Min/Max Approximation in Game Tree Searching

This paper introduces a new technique for searching game trees using generalized mean-value operators. These operators are used to approximate the min/max operators commonly used in game tree searches. The main motivation for this substitution is that the generalized mean-value operators have continuous derivatives. By taking the derivatives at each node and using the chain rule we can determine which unexpanded leaf node would most affect the backed-up (i.e., mini-maxed) value at the root thus focusing the search on the most important lines of play.

GOALS AND TECHNIQUES

Define the generalized p-mean of a, M_p(a), by:

$M_p(a) = \left(\frac{1}{n}\sum_{i=1}^{n} a_i^p\right)^{1/p}$ where for p=0, we define M_0(a) as the geometric mean.

The equation M_p(a) offers two desired qualities: 1) it is a good approximation to $\max_i(a_i)$/$\min_i(a_i)$ for large/small p values respectively; and 2) the a_i values near the max (or min) have much more effect on M_p(a) than the smaller values of a_i do.

Searches of large game trees must use various optimizations to be performant. The authors use an iterative, penalty-based search using a "min/max approximation" heuristic (henceforth abbreviated as MM) with the penalties defined in terms of the derivatives of the approximating p-mean functions. Penalties are assigned to edges such that “bad” moves are penalized more than “good” ones by taking the negative log of the p-mean derivatives for a node versus its parent. The largest derivative values (i.e., the largest sensitivity of the root node to changes in the leaf node) thus corresponds to the least penalty. Since it is iterative search, the search tree grows a single node at a time by expanding the successors of a node which has the least penalty. A static evaluation function is defined which gives an estimate of the backed-up value of a non-terminal node using features of the current move configuration. The value from the static evaluator function at each of these new leaf nodes is backed-up to all their ancestors up to the root node. The path to the best expandible tip node is then traced by the end of this search algorithm.

RESULTS

To evaluate the efficacy of the MM strategy the authors chose to implement a tournament using the familiar and simple Connect-Four game. Two players were created: one MM-based and one implemented using Minimax with Alpha-Beta and iterative deepening (henceforth called AB). Almost 1000 games of Connect-Four were played between the AB and MM-based agents with the AB players and MM players each moving first half the time.

Each strategy was compared under the same resource restrictions: five possible time bounds and five possible move bounds. AB won more frequently in time-bound games as it was able to invoke over four times the number of move operations as MM per second due to the latter's higher computational overhead per call. However, for moves-bound matches, MM won over AB.

Works Cited

Rivest, Ron. *Game Tree Searching by Min/Max Approximation*, 1987.