Research Review: Game Tree Searching by Min/Max Approximation

Victor Osório

Techniques Introduced

The paper looks the method of searching into min/max games trees based trying to find a better way to prune branches with a more efficiently than others algorithms.The authors refers to Alpha-Beta Pruning (and its successors) and B* as a good examples of how a min/max game based tree can be explored. But as said, no solution was satisfactory, so "*A method is needed which will always expand the node that is expected to have the largest effect on the value*".
The key idea is approximate the min/max functions with a continuous function, avoiding the lost of data during the prune. The Generalized Mean Values is the function presented for that.

$$M_p(a) = \left(\frac{1}{n} \sum_{i=1}^{n} a_i^p \right)^{\frac{1}{n}}$$

The $M_p(a)$ is more sensitive than min/max and it has some interesting facts that helps searching in min/max trees:

$$\lim_{p \to \infty} M_p(a) = max(a_1, ..., a_n)$$
$$\lim_{p \to -\infty} M_p(a) = min(a_1, ..., a_n)$$

A penalty-based iterative search method is a tree search where for each node we assign a penalty. This penalty means that explore that node is a bad choice, so the algorithm should choose the node with less penalty to explore.
The introduced method is a penalty-based where the penalty are defined in terms of the derivatives of $M_p(a)$, D(s, c). The derivatives of $M_p(a)$ indicates the sensitive of the root to be changed by that branch, so the proposal method wants to expand the node with maximum D(s, c), that is the node with less penalty. The method create a most promising line of play for both players.

Paper's results

About the implementation, the paper does not present good results. In a search tree, some resources are limited, for a game play, the most important resource is time. The paper present two types of resources: time and explored nodes (moves).
If the search is time bounded, AlphaBeta Prunning has a better performance than Min/Max Approximation, but if the search is explored move bounded, Min/Max Approximation has a

better performance. This behavior is explained because is expensive to calculate the penalties for each node. With time bound, the Alpa-Beta Prunning can explore tree times more moves than Min/Max Approximation.
But the paper showed that is possible to create a better way to explore search tree than Alpha-Beta Prunning using penalty based search, but at that time it was an inefficient because of evaluating time. Penalty based search can be adapted to show similar efficiency.
A good conclusion is that, in Iterative Search, the algorithm should be very fast, the implementation should not spend too much time deciding in which node should be expanded, because this lost of time means less nodes explored.