Summary of "Game Tree Searching by Min/Max Approximation" Paper

This paper [1] discusses minimax search of game trees, in the context of complete information, zero sum, two player games. We learned from class that for such games, an optimal playing strategy can be found by recursively traversing the game tree spawned from any game state s, and calculating the minimax value of children nodes of s. If the player in state s is the max(min) player, the child of s with the max(min) value is selected as the optimal move. For any non-terminal node t, its minimax value is the max(min) of the minimax value of its children if the player in state t is the max(min) player. The minimax value of a terminal node is given by the value of a utility function, usually defined so that terminal(leaf) nodes that lead to a win for the max player have higher values than those that lead to a loss. The minimax value of any other node t can then be calculated recursively starting from terminal nodes that descend from t and taking the minimax value of all other descendants of t.

In practice, due to the computational cost of searching deep into the game tree, implementation of the minimax algorithm searches only up to a certain depth, at which an estimate of the correct minimax value is calculated. The estimate is usually an utility function that estimates the value of a games state, and assigns bigger values to game states that are perceived to have a higher value to the max player. If time (or any other limited resource) permits, the algorithm expands the tree along a node (or several nodes as in iterative deepening) and updates the minimax values. For non-pathological games, increase in depth leads to an increase in the accuracy of the computed minimax values. The main idea in this paper is to give a prescription for deciding which node to expand on in the context of minimaxing the values of a utility function.

The paper first puts the question of deciding which node to expand in a general context. This is done by defining the notion of an appropriate cost function along edges of a game tree. An appropriate cost function is one whose value along an edge only depends on the child at that edge, its siblings, and its the parent. The total cost of expanding a non-terminal node is simply the sum of the costs along all edges that connect that node to the root node. The prescription for deciding which node to expand is simply to select the expandable node with the minimum cost. After each expansion, the minimax values and cost functions of ancestors of the expanded node are updated.

To apply these ideas to the minimax algorithm. The author introduces certain generalized means which are differentiable approximations to the min and max function. Let us call this approximation mimimax'. The cost of expanding a node is then defined as the negative of the logarithm of the derivative of the nodes parent minimax' value with respect to the nodes minimax' value. By the recursive definition of minimax' and the chain rule, this cost function satisfies the requirements of an appropriate cost function. The author describes implementation of these ideas in detail. Some experiments were performed using the game Connect-Four, and the author concludes that the minimax approximation strategy seems to outperform minimax search with alpha beta pruning with iterative deepening, when the number of calls to the move function[1] is a limited resource. However the minimax approximation strategy is outperformed by alpha beta pruning when the time to make a move is the limited resource. The minimax approximation strategy is outperformed when time is a limited resource because of the expensive

[1] In my understanding, the notion of the number of calls to the move function is not well explained. It cant be referring to the function called when it is a players turn to make a move, since this function is called until a winner emerges. Therefore it cannot be a limited resource. My impression is that it is referring to the number of nodes expanded in order to calculate the minimax values. This is a reasonable basis of comparing to alpha beta search with iterative deepening which expands all the nodes up to certain depth at each iteration. I would like to know what the instructor thinks about the authors use of the number of calls to the move function.

operation of calculating the cost function.

The paper concludes with some discussion and some open problmes. Most interestingly listing the problem of how this approach may be combined effectively with more traditional approaches, and how the ideas discussed may be parallelized.

References

[1] Game Tree Searching by Min/Max approximation, Ronald L. Rivest, Laboratory for Computer Science, Massachusetts Institute of Technology, 1986.