

Project report on

ARTIFICIAL INTELLIGENCE PLAYING THE RICOCHET ROBOTS BOARD GAME

for 02180 Introduction to Artificial Intelligence

Contributors: Jannis Haberhausen (s186398)
Jack Reinhardt (s186182)
Kilian Speiser (s181993)
Jacob Miller (s186093)

Contents

1	Game Rules	2
2	Game Representation	2
3	State Space and Complexity	2
4	Search Algorithms and Results	2
4.1	Recursive Depth-Limited Search	2
4.2	Informed Search: Breadth and Depth ??	2
4.3	A* Search	2
4.4	Custom Search Algorithm	2
5	Conclusion	2

1 Game Rules

2 Game Representation

3 State Space and Complexity

The state space and branching factor of Ricochet Robots is large, making the task of reaching the target non-trivial for both humans and AI search algorithms. The board consists of a sixteen-by-sixteen grid with walls placed in predetermined positions. With four robots that can be moved anywhere on the board (other than on top of another robot), this leaves a total of $(16 * 16)(16 * 16 - 1)(16 * 16 - 2)(16 * 16 - 3) = 4,195,023,360$ board configurations for a single wall setup and target placement. Depending on the board setup, some states may not be reachable, but the state space still remains large. Each robot can move in any of the four directions on the board until it reaches a wall or another robot in its path. Assuming that each robot has an obstruction in one of the four directions decreases the possible moves for each robot to three, giving the search algorithm a branching factor of $4 * 3 = 12$. This assumption will be true in most cases since the robot must be stopped by an obstruction before moving in a different direction.

The large state space and branching factor makes the problem of reaching the goal state difficult for traditional AI search algorithms with limited memory and time. In order to simplify the game and drastically reduce the size of the state space and the branching factor, we implemented Ricochet Robots in a way that allows the user to choose the size of the board (16x16 or 6x6) as well as the number of robots. In general, the size of the state space can be approximated by $(w * h)^n$ and the branching factor can be approximated by $n * 3$, where w is the board width, h is the board height and n is the number of robots. This reduced implementation of Ricochet Robots allowed us to play and test our AI algorithms without surpassing our memory and time limitations.

4 Search Algorithms and Results

4.1 Recursive Depth-Limited Search

4.2 Informed Search: Breadth and Depth ??

The team decided to develop a informed breadth first search to guarantee finding always the goal state by using as few moves as possible. However, this leads as mentioned in 3 this leads to a tree with up to 16^n knots.

- Difficulty of setting the limit to a certain level

4.3 A* Search

4.4 Custom Search Algorithm

5 Conclusion