Research Review on “Game Tree Searching by Min / Max Approximation by Ron Rivest, MIT”

The author in this paper presented a new method for searching min/max trees than normal minimax method with alpha-beta pruning. The goal of this new method is to provide a more efficient way for agents to choose the best move in the game tree. During the game tree searching process, the new method uses iterative search heuristics to expend the game tree partially, meaning at each step, a node of the current tree is chosen and the sub-tree of that node is added to the tree, then the heuristic values of the chosen node and its ancestors are updated (backed-up) by the sub-trees. The criteria of choosing the node to expand is that the node to be expanded is the one with smallest penalty values, which is defined as the negative of logarithms of the derivatives of the generalized mean-values, defined as below,

$$M_p(a) = \left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{1/p}$$

This function provides very good approximation to min and max values when the factor “p” approaches to negative infinity and position infinity accordingly. Moreover, the generalized mean-value function is a continuous derivative thus more suitable when searching into deeper levels because the derivatives measure the sensitivity of the root value to the changes of the values of the subset of root. When searching for the best moves down into the game tree, derivatives of the generalized mean value is taken. However, it’s not clear how to optimally choose the parameter p.

The author uses the game “Connect-Fore” to experiment the new method. The different resource bounds are used to compare the new method with regular minimax with alpha-beta pruning. The first measurement is elapsed CPU time measured in second; the other one is number of calls to the “move” subroutine. The results show that when using CPU time as resources limit, the regular alpha-beta pruning method wins more, while when using number of calls to subroutine as resources limits, the new method performs better. The analysis of the result shows that the overhead of the computation caused the new method ran much slower than minimax with alpha-beta pruning in which 3500 calls per second vs 800 calls per second in the min-max approximation method. This overhead should be reduced to make this method more efficient.

Following the experiment result analysis, the author discussed some general features of penalty-based schemes, including that the penalty-based scheme takes more memory to work as the tree being searched in needs to be explicitly stored, and that the penalty-based scheme sometimes focuses on improving the value of the node so that redundant computations are made, etc.

At last, the author brought up some open problems that should be considered in improving this min-max approximation method.