Paper summary - Game Tree Searching by Min / Max Approximation

This paper introduced a new technique for searching in game trees, based on the idea of approximate the generalized mean value when doing maximizing and minimizing selection. In contrast to minimax and alpha beta pruning algorithm, this algorithm decides which leaf node to expand based on a penalty weighting evaluated between its ancestor and itself, whereas alpha beta pruning prunes the subtree with out of range alpha or beta values. From almost 1000 games of Connect-Four, this method was proved to be superior than minimax search with alpha beta pruning for same number of calls to the get move routine.

The key idea for min/max approximation algorithm is to approximate with the generalized mean-value for the maximization and minimization. Since the approximation has continuous derivative with respect to all the arguments. This allows the computer to evaluate that which value is most heavily dependent while choosing the next node to be expanded.

By experiments, when CPU time rather than calls to get the move routine is the more limiting resource, minimax search with alpha beta pruning play better, because it takes less calculation(basic arithmetic) to consider which leaf to expand next or if a subtree can be pruned. However, with same number of calls to get move routine, min/max approximation performs better because it makes a better decision on which node to expand.

The experiment was to play 6*7 Connect-Four game with 49 different starting positions with 5 different CPU time limit, and 5 different move limit, with each AI moves first once. Therefore, 49 * 5 * 2 = 490 games played for each limited resource scenario. The experiments result showed that when CPU time was limited, min/max approximation AI won 186 out 490, minimax searching with alpha beta pruning won 239 out of 490, and the rest 65 games were ties. On the other side when number calls to the get move routine was limited, min/max approximation won 249 out 490, whereas minimax searching with alpha beta pruning won 190 out of 490, and left 51 ties.

The experiment also suggested that the implementation of minimax searching with alpha beta pruning called the get move routine 3500 times per second, whereas the implementation of min/max approximation called the routine 800 times per second. Therefore, this algorithm is more relevant when special-purpose hardware is used or when the get move operation is more expensive to call.

It is worth to note that the penalty based scheme requires the tree to be explicitly stored to apply the operation. In addition, penalty based scheme need to generate all successors from the tip to compute the weighting. In contrast, some successors might be pruned based on known information.

In conclusion, min/max approximation outperforms minimax searching with alpha beta pruning and iterative deepening in game tree searching when both are restricted to the same number of calls to move operation.