Game Tree Searching by Min / Max Approximation

The paper introduces a new technique for searching in game trees, based on the idea of approximating the min and max operators with generalized mean-value operators. It introduces a method which will always expand the node that is expected to have the largest effect on the value.

Min/Max approximation method attempts to focus the computer's attention on the important lines of play. The key idea is to approximate the "min" and "max" operators with generalized mean-value operators. These are good approximations to the rain/max operators, but have continuous derivatives with respect to all arguments. This allows the method to define the "expandable tip upon whose value the backed-up value at the root most heavily depends" in a nontrivial manner. This tip is the next one to be expanded, using the iterative heuristic. One of the ideas of this paper is that by using the generalized mean values to approximate the min and max functions, we can identify in an interesting way that leaf in a game tree upon whose value the value at the root depends most strongly. This is done by taking derivatives of the generalized mean value functions at each node and using the chain rule. This leaf will be the one to expand next.

A normal heuristic function for a finite tree C of configurations with root s. We split C into subsets Min and Max depending on whose turn it is to play. For each c Ɛ C we let S(c) denote the set of c's successors (or children). To move from configuration c a player selects some d Ɛ S(c). Configurations with no successors are called terminal configurations - T(C) will denote the set of terminal configurations (leaf nodes). Each leaf t Ɛ T(C) has an associated value or score v (t)

$$v(c) = \begin{cases} v(c), & \text{if } c \in T(C), \\ \max\limits_{d \in S(c)} v(d), & \text{if } c \in Max \backslash T(C), \\ \min\limits_{d \in S(c)} v(d), & \text{if } c \in Min \backslash T(C). \end{cases}$$

The paper uses a different class of heuristics called Penalty based iterative heuristics, which assign a penalty to the edge between each parent and child node such that edge representing bad moves are penalized more than those edge representing good moves. The lowest sum of penalties between the root of a tree and the tip nodes determines which node is to be

expanded. The min/max approximation heuristic is a special case of this heuristic, where the penalties are defined in terms of the derivatives of the approximating functions.

The actual implementation of generalized p-means is very computationally costly and due to this reason the paper uses actual min/max calculation method called 'Reverse Approximation' instead of generalized p-means. Since the generalized mean values are intended to approximate the min and max functions anyway, this may not introduce very much error. The main point of using the generalized mean values was for their derivatives, not for the values themselves.

RESULT

The game of Connect-four was chosen for the comparison between MinMax search with Alpha beta pruning and penalty based approach. For the evaluation function, a segment is defined as the set of four cells in a line. The score for a segment is considered to be 0 if it contains no tokens or tokens of both colors, otherwise it depend on the number of tokens and their color.

For evaluating the performance of the two method two resource bounds were chosen.

- Elapsed CPU time
- Number of call to get_move() function

For the resource bound Elapsed CPU time, alpha beta pruning method seems superior because min/max approximation strategy is computationally costly but it is complete reverse story in the other resource bound Number of call to get_move() function as each move in min/max approximation method has move information as compare to that of alpha beta one.

Resource bound per turn	MM wins	AB wins	Ties
1 second	41	46	11
2 second	40	42	16
3 seconds	36	44	18
4 seconds	39	52	7
5 seconds	30	55	13
Total	186	239	65
1000 moves	47	35	16
2000 moves	50	35	13
3000 moves	42	47	9
4000 moves	49	42	7
5000 moves	61	31	6
Total	249	190	51