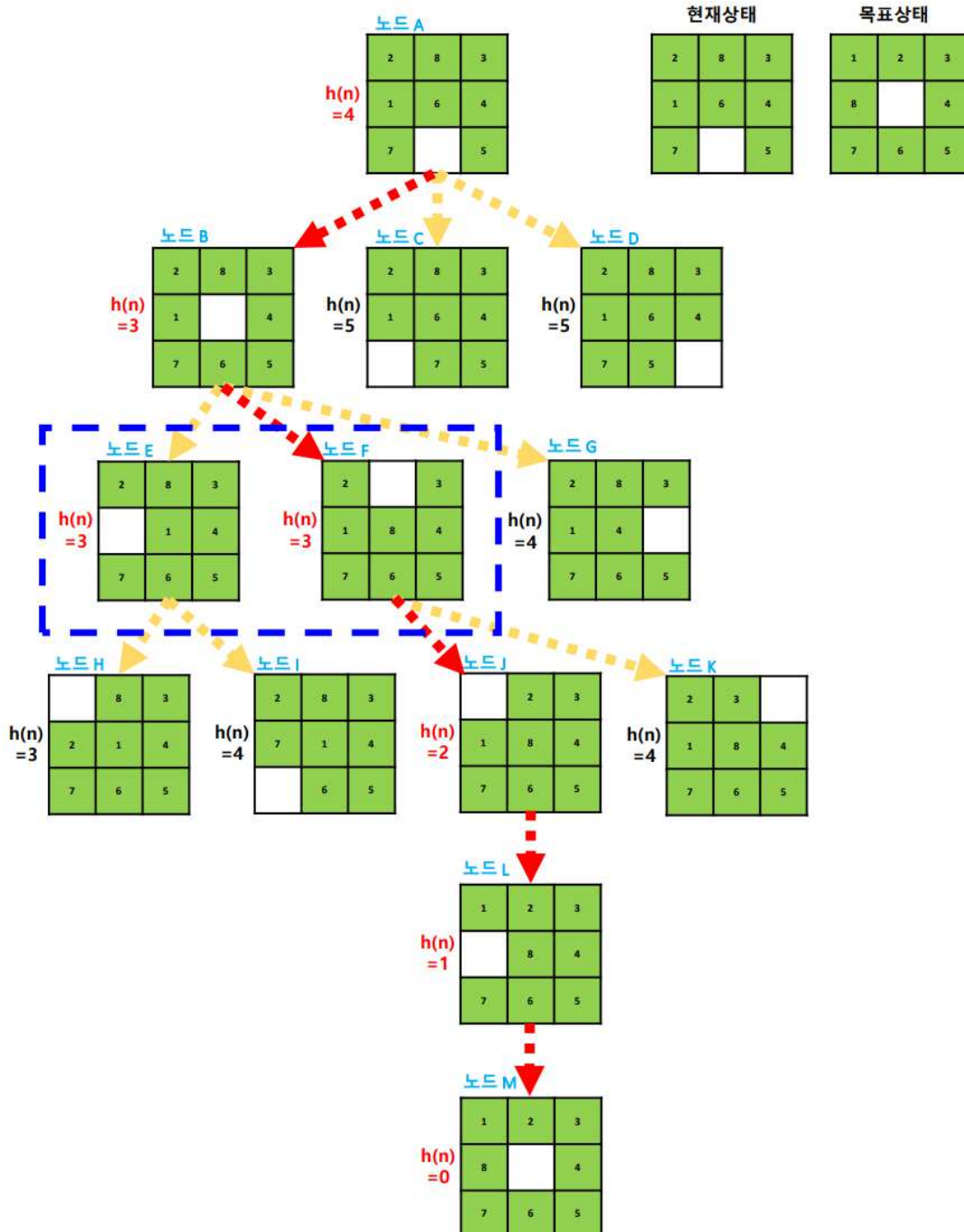
- 또한 탐색의 과정에서 자식 노드들 중에서 가장 유망한 노드가 복수인 경우, 어떤 것을 선택해야 하는가에 대한 문제가 존재한다.



- 따라서 언덕 등반 기법을 문제 해결에 직접적으로 사용할 수 있는 유용한 알고리즘으로 평가하기 보다는 휴리스틱이라는 개념을 문제 해결에 적용할 수 있도록 하는 인식적 발판으로 평가하는 것이 적절할 것이다.

3. 최상 우선 탐색의 적용

- 맹목적 탐색 방법과 비교해서 최상 우선 탐색 방법의 다른 점은 다음 노드를 결정할 때, 이미 정해져 있는 순서대로 탐색하는 것이 아니라 매 순간 가장 유망한 노드를 선택하는 것이다. 그리고 언덕 등반 기법에서는 가장 유망한 자식 노드를 하나만 선택해야 하므로 나머지 자식 노드는 고려하지 않는 반면, 최상 우선 탐색의 경우에는 가장 유망한 자식 노드가 복수일 경우, 해당 노드들을 모두 탐색의 경로에 포함시킬 수 있다는 차이점이 있다.



- 언덕 등반 기법과 비교하여 최상 우선 탐색이 가능하려면 탐색 과정에서 발생한 유망한 자식 노드들과 유망한 노드로 선택이 되었다가 탐색을 마친 노드들을 저장할 수 있는 메모리 공간이 필요하다.