Game Tree Searching by Min / Max Approximation*

INTRODUCTION

The technique uses mean value operator to search a game tree efficiently. In this method, " m i n / m a x approximation," attempts to focus the computer's attention on the important lines of play. The key idea is to approximate the " m i n " and " m a x " operators with generalized mean-value operators. These are good approximations to the r a i n / m a x operators, but have continuous derivatives with respect to all arguments. This allows us to define the "expandable tip upon whose value the backed-up value at the root most heavily depends" in a nontrivial manner.

TECHNIQUE

The game defines a finite tree C of configurations with root s. We split C into subsets Min and Max depending on whose turn it is to play. The successors of the node are explored untill the terminal configuration is reached. Each leaf has an associated score v(c).

$$v(c) = \begin{cases} v(c)\,, & \text{if } c \in T(C)\,, \\ \max\limits_{d \in S(c)} v(d)\,, & \text{if } c \in Max \backslash T(C)\,, \\ \min\limits_{d \in S(c)} v(d)\,, & \text{if } c \in Min \backslash T(C)\,. \end{cases}$$

where T(C) is the set of terminal configurations of c.

ITERATIVE HEURISTIC SEARCH

The tree is searched by taking subtrees. Each partial tree is searched until there are no successors i.e when leaf node is reached. This heuristic functions gives the estimated score of each partial tree. This proves efficient since the function backs up scores from partial trees rather than all the terminal positions.

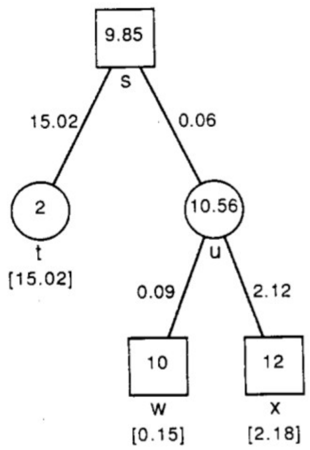
$$\hat{v}_E(c) = \begin{cases} \hat{v}(c)\,, & \text{if } c \in T(E)\,, \\ \max\limits_{d \in S(c)} (\hat{v}_E(d))\,, & \text{if } c \in Max \backslash T(E)\,, \\ \min\limits_{d \in S(c)} (\hat{v}_E(d))\,, & \text{if } c \in Min \backslash T(E)\,. \end{cases}$$

PENALTY BASED HEURISTICS

Penalty(weight) is assigned to every edge where every edge is the path from a node to its successors. The penalty is assigned in such a way that bad moves have greater penalty value than the optimal edges.

MINMAX APPROXIMATION

The figure on the right shows the minmax approximation. The min and max play is clearly viewed where the path s-u-w is chosen since this path is in more dependence with the root node.



DEPENDENCIES

This technique proves efficient for large trees. The technique leaves out some possibly optimal solutions. There are many alternate moves from root to successors. This method is highly dependent on the penalty value and reverse penalty is used.

RESULT

The min/max approximation technique tends to use lesser move operations than the traditional alpha-beta pruning technique.For software implementations more development of the min/max approach is needed to reduce the computational overhead per call to the move operator.