Chess AI

Michelle Shu

February 13, 2014

1 Introduction

2 Minimax Search Algorithm

The minimax algorithm can be applied to zero-sum multiplayer games, including chess. It is a method that an agent may use to choose actions in the game, relying on the premise that the players have opposite goals and that all players will play optimally to maximize their chance at arriving at a goal state. The minimax algorithm works toward the goal of maximizing a utility value of the game (that reflects its proximity to a winning terminal state). Additionally, it assumes that its opponent is trying to minimize the same utility value. Hence, the game tree is composed of alternating layers of max nodes and min nodes connected by edges representing game moves, where the value of a max node is the maximum of its children's values and the value of a min node is the minimum of its children's values.

The minimax player chooses the move for which its opponent can do the least damage by selecting the action that will result in the highest expected utility from the current state (the root of the search tree). For instance, in Figure 1, the minimax player will choose the move that leads to the state represented by its middle child (min node with value 9) to achieve the maximum utility value of 9.

In the optimal scenario, the computer would be able to explore the minimax tree all the way down to its terminal nodes (nodes representing the end of the game). But for the game of chess, this is computationally infeasible towards the beginning of a game, due to the sheer number of possibilities that would have to be explored.

3 Depth-Limited Minimax Implementation

4 Iterative Deepening Minimax

4.1 Results: Finding Mate in 2

Best move found at depth 1 has value 1072884021 Best move found at depth 2 has value 1072884021 Best move found at depth 3 has value 2147483647 Check! Best move found at depth 1 has value 2147483647 Best move found at depth 2 has value 2147483647 Best move found at depth 3 has value 2147483647 Check! Checkmate!

5 Heuristic

6 References

Marina Sirota, Francine Anene and Eric Mayefsky. "Strategies and Tactics for Intelligent Search". http://www.stanford.edu/~msirota/soco.

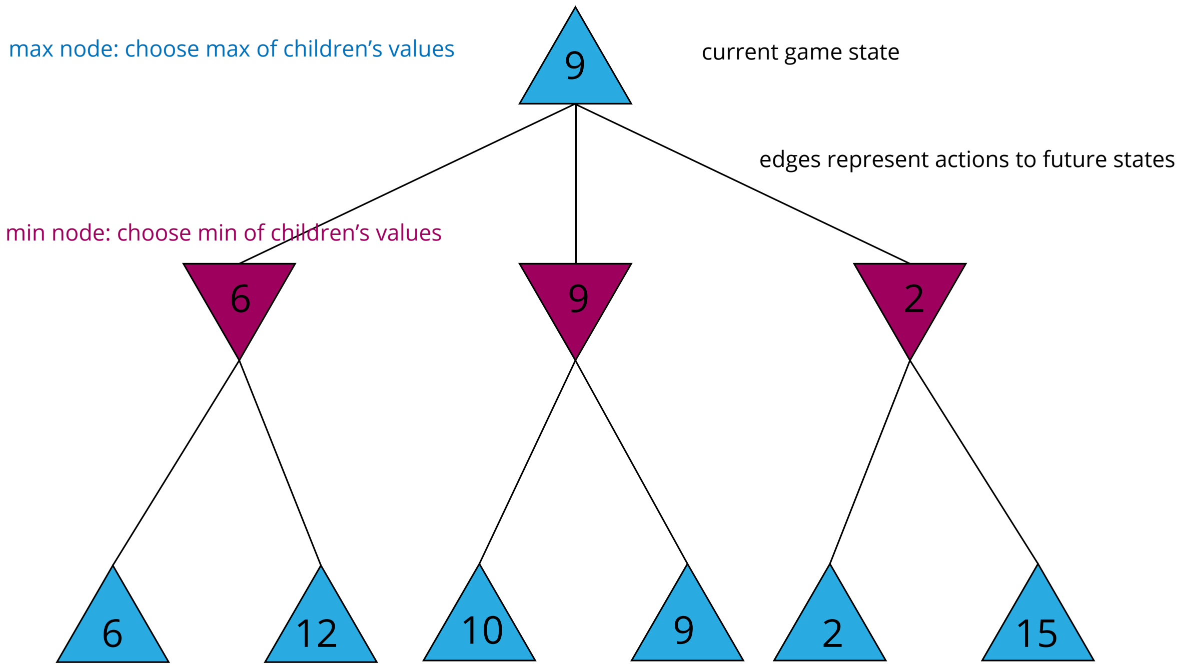

Figure 1: **Minimax Tree**

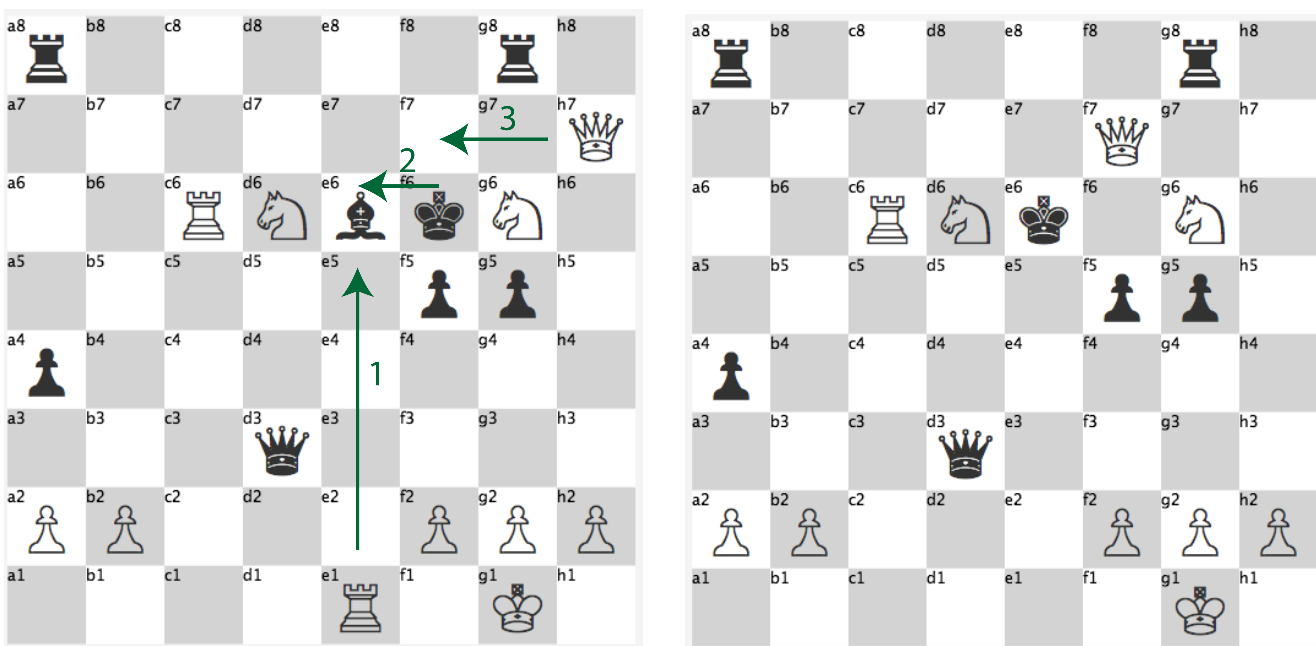

Figure 2: Mate in Two