Game Tree Searching by Min/Max Approximation

Goal

The overall goal of this paper is to assert that "A method is needed which will always expand the node that is expected to have the largest effect on the value" to obtain expert-level play. The author's proposed the min/max approximation method to identify important lines of play. These approximations are applied to a generalized mean-value operator. The result is a good approximations with respect to all arguments to provide a path to the next best move.

Techniques

The generalized mean values are simply the mean of the values of the possible moves. They help identify which move will have maximum effect vs. those the have smaller values which have smaller effect. This sensitivity helps approximate the which leaf value in a game tree will have the most effect on the game's root. "For a small game, the game tree can be explored completely, so optimal play is possible. On slightly larger trees minimax search with alpha-beta pruning may produce optimal play even though small fraction of the game tree is explored". The author's used pruning while utilizing various depth search techniques under time-limits. Once, the time limit was reached the last search result was returned. Penalty-based iterative search methods applied a weight to every edge in the game. Bad moves are penalized more than edges representing good moves. To identify the weight, "reverse approximation" was implemented.

Results

The author's chose Connect-Four as the basis for their experiment. The penalty-based heuristic constant, 0.05, was chosen prior to testing. As the match's resource time per turn increased alpha-beta pruning techniques win ratio also increased. As the match's total moves per turn increased minimax approximation recorded higher win totals. The author notes, that penalty-based schemes may not perform well unless they are given a large amount of memory to work with. Also, like my implementation for the project, no information was carried over from a search at one depth to the search at the next and children of a node were searched in order of their static evaluations, best-first.