Jonathan Sullivan
Udacity Artificial Intelligence Nanodegree
February 20, 2017

AI Research Paper Summary

Review of Game Tree Searching by Min / Max Approximation by Ron Rivest, MIT

This paper introduces a number of game-playing algorithms. It starts by making the case for using the generalized mean values to approximate min and max values. The paper then introduces the notion of game trees. The paper then touches on minimax search with alphabeta pruning. It mentions how as of now this is the best method we have for playing games.The paper then makes a case for penalty based search method and the "min/max approximation" heuristic. The paper then compares a new technique using the formally discussed method with that of minimax search with alphabeta pruning.

The paper's goals or techniques introduced (if any). This paper introduced the ideal of penalty based search. With minimax search with alphabeta pruning we are essentially doing a DFS on a tree that is a subset of the game tree , However using a penalty based algorithm we can use BFS when we are uncertain about which move to make and the more certain we become the more we can move towards DFS. Also by approximating max/min from generalized means 2 paths that look equivalent under a minimax search do not look equivalent in max/min approximation. Max/Min approximation is able to find the better move by also taking in to account the second best and third best moves. This allows us to transverse our game tree with more intelligence. The author then test this algorithm performance, using the minimax algorithm with alpha-beta pruning as a baseline.

In conclusion we see that the penalty based search with the "min/max approximation" heuristic performed better and worse than the minimax search with alphabeta pruning. It is all dependent on how you express time. In the case were time is measured in seconds the minimax search with alphabeta pruning does a lot better. However when time is measured in steps penalty

based search with the "min/max approximation" heuristic performed better. This is mainly because of the "min/max approximation" heuristic is expensive to calculate.