# Game Tree Searching by Min / Max Approximation*

Summary of paper by Sanyam Chaudhary: AIND – Jan 2017 cohort

## Summary of paper's goals:

Paper discussed current MinMax with alpha beta pruning method and suggested an improvement over it. In one-line purpose of the paper ***is to explain a method which identifies, in a game tree which node to expand next***.

The improvement was done in two ways

a) Instead of expanding every relevant node (irrelevant are pruned nodes by alpha beta pruning) paper suggested approximating which *node* to expand next. It is done by assigning the derivatives of generalized means at each node. Algorithm then uses that value to determine which node to expand next. Unlike Alpha beta; this algorithm falls under the category of iterative heuristic; i.e. it grows tree one step at a time.
b) While implementing; algorithm approximates the formulae and uses a weight>0(starts with weight 0) termed as Penalties to assign to each node; Penalties of a node is the sum of penalties from that node up until the root node. Using that value; algorithm determines which current leaf node has maximum impact on root node. Key difference here as opposed to regular AB search is that node at a different depth could be expanded.

Important points to note about this and this class (iterative heuristic) of algorithms.

a) Unlike depth first search schemes; this requires trees to be explicitly stored (as the expansion isn't uniform) hence memory requirements are large.
b) Penalty based scheme like this one are more geared towards improving the value of estimate at the root rather than towards selecting the best move.

## Summary of paper's results:

Experiments were conducted with 49 different starting positions of Connect-Four game. Game was based under two resources constraints one is Time bound 1 to 5 seconds with 1 second interval and Moves bound 1000 – 5000 moves with 1000 moves interval.

In Time based categorie AB came out superior by roughly 15 % however in moves based bound MM approximation came out on top by roughly 25%. In addition to that, number of distinct positions considered by AB was roughly *3 times* of that of approximation method.