Game Tree Searching by Min/Max Approximation

The research conducted by Mr. Rivest at MIT is focused on improving the results of searching game trees. Mr. Rivest notes that while there has already been a lot of positive work done in this space, with techniques such as min/max with alpha-beta pruning, there is still room for improvement, especially given the large number of possibilities in a game like chess. The problem asserted is how to "always expand the node that is expected to have the largest effect on the value." The solution provided in this paper is "min/max approximation."

The key idea, as stated in the paper, is to "approximate the min and max operators with generalized mean-value operators." Mr. Rivest argues that being able to have continuous derivatives of all arguments allows for finding the expandable tip whose value the backed-up value at the root most heavily depends. Basically, generalized means are valuable because they're more sensitive than the simple min or max functions which allows us to be more precise.

When game trees are so large that evaluation functions are necessary, this technique requires only a single static iterative evaluator. The method explained in the paper utilizes a penalty-based iterative search method in which every edge in the game tree is assigned a nonnegative penalty. Then, expand the node that has the smallest penalty. Further, the "min/max approximation" heuristic is a special case of the penalty-based search method which utilizes derivatives of the approximating functions. Using derivatives, the heuristic measures the sensitivity of the root value to changes in the tip values and will ultimately recommend expanding the tip with the largest sensitivity.

I suspect that many people may have issues with this method because of the computational difficulty of computing the generalized p-means, which the author does not try to not address. Choosing different p-values at the various levels of the tree is one alternative

proposed by the author to help with the overhead. The results given by Mr. Rivest exacerbate this point: “However, when CPU time rather than calls to the move operator is the limiting resource, minimax search with alpha-beta pruning seems to play better.” On the positive, given the same number of calls to the move operator, the approach explained in this paper was superior than minimax with alpha beta pruning. Maybe if computing power continues to improve this method will become much more reasonable for games with large search trees.