Research Review

Ronald L. Rivest
'Game Tree Searching by Min/Max Approximation'
Artificial Intelligence 34(1988): 77-96

Submitted by Rima Laidani
Artificial Intelligence Nanodegree, Udacity
08/14/2017

Overview

The paper under review proposes a method for traversing a game tree based on an implementation of an iterative and penalty-based scheme.
The new search scheme is called Min/Max Approximation and seeks to identify a path to a terminal node in the game that is expected to be most relevant in estimating the value at the root.
As the name suggests, in this method the min and max operators (as known in the minimax algorithm) are here approximated by generalized mean-valued operators.

Iterative Search Heuristics Details

The algorithm proposed in the paper falls within the iterative search heuristics framework.
A tree is iteratively grown from the root by expanding one of its expandable terminal nodes each time. The algorithm then adds the leaves' successors to the tree and updates the values of their ancestors after computing an estimate of their values. More specifically, the mentioned updates are implemented using the generalized p-mean operator. The value of p is a large positive or negative real number depending on whether the ancestor is a max or min operator.[1]
The paper tackles the question regarding which expandable leaf should be chosen by applying a penalty-based strategy.

Penalty-based Iterative Search Methods

The penalty-based approach attaches a nonnegative weight to each edge such that bad moves carry higher penalty (weight) than good moves.
The weight of an edge sprouting from a newly expanded leaf depends on the estimates of the static function of all the leaves derived from the same parent and the one of the parent itself.
The penalty of an edge is the sum of all weights encountered from the root to the edge under examination.

Min/Max Approximation Search

Min/Max Approximation is an implementation of the penalty-based iterative search scheme presented above, where the penalties are calculated as derivatives of the generalized mean values.
This can be formally expressed as follows:
Given a subtree E rooted in x and y any node in E, we define

---

[1] Please refer to the appendix below for a brief overview of the definition of generalized mean values as used in the paper.

$$D(s,c) = \partial v_E(s)/\partial v_E(c)$$

as a measure of the sensitivity of root value $v_E(s)$ to changes in tip value $v_E(c)$.
At each step we aim at expanding a tip node with largest $D(s,c)$, thus reducing uncertainty in the estimated value at the node. To do so, we can define the weight between a node x and father or x as:

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

EXPERIMENTAL RESULTS

The paper presents some initial experimental result related to the comparison of the Min/Max Approximation technique (MM) with the Iterative Deepening Minimax Alpha-Beta Pruning approach (AB).
The experimental game used was Connect-Four, where two players alternate turns by dropping their tokens in a 6x7 grid. Tokens are of two colours (one per player). A legal move is in any lowest unoccupied cell in a column. The grid is visible to both players. The winner is the one that first completes a line of 4 tokens of its colour (horizontally, vertically or diagonally).
There were 2 main types of constraints to work with: an elapsed CPU time (measured in seconds, from 1 to 5 with increments of 1 second) and number of calls to the move subroutine (from 1000 to 5000 with increment of 1000).
The experiment consists of 49 opening position, the first 2 moves (the first move being an initial setting of the game). Each opening position was played twice to allow each agent to make the first legal move. Thus, the complete experiment consisted of 98 games, played 10 times each, 5 under the time constraint framework and 5 under the limited number of move constraint.
Results showed that under the time bound, AB outperformed MM, winning ca 49% of the times. MM won 38% of the games and the AB and MM ended in a tie ca 13% of times.
However, when the games are played under the number of moves constraint, MM outplayed AB, winning ca 51% of the times, whilst the game ended in a tie about 10% of the times.

CONCLUSIONS

The paper presented a new approach to game tree searching.
Experimental results show that MM outperformed iterative deepening with alpha-beta under number of moves constraints. However, MM performed worst than the iterative deepening when CPU time was the resource bound. Also, the author outlined how the penalty-based algorithms may not perform well unless they are given large amount of memory when compared to depth-first search schemes.
Also, the proposed algorithm requires evaluating all current tree tip's successors while other schemes can skip evaluating some of them.

Finally, the proposed approach seems inefficient compared to depth first search schemes as there are many back and forth from the root to the leaves.

APPENDIX

Generalized Mean Values

The generalized mean values definition used in the paper is presented below:

Given a vector of n positive real numbers $\boldsymbol{a} = (a_1, \dots, a_n)$ and p a nonzero real number, the generalized p-mean of **a** is:

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

Of particular interest are the following aspects of the generalized p-mean:

1) for large value of p, $M_p(\boldsymbol{a})$ tends to approximate the maximum value in **a**
2) for small value of p, $M_p(\boldsymbol{a})$tends to approximate the minimum value in **a**
3) the generalized p-mean has continuous derivatives with respect to each variable a_i, making them more suitable for sensitivity analysis.