Game Tree Searching by Min/Max Approximation

Ronnal L Rivest

(summary)

The paper introduces a particular penalty based iterative search heuristic. A penalty based iterative search heuristic is a method of gameplay that grows the tree one step at a time. The basic elements of the methodology are two: 1) a heuristic value function and 2) a penalty function. The algorithm begins with a minimal tree (call it explored tree) and the heuristic value of each node is calculated per the min-max algorithm and the heuristic value function. At each step, from the currently explored tree a node is chosen to explore, based on a penalty function (the node with the minimum penalty). The node is then expanded to the next level, and an updated penalty function and heuristic value for each of the new “tips” are calculated (tips are defined as terminal nodes of the explored tree, i.e., notes that have either not yet been explored, or are terminal nodes of the game). The heuristic value of all the nodes in the updated tree can now be updated as well. Authors note that it is only necessary to update the ancestors of the node we just expended (which includes the node itself).

This allows for a growing the tree in a stepwise fashion with a ready move available at each step. This is similar to iterative deepening but with two key differences: 1) At any point, the tree can be explored at different depths along different paths, and 2) A search from the root node does not need to be restarted at every iteration

This paper presents a particular penalty function for these penalty based iterative search heuristics. The penalty function is suggested based on two key observations:

1. A penalty function that measures, for any given tip c of the currently explored tree, the potential impact on the backed up heuristic value of the root node (say s) is potentially a good penalty function, since picking a tip that is likely to have the maximum impact on the value of the root node should make sense (in this regard, the paper selects a monotone function of the negative of the impact on the value of the root node as the penalty function)
2. Generalized mean values provide a good approximation of the min and max functions, are are differentiable, thus allowing us to approximate the potential impact on the heuristic value of the root node by any tip in the currently explored tree, in the spirit of point 1 above

The author combines these ideas, and suggest calculation of the penalty function by adding weights to the edges between nodes and their children. The weight is given by negative log of the derivative of the value of the parent node, with regard to the value of the child node.

The author tests the strategy on the game of Connect-Four against a standard player that does alpha-beta pruning with iterative deepening. The experiment contains two parts, one with a CPU-time bound and another one with move bound (where number of calls to the move subroutine are limited). The results show that the MM algorithm underperforms the AB algorithm in with a 186:239:65 ratio of Win:Loss:Tie when the constraint is CPU-time. However, when the constraint is number of moves, MM outperforms AB with a 249:190:51 ratio.

The authors note the general constraints and drawbacks of a penalty based iterative search heuristic, and that further improvement of their MM strategy is possible.