Name : Virendra Panchal
Roll No : 211137

DOMS | Page No.
Date / /

7/11/23

# AI Assignment

Q. Explain alpha-beta pruning. How can alpha-beta pruning improve min-max search algorithm?

→ Alpha-beta pruning is a technique used in the min-max algorithm, which is a decision-making algorithm commonly used in two-player, perfect-information games like chess or tic-tac-toe. The goal of the min-max algorithm is to find the best move for a player assuming that the opponent also play optimally.

In a min-max search tree, each node represents a game state, and the edges represent possible moves. The algorithm recursively explores the tree, evaluating each node to determine the best move for the current player. It does this by assigning a value to each node representing the expected outcome of the game assuming both players play optimally.

Alpha-beta pruning enhances the efficiency of the min-max algorithm by reducing the number of nodes that need to be evaluated. It does this by maintaining two ~~val~~ values, 'alpha' and 'beta', the represent the minimum score the maximizing player is assured of and the maximum score the minimizing player is assured of respectively.

The key idea is that as the algorithm progresses along a certain path, it keeps track of the best score

found so far ('alpha' for the maximizing player and 'beta' for the minimizing player). When a node's value exceeds the opponent's 'beta' (for the maximizing player) or 'alpha' (for the minimizing player), it means that the opponent has found a better move earlier in the search, so there's no need to explore it further down that path. This is called 'pruning' because it cuts off unnecessary branches of the search tree.

Here's how alpha-beta pruning works in detail:

1. Start at the root of the tree.
2. Perform a depth-first search, evaluating nodes and updating 'alpha' and 'beta' values as you go.
3. When you reach a leaf node (a terminal game state), return the value of that node.
4. As you backtrack up the tree, update the 'alpha' and 'beta' values based on the values returned from child nodes.
5. If at any point 'alpha >= beta' for a maximizing node (or 'alpha <= beta' for a minimizing node), stop evaluating the rest of the children and prune the branch.

Alpha-beta pruning can significantly reduce the number of nodes evaluated, making it much more efficient than a naive min-max search, especially in large game trees.

In summary, alpha-beta pruning improves the efficiency of the min-max search procedure by intelligently avoid the evaluation of nodes that

Date / /

are guaranteed to be suboptimal based on the current information available. This makes it a crucial technique in game-playing AI algorithms.