

Selected paper: **Game Tree Searching by Min/Max Approximation - Rivest**

Goals or techniques introduced

The paper introduces a new penalty-based iterative search method based on the idea of approximating *min* and *max* by generalized mean values.

It first starts defining generalized mean values as:

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

In order to give some context, it defines game tree, minimax, score function (evaluation function) and iterative deepening.

Finally, it arrives to the definition of iterative heuristic, which *grow* the search tree one step at a time.

So, the goal of the technique, introduced in the paper, is obtain a metric in each leaf that is used to evaluate in which node the partial tree could be expanded in order to achieve better results.

In particular, the technique presented **min/max approximation** is an special case of penalty-based search method where the score of each node is calculated using this approximating function:

$$\tilde{v}(c) = \begin{cases} \hat{v}(c) & \text{if } c \text{ is leaf of the partial tree} \\ M_p(\tilde{v}(d_1), \tilde{v}(d_2) \dots \tilde{v}(d_n)) & \text{if } c \text{ is Max node not leaf} \\ M_{-p}(\tilde{v}(d_1), \tilde{v}(d_2) \dots \tilde{v}(d_n)) & \text{if } c \text{ is Min node not leaf} \end{cases}$$

Where $\hat{v}(c)$ is function uses to calculate score in Minimax and $d \in S(c)$ c node children.

It assigns a nonnegative *penalty* or weight $w(x)$ to each edges of the tree using derivatives:

$$w(x) = -\log(D(f(x), x))$$

$$D(x, y) = \frac{\partial \tilde{v}_E(x)}{\partial \tilde{v}_E(y)}$$

where $f(x)$ is father of node x in partial tree and E indicates the function applied to the partial tree.

Each leaf of the partial tree to be expanded is going to have the sum of the edges from the root to the leaf. The less value leaf is the best candidate to be expanded.

Paper results

Min/max approximation plays in favor of those moves in which min/max could be achieved in several ways. For example, $M_{10}(35, 40) > M_{10}(30, 40)$ while $\max(35, 40) = \max(30, 40)$. So, the nodes better rated have a good backup or better secondary moves.

In a Connect-Four game experimental results, the presented technique beats alpha-beta implementation on a move-based resource limit but it loses against it in a time limited context.

The alpha-beta implementation considered almost three times positions than min/max approximation over a time limited context.

Also, the move operator is called 3500 times per second in alpha-beta against 800 times per second of the proposed implementation.

As a conclusion, the min/max approximation does a better resource allocation searching in deep in the most promising sections of the game tree and shallowly in the unpromising parts.