Research review of Game Tree Searching by Min / Max Approximation:

Technique:

The paper introduces new techniques of game-playing and search of game trees using minimax principle, evaluation functions, alpha-beta pruning, heuristic techniques.

The underlying mathematical principle of Generalized mean values are applied to to the problem of searching game trees.

What is generalized mean values - it is a generalization of arithmetic mean and geometric mean for n positive numbers.
Example:
Arithmetic mean(AM) of a1, a2, a3...aN is AM = (a1+a2+a3...aN)/N
Geometric mean(GM) of a1, a2, a3...aN is GM = $\sqrt[N]{a1.a2.a3...aN}$
Let's simplify it for N = 2, then
AM = (a1+a2)/2 and GM = $\sqrt[2]{a1.a2}$
If we consider the AM-GM inequality then
(a1+a2)/2 >= $\sqrt[2]{a1.a2}$
=> (a1+a2)/2 >= $(a1.a2)^{1/2}$
=> $((a1+a2)/2)^{2}$ >= a1.a2

will be equal only if **a1=a2**, we can generalize this for N integers AM = GM if a1=a2=a3…=aN

The above concept can be generalized for power mean rule which is shown below.

$$M_p(a_1, a_2, \ldots, a_n) \equiv \left(\frac{1}{n}\sum_{k=1}^{n} a_k^p\right)^{1/p},$$

where p = non zero real number. The minimum and maximum are defined as follows:

$$\begin{aligned} M_{-\infty}(a_1, a_2, \ldots, a_n) &\equiv \lim_{p\to-\infty} M_p(a_1, a_2, \ldots, a_n) \\ &= \min(a_1, a_2, \ldots, a_n) \\ M_{\infty}(a_1, a_2, \ldots, a_n) &\equiv \lim_{p\to\infty} M_p(a_1, a_2, \ldots, a_n) \\ &= \max(a_1, a_2, \ldots, a_n). \end{aligned}$$

Now, the value of a root in the binary tree can depend on either the maximum or minimum of its leaves which can be evaluated using chain rule.

Searching the game tree using MAX/MIN functions:

Solving 2 player game: MIN and MAX taking turns, MIN/MAX will use the search tree to find the next move.

Minimax Algorithm:

Search the game tree in a depth first search(DFS) to find the value of the root.

1. Generate the whole game tree to leaves.
2. Apply functions to leaves.
3. Select a proper value of the node depending on if it is MAX or a MIN function.
    a. MAX Node: select the max of its child values
    b. MIN Node: select the min of its child values.
4. When the tree reaches the root select the MAX value and following corresponding move.

Evaluation Functions:

Estimates the board configuration for a player at a given point in time.
Assigns a score based on the player and its opponent, typical values are MIN = -inifinity(loss) to MAX = infinity(win).

Alpha Beta pruning:

- DFS to generate a partial tree
- Iterative deepening
- Cut off search by applying static evaluation function alpha = highest value choice of a node (MAX) and beta = lowest value choice of a node (MIN)
- Pass current values of alpha and beta down to child nodes during search
- Update alpha and beta during search, MAX updates alpha at MAX nodes and MIN updates beta at MIN nodes
- Prune whenever alpha >= beta, thus significantly reducing time for optimal search.

Results:

The outcome of 98 games, in which 49 times minimax(MM) moved first and 49 times alpha-beta (AB) moved first, for time bound turns AB outperformed MM since pruning strategy of AB is much faster compared to MM, but for move based resources MM outperformed AB since MM will always perform better if the tree can be drawn full instead of partial (MM is much more resource intensive in terms of optimization).