I read the article: **Game Tree Searching by Min / Max Approximation**
By: Ronald L. Rivest

This paper introduced the technique for game trees searching, based on the idea of the min and max heuristic with generalized scoring operators for each node of the tree. The combinatorial explosion of possibilities in a game, and even special-purpose hardware soon reaches its limits when analysing all the possibilities for a determined move

The method, "min/max approximation," attempts to focus the computer's attention on the important lines of play. The key idea is to approximate the "min" and "max" operators with generalized mean-value operators. This allows to define the "expandable tip upon whose value the backed-up value at the root most heavily depends" in a nontrivial manner. This tip is the next one to be expanded, using game search heuristic. This technique is basically iterative deepening with alpha-beta pruning using a penalty based scoring to evaluate each node.

After making experiments the authors come to the conclusion that:

First, we note that penalty-based schemes--like all iterative schemes-- requires that the tree being explored be explicitly stored. This algorithm would not perform well unless they are given a large amount of memory to work with.

Second, we note that the penalty-based schemes are oriented towards improving the value of the estimate at the root, rather than towards selecting the best move to make from the root. For example, if there is only one move to make from the root, then a penalty-based scheme may search the subtree below that move extensively, even though such exploration can't affect the decision to be made at the root.

Third, we note that penalty-based schemes may appear inefficient compared to depth-first schemes, since the penalty-based schemes spend a lot of time traversing back and forth between the root and the leaves of the tree, whereas a depth-first approach will spend most of its time near the leaves. We imagine that the penalty-based schemes could be adapted to show similar efficiencies, at the cost of not always selecting the globally least-penalty tip to expand.

Fourth, we observe that penalty-based schemes do spend some time evaluating non-optimal lines of play. However, the time spent examining such lines of play decreases as the number of non-optimal moves in the line increases, according to the weights assigned to those non-optimal moves.

The "min/max approximation" heuristic will allocate resources, searching shallowly in unpromising parts of the tree, and deeper in promising sections. We might also call this approach the "decreasing derivative heuristic," since the nodes are expanded in order of decreasing derivative .

It is important to note that the efficiencies exhibited by alpha-beta pruning can also appear with this scheme. Once a move has been refuted (shown to be non-optimal), its weight will increase dramatically, and further exploration down its subtree will be deferred. However, this depends on the static evaluator returning meaningful estimates.