# Game Tree Searching by Min / Max Approximation

## Goals:

- Present an iterative method for searching minimax game trees based on the idea of approximating the "min" and "max" operators by generalized mean-valued operators. This approximation is used to guide the selection of the next leaf node to expand, since the approximations allow one to select efficiently that leaf node upon whose value the (approximate) value at the root most highly depends.

- The connection between "max and min" and Generalized mean values is that for any vector of numbers a = ($a_1$, …, $a_n$) , let "p" nonzero real number, the generalized mean values is represented with the equation
$$M_p(\boldsymbol{a}) = \left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{1/p}.$$
, for p tends to +∞, then $M_p$ = max($a_1$, …, $a_n$) and p tends to -∞, then $M_p$ = min($a_1$, …, $a_n$)
$$\lim_{p\to\infty} M_p(\boldsymbol{a}) = \max(a_1, \ldots, a_n),$$
$$\lim_{p\to-\infty} M_p(\boldsymbol{a}) = \min(a_1, \ldots, a_n).$$

- So instead of using the classical heuristic algorithms in the following $fig_1$ , we will use the approximating version like in $fig_2$

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

Fig 1

$$\tilde{v}_E(c) = \begin{cases} \hat{v}(c), & \text{if } c \in T(E), \\ M_p(\tilde{v}_E(d_1), \ldots, \tilde{v}_E(d_k)), & \text{if } c \in Max\backslash T(E), \\ M_{-p}(\tilde{v}_E(d_1), \ldots, \tilde{v}_E(d_k)), & \text{if } c \in Min\backslash T(E), \end{cases}$$

Fig 2

## Results:

- Experimental results from almost 1,000 games of Connect-Four' suggest that our scheme is superior to minimax search with alpha-beta pruning, for the same number of calls to the move routine.

- The implementation had higher overhead, so that further work is needed before it becomes competitive when CPU time per turn is the limiting resource.