

Research Review of Game Tree Searching by Min / Max Approximation

The paper "Game Tree Searching by Min / Max Approximation" by Ronald L. Rivest, mainly talks about a penalty-based iterative min / max game tree search based on approximating the min and max operators with generalized mean-value operators. This summary of the paper briefly goes through the techniques used and the result for Connect-Four Game.

Techniques Used

Generalized mean-value operators

Generalized p-mean of a, $M_p(a)$ is defined as $M_p(a) = (\frac{1}{n} \sum_{i=1}^n a_i^p)^{1/p}$, which is a good approximation to max or min of a, for large positive or negative values of p, but does not have continuous derivatives with respect to each variable a. Thus, it can be used to find the leaf whose value are most important for root's value.

Searching a Game Tree using iterative heuristics

Iterative heuristics does not grow the tree in a uniform depth, instead, it tends to search deeper in the branches that are more important.

Penalty-based iterative search method

Penalties or weight, are assigned to each edge in the game tree, which represents if this move is better or worse than its siblings (bad moves have more penalty than good moves). And penalty at leaves are defined to special value ω . So that when search the game tree, we can always start with the tip node with lease penalty.

Min / max approximation

In min/max approximation heuristic, the penalties are defined in terms of the derivatives of the approximation function. Here, the approximation function is generalized mean-value operators.

Results

This experimental result is based on Connect Four game, and the Min/Max Approximation plays against alpha-beta pruning. The static evaluation function is defined as return integer 1-1023 where 1 representing a win by Red and 1023 representing a win by black (other details see the paper). The penalty in this game is defined as "0.05 plus the absolute value of the difference between the natural logarithm of the value of the node and the natural logarithm of the value of his "best" sibling. Two resource bonds, elapsed CPU time and number of moves, are used in the experiment. When elapsed CPU time are limited, alpha-beta pruning wins more times. When number of basic move subroutine are limited, min/max approximation performs better. The experiment shows that, the min/max approximation should reduce the computational overhead per call of each move. And further validation is needed and the approach need to be refined to become more efficient and competitive.