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 is superior to minimax search with alpha-beta pruning however has higher overhead.

One of the ideas of the paper is that by using the generalized mean values to approximate the min and max functions, we can identify that leaf in the 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.

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

Generalized p-mean

}

$$\frac{\partial M_p(\boldsymbol{a})}{\partial a_i} = \frac{1}{n}\left(\frac{a_i}{M_p(\boldsymbol{a})}\right)^{p-1} .$$

The partial derivate of generalized p-mean with respect to $\mathbf{a_i}$

The main point of using the generalized mean values was for their derivatives. Not for the values itself, this is called the "reverse approximation".

The game *Connect-Four* was chosen as a basis for the paper experiments with two different resource bounds: elapsed CPU time (measured in seconds), and call to the basic move subroutine (measured in thousands of calls).

For each experiment, they consider 49 different starting positions. Each starting position was defined by specifying the first two moves of the game and for each starting position, two games were played – one with alpha-beta pruning (AB) moving first, and one with Mini/Max approximation (MM) moving first. Thus a complete experiment consist of 98 games.

TABLE 2. Experimental results

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

Notice that based on time usage alone, alpha-beta pruning seems to be superior to min/max approximation approach, however, if the comparison is based on move-based resource limits, the min/max approximation is definitely superior.

Other observations were:

- The number of distinct positions considered by alpha-beta was approximately three times larger than the number of distinct positions considered by Min/Max approximation when a time bound was in effect.

- Alpha-beta pruning called the move operator approximately 3500 times per second, while Min/Max approximation called the move operator approximately 800 times per second.

In conclusion the experimental results indicate that Min/Max approximation outplays alpha-beta with iterative deepening, when both schemes are restricted to the same number of calls to the move operator.