Research review of game tree searching by min-max approximation

Synopsis

The paper use a different searching heuristic to approximate the min and max operators by generalized mean-value operators, which gives propability measure to select branches to explore further. According to the experiment results, the method has a better performance than alpha-beta pruning when the step counts are limited, but consumes more memory and computational resources.

Techniques

generalized mean as min-max approximation

$$ M_{p} (a) = (1/n \sum_{k=1}^n a_{i}^p) ^ {1/p} $$

For large positive and negative p values, the generalized mean can approximate the max and min. But it has continuous partial derivatives with respect to each a_{i}, thus is more suitable for "sensitivity analysis". We can use chain rule to the derivatives and determine which node will affect the root most.

iterative search heuristics

Compared to the fix-depth tree searching and iterative deepening method implemented in our Isolation project, the paper choose a iterative search heuristics to grow the search tree one step at a time. When a expandable tip in the partial tree is selected and expanded, its successors are added to the partial tree, and new backed-up values of its ancestors are updated by the values of the new tree leaves.

using generalized mean as penalty-based iteratives

Every edge in the game tree is assigned a non-negative weight which can be calculated from the value function of the father nodes and its siblings. The good moves have smaller weight than the bad moves. We sum the weights of the edges between the expandable node and the root of the subtree to calculate the penalty of the expandable node. If the penalty value weighs the smallest, when this node is searched for further depth, the root value will be greatly affected. On the other hand, when the other siblings of this node are searched, the root approximate value will not depend upon these nodes as much as the minimal penalty node. We also trace the route of nodes from the subtree root to the selected minimal-penalty expandable node. When this expandable node is searched in further depth, its child nodes are added, values of all the route nodes are updated, and the penalty of this node will grow. We can stop searching the expandable children of this node further and switch to its expandable sibling with less penalty and expand in another direction. So it will save a lot of search moves.

search by min/max approximation

The paper define a sensitivity to measure how many the root value will change corresponding to the change of tip node, which is computed by the ratio of derivatives of the generalized mean approximate value. The tip nodes with the largest sensitivity are picked for further expansion. The weight on the edges between the father node and its child are also defined as the negative of the logrithm of the sensitivity between the two nodes. The tip node with the smallest penalty as the sum of weights along the path is chosen to expand, which is equivalent to the node with the largest sensitivity.

Implementation and results

As the power and root computation costs in calculating the generalized mean is large, we can reverse the approximation, and use the original backed-up node values instead to get the weight. The weight formula retains the chain rule properties of the approximation, and takes account of the less optimal move controlled by the parameter p. When p is large, you can go confidently into the deep and narrow trees, which penalize the less optimal moves more.

The Connect-Four game on a 6 * 7 board is played when black moves first. The static evaluator return integers from 1(a win by red) to 1023(a win by black). For the non-wining positions, we consider the scores of all possible segments with four cells in a row, when three blacks in a segment is scored 30, two blacks is scored 20, one black is scored 10 and vice versa for the red. By adding the move advantage for the player(16 for black and -16 for black) and the neutral value(512) to all segment scores, the static evaluator for this position is assigned and rescaled from 2 to 1022 if necessary.

Under the two different limit settings as fixed time and fixed moves, the experiment compare the results of our min/max approximation penalty and minimax search with alpha-beta pruning methods. When the time bounds for each run is restricted, the approximation method performs not as good as the alpha-beta pruning method. But when the number of steps for each run is restricted, the approximation method wins more than the alpha-beta pruning method, which results from better choice for selecting moves to obtain minimal penalty for the tree expansion.

In [ ]: