DA221 Assignment report 2 (tic tac toe game)

Onkar Shinde (220150021)

Minimax Rule

The minimax rule backs up values from the children of a node.
For a MAX node, it backs up the maximum of the values of the children and for a MIN node, the minimum.

Evaluation Function

e(p) = number of directions open for Max - number of directions open for Min
e(p) = + inf if win for Max and e(p) = - inf if win for Min

- ❖ Examining b^d leaf nodes, where each node has b children and a d-ply search is performed without alpha - beta pruning.

Alpha-beta Pruning

It is not actually a new algorithm, rather an optimization technique for minimax algorithm.
It cuts off branches in the game tree which need not be searched because there already exists a better move available. It passes two extra parameters in the minimax function.

I. alpha(n): The best value that MAX currently can guarantee. It starts at infinity and only increase.
II. beta(n) The best value that MIN currently can guarantee. It starts at infinity and only decrease

Alpha-beta is guaranteed to compute the same value for the root node as computed by minimax, with less or equal computation

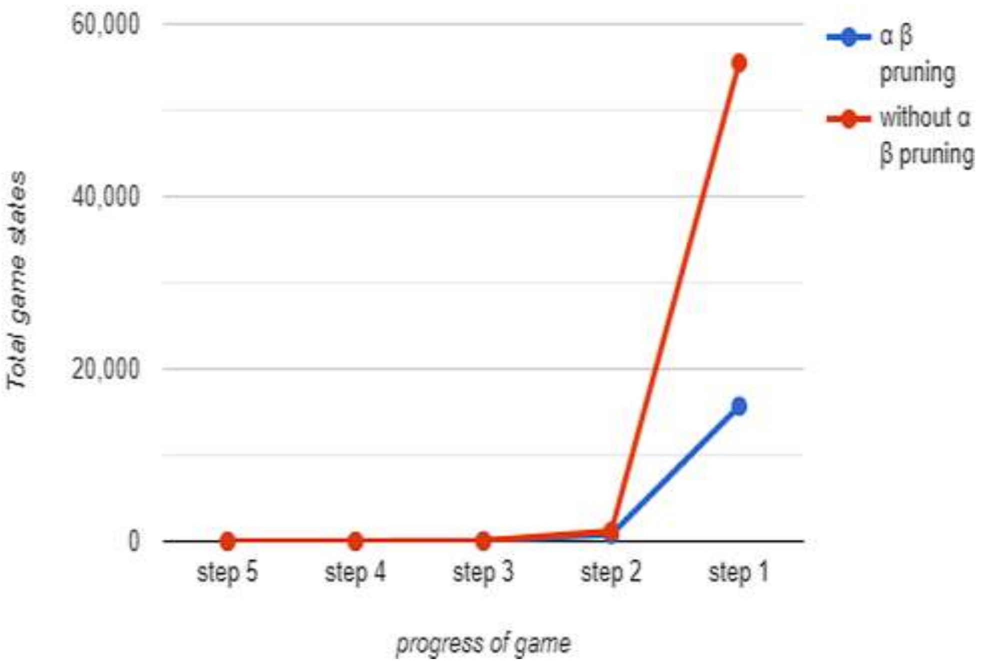
- ❖ Examine only $(2b)^{(d/2)}$ leaf nodes. Result is you can search twice as deep as minimax.

Results and Analysis

The primary objective was to compare the total number of game states explored with and without Alpha-Beta pruning. The following observations were made:

- ➢ Without pruning, the algorithm explored a significantly larger number of states, especially as the game progressed.
- ➢ With Alpha-Beta pruning, the algorithm consistently pruned branches, leading to a substantial reduction in the total number of explored states.

Below is graph plotted for played tic tac toe game:



Time analysis	
without α β pruning	**With α β pruning**
3681 ms	1000 ms

The above result is based on 10 games played