Game Tree Searching by Min/Max Approximation

This paper describes a new technique for searching in game trees, based on idea of approximating the min and max operators with mean-value operators. It uses the fact that for any set A =[a_1, a_2, … a_n], the p-mean values[$(1/n \sum (a_i)^p)^{1/p}$] come closer to min and max values of the set A for p = ∞ and p = -∞ respectively. The partial derivative of the p-mean w.r.t to any a_i of the set is in the range 0 to 1.

In a two person, zero sum perfect information game between min and max, the current node stores the state of the game and all the possible futures game states are at next level (children). Each level of the tree alternatively takes min and max positions respectively starting with the max at the root. In this game tree the completely optimal solution can be found by going down till the leaf and propagating the utility value to the parent. The parent node selects the node with maximum utility if it's in max level or the one with minimum utility if it's in the min level. This way the best possible next move from any node is selected.

In most games of significant size going down all the way to terminal node (leaf) is not possible so we choose a method called iterative deepening where the tree is searched to finite depth, increasing the depth incrementally until a timer expires. When the search reaches the end of current depth iteration, we calculate the static value of the node called iterative search heuristic and propagate the value up to the root of current partial tree as described above.

In penalty based search the edges connecting a node with its children are given a non-negative penalty(weights) such that the edge that leads to better move is penalized less compared to others. This paper discusses which child nodes of the current node should be expanded first during iterative deepening by using **Min max approximation**. The "Min Max Approximation" heuristic is special case of the penalty-based search method, where the penalties are defined in terms of the derivatives of the approximating functions i.e. the utility function is replaced by partial derivative with a large value of p.

This approximation is used to guide the selection of the next leaf node to expand. The approximations allow one to select efficiently a particular leaf node whose value affects the root value the most. The paper also presents its case by posting the experimental results of 1000 games and comparing it with traditional Minimax search with alpha-beta pruning. Due to the fact that approximations take heavy computational time, which add extra overhead on CPU resources, further work is needed when CPU time is scarce.