Game Tree Searching by Min/Max Approximation - summary

Josip Matijević

March 2017

This is a short summary of the paper Game Tree Searching by Min/Max Approximation by Ronald L. Rivest.

The paper presents a new technique for searching game trees by approximating the min and max operators with generalized mean-value operators. The idea of this technique is to always expand the nodes that are expected to have the largest effect on the value of the root.

Unlike iterative deepening approach where we search the whole tree up to some depth and repeat this process by increasing the depth until we run out of time, with iterative heuristics we expand the search tree one step at a time. At each step we decide which leaf of the search tree we expand next and then use the values of new leaves to update the values of leaves' ancestors. The purpose of the paper is to give a new approach on deciding which leaf we should expand next.

In penalty-based iterative search methods, we assign a nonnegative penalty (weight) to each edge in the search tree in a way that the best moves have the lowest penalty attached to them. Then we always expand the node that has the lowest penalty. The algorithm described in this paper is a specific instance of this method where the penalties are the derivatives of the approximating functions called general mean values.

First we define the general mean value. For a vector of positive real numbers $a = (a_1, ..., a_n)$ of length n and a nonzero real number p, we define the generalized $p - mean$ of a to be

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

The most important facts about the generalized mean values regarding the application to game tree searching are:

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

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

This means that the generalized mean values are a good approximation to maximum and minimum values of a. The difference is that the generalized mean has continuous derivatives for each variable a_i. This enables us to identify the leaf in a game tree that impacts the root value the most.

Let y be any node in E_x, where E_x is the subtree of our search tree rooted at x. Then we have

$$D(x, y) = \frac{\partial v_E(x)}{\partial v_E(y)}$$

where $v_E(c)$ is the generalized mean value of the vector containing all children to node c or the value of our static evaluator if c is a terminal node. If s is the root of our search tree, we want to expand the next expandable node with the largest value $D(s, c)$ as $D(s, c)$ measures the sensitivity of the root value to changes in the node c.

In the context of penalty-based iterative heuristic, we can define the penalty of the edge between node x and its father $f(x)$ as

$$w(x) = -\log(D(f(x), x))$$

Since we want to expand the node with the largest $D(s, c)$, we choose the node with the least penalty, where the total penalty of the node is the sum of all the penalties of the edges between the root and the node.

Results presented in the paper show that the min/max approximation performs better than minimax search with alpha-beta pruning when the limiting factor is the number of calls to the move operator. If the CPU time is the limiting factor, minimax with alpha-beta pruning still perfoms better and further improvements to the min/max approximation algorithm are needed.