Summary of the <u>Game Tree Searching by Min/Max Approximation</u> by Ronald L. Rivest, MIT published in 1988

Goals and Techniques:

Consider a two-person zero-sum perfect information game between players.
The m*inimax algorithm with alpha-beta pruning* is used to determine the best possible move. Searching a game tree has a exponential growth of children and because of this combinatorial explosion hardware soon reaches its limits. *Heuristic approximations* are needed to handle difficult games another popular approach is i*terative deepening*. Also *iterative heuristics* can be used. Iterative heuristics “grow” the search tree one step at a time. The tree grown by an iterative heuristic need not be of uniform depth: some branches may be searched to a greater depth than others.

The paper read proposes an iterative heuristic for searching in game trees, based on the idea of approximating the min and max operators with generalised mean-value operators. These are good approximations for the min/max operators, but we want to have continuous derivatives with respect to all arguments. This allows to define a “expandable tip” upon whose value the backed- up value at the root most heavily depends. This tip then will be the next to expanded, using the proposed heuristic.
The heuristic proposes a *penalty-based iterative search method*. A nonnegative “penalty” or “weight” is assigned to every edge in the game tree such that edges representing bad moves are penalised more than edges representing good moves. In the next step the tip node with the leas penalty will be expand.
It is not clear how to “optimally” choose the penalty parameter p. As p gets large, the heuristic should grow very deep but narrow trees. For small, the heuristic should grow rather broad trees.

Experimental Results:

In the experiment *Connect-Four* with a frame of 6 x 7 array of cells is used. The two players are Black (maximizing player) and Red (minimizing player).
For this heuristic a segment is a set of four cells in a line. The score for a segment depends on the number of one players tokens in it.
The static evaluation for a non-winning position is the sum of three components:
1. The neutral value 512
2. A “move bonus” of 16 the player whose turn it is to play
3. The sum, over all possible segments, of the score for that segment.

The game was implemented in C on a DEC MicroVax Workstation in 1988. At that time this experiment was run in parallel overnight on ten such Workstations.

The experiments demonstrate that this approach can produce play superior to that produce by minima search with alpha-beta pruning, for the same number of calls to the underlying “move” operator. However, when CPU time rather tan calls to the move operator is the limiting resource, minimax search with alpha-beta pruning seems to play better.