Review Summary of the paper “Game Tree Searching by Min / Max Approximation” by Ronald L. Rivest ,Laboratory for Computer Science, MIT Cambridge
https://people.csail.mit.edu/rivest/pubs/Riv87c.pdf

Summary:

In the paper,we will discuss the key findings of the paper “Game Tree Searching by Min/Max Approximation” by Ronald L. Rivest from Laboratory for Computer Science,MIT Cambridge.

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. MinMax along with Alpha-Beta Pruning and Iterative Depth Search played an important role in reducing the computational burden of deterministic game playing seach tree.But,still there is scope for improvement in this direction.This paper discusses a new technique to decrease the computational burden of game playing search tree called Min/Max approximation.

The key idea is to approximate the "min" and "max" operators with generalized mean-value operators. These are good approximations to the rain/max operators, but have continuous derivatives with respect to all arguments. This allows us to define the "expandable tip(node) 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 our heuristic.

This paper introduces 'Penalty-based iterative search methods” which assigns a nonnegative "penalty" (or "weight") to every edge in the game tree such that edges representing bad moves arc penalized more than edges representing good moves. The idea is then to expand that tip node which has the least penalty.

The "min/max approximation" heuristic is special case of the penalty-based search method, where the penalties are defined in terms of the derivatives of the approximating functions.

Key Results from this paper:

Experimented on game of Connect-Four.

Two different resource bounds were used: elapsed CPU time (measured in seconds), and calls to the basic "move" subroutine (measured in thousands of calls). Note that for the move bound, the min/max heuristic must explicitly pay for moving down the tree from the root every time.

Based on time usage alone, alpha-beta with iterative depth search seems to be superior to min/max approximation approach.

Based on move-based resource limits, the story is reversed: min/max approximation is superior to Alpha-Beta with Iterative Depth Search.

Also observed that the implementation of minimax search with alpha-beta pruning called the move operator approximately 3500 times per second, while implementation of the min/max heuristic called the move operator approximately 800 times per second. When special-purpose hardware is used, or when the move operator is expensive to implement, the move-based comparison would be more relevant. For software implementations more development of the min/max approach is needed to reduce the computational overhead per call to the move operator.

Penalty-based schemes like all iterative schemes-- requires that the tree being explored be explicitly stored. Unlike depth-first search schemes (e.g. minimax search with alpha-beta pruning), penalty-based schemes may not perform well unless they are given a large amount of memory to work with.

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. By contrast, the B* algorithm, another iterative search heuristic, is oriented towards making the best choice, and will not waste any time when there is only one move to be made.

Penalty-based schemes as presented require that a tip be expanded by generating and evaluating all of the tip's successors. Many search schemes are able to skip the evaluation of some of the successors in many cases.

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.