#Game Tree Searching by Min / Max Approximation

###Introduction

This 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.

#####Techniques used
Consider a two-person zero-sum perfect information game between players Min and Max which begins in a starting configuration s with Max to move, after which they alternate turns. The game defines a finite tree C of configurations with root s. We split C into subsets Min and Max depending on whose turn it is to play. For each c in C we let S(c) denote the set of c's successors (or children) To move from configuration c a player selects some d in S(c); his opponent must then move from configuration d. Configurations with no successors are called terminal configurations--these form the leaves of the tree. The game stops when a terminal configuration is reached.

The general process of minimax decision is defined by the pseudocode here. (https://github.com/aimacode/aima-pseudocode/blob/master/md/Minimax-Decision.md)

Exploring a game tree can result in visiting too many leaves. Hence the author, Ronald L. Rivest has devised a method "min/max approximation" to prune the search trees. A non-negative "penalty" (or "weight") is assigned to every edge in the game tree, such that edges with bad moves are penalized more than those with good moves. Penalties are defined in terms of the derivatives of the approximating functions.

A cost function is applied to calculate the penalties on every edge. The penalty is defined as `w(c) = -log(D(f(c), c))` where `D(f(x))` represents the derivative of function `f(x)`. We would then expand the tree over the tip with the *least penalty* after doing summation over all the nodes. The penalty is multiplied with the values obtained from leaf nodes to find out the best branch of the game tree to proceed further.

#####Results
The game of Connect-Four was used with resource bounds being either number of moves or CPU time. Based on time usage alone, alpha-beta pruning (https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning) seems to be superior to the implementation of the min/max approximation approach. However, based on move-based resource limits, min/max approximation is definitely superior to generic alpha-beta pruning.

Also, the number of *distinct positions* considered by alpha-beta pruning was approximately three times larger than the number of distinct positions considered by min/max approximation when a time bound was in effect. Overall, the results are extremely encouraging.

#####Reference
R. L. Rivest. 1987. Game tree searching by min/max approximation. Artif. Intell. 34, 1 (December 1987), 77-96. DOI=http://dx.doi.org/10.1016/0004-3702(87)90004-X