Review of "Game Tree Searching by Min / Max Approximation" by Ronald L. Rivest

by Salih Ergut

Introduction

In the article, Rivest defines a penalty based iterative search algorithm. He first introduces minimax and alpha-beta pruning search algorithms.

For a small three minimax provides the optimal solution and for slightly larger trees alpha-beta pruning provides the optimal solution. However, as the tree size grows it is not possible to run the alpha-beta pruning in a limited execution time. For that reason tree depth needs to be limited. Since most of the times the tree does not terminate at the given d, a heuristic function is used to approximate the evaluation function. Iterative deepening selectively increases the tree search depth so that the decision is based on the most accurate evaluation for a given limited execution time.

Penalty-based iterative search algorithm

The main idea of penalty-based iterative search algorithm is to decide which leaf to expand. Rivest used a continuous generalized mean-valued operator approximation for minimum and maximum functions.

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

And as *p* becomes large $M_p(a)$ approaches to min and max depending on the sign of *p*

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

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

This way he can compute the sensitivity of each tip with respect to the tree root by taking partial derivative. The one with highest derivative (or sensitivity) is expanded as that leaf has the most uncertainty and thus have the highest effect on the root value. This way penalty-based algorithm can explore the tree deeper, and therefore allows the tree search to grow non-uniformly.

The parameter *p* impacts on the final tree structure. It's not clear how to choose this parameter optimally. One option to use a dynamic *p* value based on the different levels of the tree. A large *p* indicates the confidence in the evaluator results, and small *p* is used for cases where we have less confidence.

Following figure shows how the leaf w has been expanded and thus modifies the root value.

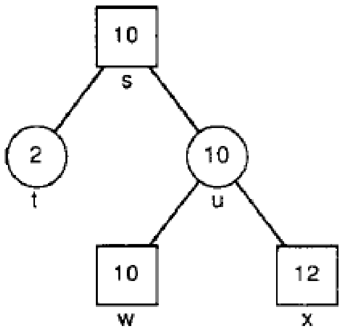

FIG. 2. A partial game tree.

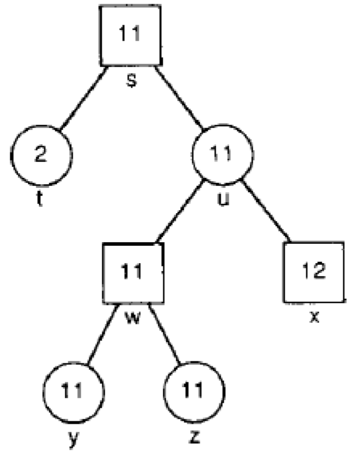

FIG. 3. An expanded game tree.

Results

He implemented a connect-four game and compared the performances of two approaches. Connect-four was chosen as it is well-known and easy to implement. Thousand games were run to observe the performance of newly identified algorithm.

During the playoffs, each strategy uses similar amount of resources such as CPU time, memory etc. so that we can compare different strategies fairly.

In order to eliminate any bias towards initial starting point 49 different locations were considered in the study. For each location two games were played so that the order of start would not have an impact on the overall performance evaluation. An opening book of 7 moves is used in the game.

Experimental results show that the proposed algorithms performs better than minimax and alpha-beta pruning In terms of number of calls to move operator. However the latter performs better when CPU time is the limiting factor.

Penalty based schemes also require larger memory. They need to store the tree during exploration

Open questions

The paper also lists several open-ended questions such as:

- Choosing the right generalized mean value functions.
- Parallelization of algorithms