Review of Game Tree Searching by Min / Max Approximation[1]

Yun Ling

This review is a brief summary of techniques introduced in this paper. It discusses about the Min/Max approximation and penalty-based schemes. In the last part, the writer also gives us some conclusions and notions

The main purpose of the paper is to develop a better decision tree searching implemented in gaming or other area. Because of the combinatorial explosion of the possibilities in game. An effective heuristic function is necessary to reduce all possible choices and gives the player an optimal result within the time limitation.

First, the writer talks about an algorithm named generalized p-mean values for calculating extremum. Compared with general min/max functions, it returns a continuous value rather than discrete value. That feature is the essential function in this paper. And it is widely used in approximate the min and max functions.

Then, the writer introduced the game tree searching. A general searching way to handle a very large tree is iterative deepening. It receives a static deep value previously. After that the function will expand each of the sub node shallower than d and calculate the backed-up values if time permits. But it is still impossible to deal with large factor trees or more complex trees. So, the writer shows us a static evaluation functions to estimate the backed-up value. When expand new node, it will recompute the node itself and all ancestor node and gives them new values. Then find next the optimal node for future expand. So it won't need to expand all node in the tree.

After explaining the static evaluation method, the writer raises a specific instance of the method name penalty-based iterative search. The main thinking is assigning each edge a weight for calculate to determine which node should be expand. And give a penalize when expand a bad node.

Next, the writer uses all the knowledge declared upon to do an experiment based on connect-four game. To conclude the experimental results. If we based on time usage alone, alpha-beta is superior to implementation of the min/max approximation approach. But if we base the comparison on move-based resource limit, the min/max approximation becomes superior. That mainly because of the Fragmentation lossage in alpha-beta pruning and the inefficiencies of min/max approximation.

Finally, the writer discussed about several features of penalty-based schemes. For example, using Penalty-based schemes need to storage the tree, which can occupies a large amount of memory to work with. And it will be inefficient because of its structure and way of evaluating the node. In some statement, penalty-base schemes are oriented towards improving the value of the estimate at the root instead of towards selecting the best move to make from the root.

This is what the main idea of the paper.

---

[1] Ronald L. Rivest, Game Tree Searching by Min / Max Approximation, Laboratory for Computer Science, MIT, Cambridge, MA 02139, U.S.A.