research_review

June 19, 2017

1 Goals and techniques

The goal of the paper was to introduce a technique to enhance a game tree searching algorithm by using min/max approximation. The aim of the min/max approximation, is to expand the node in tree, which has the largest expected effect on the value of the root node. In order to find this best "expandable tip", the authors suggest to use generalized mean-value operators, which are an appoximation of the min/max operators, but have continuous derivations with respect to all arguments. The min/max approximation is used in the framework of iterative heuristics, which grows the search tree one step at a time: at each step a tip node of the current tree is chosen and the children of that node are added to the tree, as a consequence some branches of the tree might be much deeper than other branches. The question arises, how to choose which node has to be expanded. The authors present a method called penalty-based iterative search method, which assigns nonnegative "penality" (or "weight) to every edge in the tree such that edges representing bad moves are penalized more than edges representing good moves, then those tip nodes are expanded which have the least penalty. The "min/max approximation" heuristic is a special case ot the penalty-based search method, where the penalties are defined in terms of the approximating functions. In detail, this means first of all that the static evaluation function defined in equation (9) (see paper) are approximated by using the min/max approximation see equation 10. The static evalution function is a score value for each node in the tree, it is an exact value for terminal nodes and an estimate for the other nodes. Then the penalty value of a tip node is defined as the sensitivity of the root value to changes in the tip value, which is the quotient of the derivation of the approximated static evaluation function of the root node and the tip node. The penalty value is calculated for every node and the tip node with the least penalty value is expanded.

2 Key results

The authors evaluated the performance of the min/max approximtaion algorithm by letting it play 980 games of the game Connect-Four against the minimax search algorithm with alpha beta pruning. Note that in order to obtain comparable results both algorithms used the same static evaluation function. Based on the time usage alone, alpha-beta seems to be superior to the min/max approximation, on the other hand if there are move-based resource limits the min/max approximation is superior. In addition to this the authors note that the min/max approximation algorithm allocates resources in a sensible manner, searching shallowly in unpromising parts of the tree, and deeper in promising sections. And that the efficiencies exhibited by alpha-beta pruning can also appear with the min/max approximation. 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. If the static evaluator were to return only constant values except at terminal positions, the min/max approximation would perform a breadth-first search.