

Research Review

Game Tree Searching by Min/Max Approximation

This paper introduces us new methodologies of searching my Min/Max Approximation game trees and compares them with the minimax search algorithm with alpha-beta pruning. The main objective of this paper is to reduce the time of search for the best move. The methods discussed in this paper include Penalty-based iterative search methods and Searching by Min/Max Approximation using generalized mean methods than just minimax methods

The author of this paper explains the main reason to choose the generalized mean is because generalized mean offers a more suitable method for sensitivity analysis. Min/Max functions are in general discrete and as we know that differentiating a discrete function gives rise to a discontinuous function which will not be appropriate for accurate sensitivity analysis. But the generalized means function is a continuous and differentiable function and hence generalized means provides us a better sensitivity analysis. Therefore, by implementing the generalized mean values methodology to approximate the minimax function values, it is very much possible to identify the leaf node that has the greatest impact on the root node.

For some familiar games (for example: Chess) as we increases the value of d one unit at a time, the accuracy of the value computed for the root seems to improve. However, there are games (called *pathological*) where the accuracy decreases with increasing values of d . But, in this paper the author assumes that the game of analysis will be *non-pathological*.

Hence, to explore the non-pathological games, a different class of heuristics are the *iterative* heuristics, which "grow" the search tree one step at a time. At each iteration a tip node (or leaf) of the current chosen tree, and add the successors of that tip node to the search tree. Then the values that are provided by the static evaluator at the newly added nodes are used to provide a new backed-up values to the nodes ancestors. Examples of such iterative techniques are the Berliner's B^* algorithm, Nilsson's "arrow" method, Palay's probability-based method, and McAllester's "conspiracy number" method.

The major challenge addressed in this paper is expandable tip c of E should be picked. This paper a new answer to this challenge. What this paper does not address is how one might gauge the accuracy of the current estimate of the value at the root, in case one wishes to use a termination condition based on accuracy instead of a termination condition based on time.

In Penalty-based iterative search methods author presents a general method for choosing which leaf to expand in an iterative method. In this method, we assign a non-negative "penalty" (or "weight") to every edge in the game tree such that edges representing bad moves are penalized more than edges representing good moves. Here, the penalty of a tip c is the sum of the

penalties of all the edges between node c and the root s . We then expand the tip node that has the least penalty.

In **Searching by min/max approximation** heuristic which is a special case of the penalty-based search method, where the penalties are defined in terms of the derivatives of the approximating functions. In this we have to focus more on the derivatives with respect to each argument than focusing on approximate values. $D(s, c)$ measures the sensitivity of root value $v_E^{\sim}(s)$ to changes in the tip value $v_E^{\sim}(c)$. We should expand next that expandable tip c with the largest value $D(s, c)$. This will reduce the uncertainty in $v_E^{\sim}(s)$ in the most efficient manner.

After describing the above mentioned methodologies, the author then explains how to achieve the computation of **generalized p-means**. However, to get the approximate computation of generalized p-means is very intensive and the whole purpose of heuristic evaluation function being fastest is defeated. Because the computation is very complex, the author introduced a new idea called **reverse approximation**. With reverse approximation idea the weight $w(c)$ is computable from the equation $w(c) = \log(n) + (p - 1) \cdot (\log(v_E^{\sim}(d)) - \log(v_E^{\sim}(c)))$ where c has n siblings, and where d is that sibling of c with the most favorable value $v^{\wedge}(d)$ for the player to move from $f(c)$.

The paper presents us with the performances of the proposed methods against conventional minimax search with alpha-beta pruning for over 1000 games of Connect-Four. It is observed that based on the time usage alone, alpha-beta seems to be superior to our implementation of the min/max approximation search. However, the story reverses if we observed the performance based on move-based resource limits.