Game Tree Searching by Min / Max Approximation

The paper presents an iterative method for searching minimax game trees. The algorithm approximates the “min” and “max” operators by generalized mean-valued operators. It is used to pick the next leaf node upon whose value the value of the root most highly depends. The game of Connect-Four is used to analyze the method.

The approximation tries to ensure that resources are spent on the important lines of play. Mean-valued operators have continuous derivatives with respect to all arguments. The expandable tip (on which the value of the root most heavily depends) is determined by taking derivatives of the generalized mean value functions at each node and using the chain rule.

The authors argue that generalized means are more suited for "sensitivity analysis" than the min or max functions used by the Minimax algorithm. The discontinuous nature of the min/max functions make them difficult to use, whereas the generalized means function is continuous. Generalized means approximation belongs to a class of heuristics called iterative heuristics which grow the search tree one step at a time.

The heuristic is an example of the penalty-based search method, where the penalties are defined in terms of the derivatives of the approximating functions. The algorithm assigns a penalty to every edge in the game tree such that edges with bad moves are penalized more than edges with good moves. The penalty of a leaf is the sum of the penalties of all the edges between that node and the root. The node with the min penalty is expanded next. The successors of that node are added to the tree. The evaluator function is run at the new leaves which give new backed-up values to the leaves' ancestors. The penalties are updated for all the edges involved in the operation.

One challenge implementing the above heuristic is to deal with the computational difficulty of computing the generalized p-means. According to the paper, there are tradeoffs between different values of p. For large p, the heuristic should grow very deep narrow trees. For small p, the heuristic should grow rather broad shallow trees. One performance drawback is that the Tree being explored has to be explicitly stored unlike minimax search with alpha-beta pruning. One by-product is that the penalty-based schemes are oriented towards improving the value of the estimate at the root, rather than making the best move from the root.

Empirical results show that the scheme outplays alpha-beta with iterative deepening, when both schemes are restricted to the same number of calls to the move operator. The algorithm allocates resources optimally, searching shallowly in unpromising sections of the tree, and deeper in promising sections.