Review of "Game tree searching by Min/Max approximation"

Shreyash Gotee

August 22, 2017

Introduction

In his paper "Game Tree Searching by Min/Max approximations"[1], Dr. Ronald L. Rivest proposed using generalized p-mean values for the minimax algorithm instead of using the exact values. He proposed that this technique would lead to the selection of leaf nodes on which the root node depends heavily. The results of the experiments conducted by his team show that this technique works better than 'minimax with alpha-beta pruning' when bounded by the number of function calls that can be made to a move routine.

Techniques

By using Minimax with approximated values, this technique tries to focus computer's attention on important lines of play. These approximate values are calculated by the 'min' and 'max' operators using th formula:

$$Generalized\ p - mean, M_p = (1/n \sum_{i=1}^{n} a_i^p\ )^{1/p}$$

where $a = (a_1, ..., a_n)$ is a vector of $'n'$ positive real numbers and $'p'$ is a non-zero real number

Note that $M_0(a)$ and $M_1(a)$ give the geometric and arithmetic means of the vector a respectively. Also note that $lim_{p\to\infty} M_p(a) = max(a_1, ..., a_n)$ and $lim_{p\to-\infty} M_p(a) = min(a_1, ..., a_n)$

Large values of 'p', either positive or negative, lead to better approximations. Since we are using a generalized mean, $M_p(a)$ also has a continuous derivative. This makes it more suitable for sensitivity analysis. Furthermore, using this continuous derivative along with the chain rule helps the computer select the next best node that should be expanded.

Dr. Rivest used iterative search heuristics along with a penalty-based minimax approximation method to estimate the value of the root node. For this estimation, a static evaluation function ($\hat{v}$) that produced these estimated values of $v(c)$ was used. Whenever an expandable node was expanded, the estimated values of all its ancestors were updated. Each bad move was penalized heavily so that the computer refrained from expanding those bad nodes. In each iteration, the expandable node with the least penalty was selected for expansion.

Results

Dr. Rivest applied his method to play "Connect Four". The static evaluator was a part of the move and unmove operators. The system was bounded by (a) CPU run time, and (b) Total

Number of function calls to the move/unmove operators. Black was considered as the MAX player and Red was the MIN player. The static evaluator returned values in the range 1 to 1023, 1 being the loss for Black and 1023 being the win. 512 was a neutral position which was also the starting point of the game. The value returned by the static evaluator was a sum of the neutral value (512), a move bonus of 16 points ('+16' for Black and '-16' for Red) and the value of the segment where the next move was being played. A 'segment' was a set of four cells in a line. The scores for a segment were allotted as follows:

- Segment with no tokens or tokens of both colors - 0
- Segment with 1 Black token - 1
- Segment with 2 black tokens - 10
- Segment with 3 black tokens - 50

The scores for Red tokens followed the same system, but the values were negative.

It was observed that when played under time constraints, Minimax algorithm with alpha-beta pruning would perform better. However, when played under the constraints of function calls, Min/Max approximation method works better than alpha-beta pruning (800 calls per second to 3500 calls per second). Furthermore, they also observed that the value of 'p' was directly proportional to the degree of confidence in accuracy of values returned by the evaluation function.

Problems

Though this technique proved to work better than alpha-beta pruning in certain conditions and was also ideal for sensitivity analysis, the computations for calculating the generalized mean are very heavy. Which means that this technique can not be used on smaller or hand-held computers. Also, this technique has a very high overhead.

References

[1] Ronald L. Rivest, *Game Tree Searching by Min/Max approximation**. Laboratory for Computer Science, MIT, 1988. Elsevier Science Publishers B.V. (North-Holland)