

안녕하세요 (interesting dinner attitude*)IDA*팀 텀프로젝트 결과를 발표할 통계학과 최원웅이라고합니다. 저희 팀명은 수업 시간표상 끝나고 팀플을 진행하였는데, 항상 저녁 시간이 늦어져서 저녁 식사는 좋은 마음가짐으로 먹자라는 의미에서 정하였습니다...

저희 팀원은 저를 제외한 컴퓨터과학부 20학번 오규민, 이수빈 학생입니다.

먼저 발표의 순서에 대해 설명드리겠습니다.

저희 팀에서는 sliding puzzle problem을 해결하기 위해 IDA* 알고리즘을 사용하였습니다. 따라서 해당 알고리즘의 간략한 소개

저희 알고리즘의 동작 과정을 간략하게,

저희 알고리즘에서 성능 향상을 위해 개선한 부분,

저희 알고리즘에서의 시간 복잡도에 대한 순서로 발표를 하겠습니다.

1. IDA*란?

우선, 저희는 sliding puzzle 문제를 해결하기 위한 알고리즘으로 IDA*를 사용하였습니다. IDA*란 iterative deepening A_star의 약자로, 정의에서 알 수 있듯이 A_star 알고리즘을 최적화한 (보완한)알고리즘입니다. 일반적인 A_star 알고리즘은 BFS 방식을 따라 경로를 탐색하는데, 깊이가 깊어질수록 다음 노드를 탐색하는데 사용되는 메모리의 양이 많아진다는 단점이 존재합니다. 이를 보완하여 IDA* 알고리즘은 DFS 방식을 따라 휴리스틱 경로를 탐색합니다.

또한, IDA* 알고리즘은 탐색에 대한 제약이 추가됩니다. 깊이에 대한 상한을 설정하여 해당 깊이 안에 근이 존재하는지를 탐색합니다. 만약 주어진 깊이의 상한 내에서 근을 찾지 못할 경우 깊이의 상한을 늘려가며 반복적으로 경로를 탐색합니다.

최초 깊이의 상한값은 초기 상태에서의 예상 비용(휴리스틱 값)에서 시작하여 알고리즘이 반복할 때마다(제약 조건 내에서 근을 찾지 못할 때마다) 깊이의 상한값이 점차 증가합니다.

각 반복에서 다음 반복에 사용되는 깊이의 상한값은 현재 상한값을 초과한 모든 값 중에서 최소 비용을 선택합니다. 이러한 IDA* 알고리즘을 통해 저희 팀은 sliding puzzle 문제를 해결하였습니다.

2. 알고리즘의 동작 과정

저희 알고리즘의 A* 알고리즘을 따라 휴리스틱 값과 현재 상태의 값의 합을 통해 cost value를 계산하여 경로 탐색을 실시합니다. 휴리스틱 값으로는 맨해튼 거리를 이용하였습니다.

알고리즘의 동작 방식은 다음과 같습니다.

1. 최초 퍼즐 상태에서 휴리스틱 추정치를 계산하여 이를 깊이의 상한값으로 설정합니다.
2. 해당 제약을 바탕으로 깊이 우선 경로 탐색을 진행합니다.
3. 만약 해당 제약 내에서 깊이 우선 경로 탐색을 하면서 휴리스틱 값이 0이 되는 노드를 찾을 경우 정답 상태에 도달한 것으로 판단하여 경로 탐색을 종료합니다.
4. 만약, 깊이의 상한값 내에 해를 찾지 못한 경우 상한값을 증가시킨 후 깊이 우선 경로 탐색을 실시합니다.
5. 해를 찾을때 까지 위의 과정을 반복하는 방식으로 저희 알고리즘이 동작됩니다.

3. 개선점

저희 알고리즘에서는 각 노드에서 휴리스틱 값을 계산할 때 필요한 연산 수를 최적화하였습니다.

기존 코드에선 자식 노드를 생성할 때 매번 휴리스틱 값을 새로 계산하여 경로 탐색을 실시하였습니다. 저희는 이러한 연산 과정이 매번 퍼즐의 크기만큼 반복이 되기 때문에 불필요한 연산이라고 생각하였습니다. 이를 해결하기 위해서 자식 노드의 휴리스틱 값을 계산할 때 부모 노드의 퍼즐의 정보를 이용하는 방식을 사용하였습니다.

이를 통해서 휴리스틱 값을 산출할 때 최초의 퍼즐을 제외하고 단 한번의 연산을 통해서 계산되도록 계산 과정을 최적화하였습니다.

이러한 개선을 통해서 저희 알고리즘은 탐색에 사용한 모든 노드에서 휴리스틱 값을 계산하는데 사용되는 전체 연산의 크기를 줄였습니다.

4. 알고리즘의 시간 복잡도

저희 알고리즘의 시간 복잡도에 대한 분석 결과를 말씀드리겠습니다.

경로를 탐색하는 과정에서 사용되는 연산을 살펴보면,

총 탐색하는 노드의 개수만큼 휴리스틱 값의 계산이 이루어진다고 판단하였습니다.

따라서 기존 방식의 시간 복잡도는 휴리스틱 값의 계산을 위해 퍼즐의 크기만큼의 연산이 사용되었습니다. 즉, 열의 크기 * 행의 크기 * 탐색하는 총 노드의 수 만큼의 시간 복잡도를 갖는다고 분석하였습니다.

개선된 저희 알고리즘에서는 휴리스틱 값을 최초 한번을 제외하면 단 한번의 연산을 통해 이루어지므로, 시간 복잡도는 탐색하는 노드의 총 수만큼을 따른다고 분석하였습니다.

이상으로 IDA* 팀의 발표를 마치겠습니다 감사합니다.