인공지능(게임 대결)

|  |  |
| --- | --- |
| 이름(학번) | 김정호(202214205),인상민(),김록연(),임종민() |

1. 구현 과정 및 방법

MCTS, Minimax-알파베타 cutoff는 각각 장단점이 존재한다.

고려해야 할 전체 경우의 수가 한정되고 적다면 Minimax-알파베타 cutoff를 사용하여 모든 경우를 탐색하고 최적의 수를 두는 것이 맞겠지만,

해당 콰트로 게임의 전체 경우의 수인 16! X 16! 이므로 MCTS와 Minimax-알파베타 cutoff 중 하나의 알고리즘을 사용하여 구현하는 데에는 해당 과제의 요구사항인 무조건 승리하는 것과 성능부분에서 큰 제약이 존재한다.

따라서 판에 남은 빈칸의 수를 기준(두 알고리즘의 전환시점)으로 초반에는 MCTS, 후반에는 Minimax-알파베타 cutoff를 사용하여 기존에 존재했던 방대한 탐색이란 단점을 보완할 것이다. 이때 Minimax 알파베타 cutoff는 남은 빈칸을 기준으로 완전탐색하므로 고려해야 할 경우의 수를 대폭 감소할 수 있게 되어 최적의 수를 두게 된다.

2. 성능 평가

초반에는 MCTS, 후반에는 Minimax-알파베타 cutoff를 사용하므로 성능부분에서의 고려사항이 존재한다. 전환시점을 보드에서의 남은 빈칸의 수로 설정했으므로 만약 남은 빈칸의 수를 높게 설정(즉, Minimax-알파베타 cutoff를 많이 사용)한다면 무조건 이기는 것이 가능하지만 성능부분에서는 많이 취약할 것이다. 따라서 10팀은 시뮬레이션을 통해 성능평가를 하였고 최적의 전환시점을 찾을 수 있게 되었다.

(전환시점에 따른 사진 첨부하면 좋을 거 같습니다.)

“”성능 평가에 대해 추가로 작성하셔도 됩니다””

3. 주요 함수 설명

“”구현한 함수 중 설명하고 싶은 함수 작성””

4. 팀원 역할

Minimax-알파베타 cutoff(김정호, 김록연)

Minimax-알파베타 cutfoff, MCTS 알고리즘 전환시점 및 두 알고리즘 합치기(김정호, 김록연)

보고서(팀원 전체)

“”각자 맡은 역할 쓰시면 됩니다””