ARTIFICIAL INTELLIGENCE NANODEGREE

PROJECT 2

RESEARCH REVIEW

-GRANDHE SAI CHETAN

Goals & and how to accomplish it:-

The goal of the paper was to introduce a technique to enhance a game tree searching algorithm by using minimax approx. which tells us to expand the node in tree, which has the largest expected effect on the value of the root node.

To find this,

The authors suggest to use generalized mean-value operators, which are an approx. of the minimax operators, but have continuous derivations with respect to all arguments. The minimax approx is used in the framework of iterative heuristics, which grows the search tree one step at a time: at eveery step a tip node of the current tree is chosen and the children of that node are added, because of which branches of the tree might be much deeper than other branches. Now we have to decide how to choose which node has to be expanded. The authors present a method called ***penalty-based iterative search method***, which assigns nonnegative "penalty" 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 value. The "minimax approx" heuristic is a special case of the penalty-based search method, where the penalties are defined in terms of the approximating functions. The static evaluation 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.

Results:-

The authors evaluated the performance of the minimax approx. algorithm by letting it play 980 games of the game Connect-Four against the minimax search algorithm with alpha beta pruning. 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 minimax approx. , on the other hand if there are move-based resource limits the minimax approx. is superior. In addition to this the authors note that the minimax approx 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 minimax approx. Once the move is rejected (which is non-optimal), its weight will increase, and further exploration down its sub-tree will be deferred. However, this depends on the static evaluator returning meaningful estimation. If the static evaluator returns only constant values except at terminal positions, the minimax approx would perform a breadth-first search.