TIC TAC TOE USING APLHA-BETA PRUNING

-Prithvi Patil
-Gagandeep
-Atharv R Belagali
-Sagar Babu

Introduction

The Alpha Beta pruning is one of the fundamental concepts to start with. By using regular paper-pen game concept such concepts can be taught easily.A simple version of the minimax algorithm, stated below, deals with games such as tic-tac-toe, where each player can win, lose, or draw. If player A can win in one move, his best move is that winning move. If player B knows that one move will lead to the situation where player A can win in one move, while another move will lead to the situation where player A can, at best, draw, then player B's best move is the one leading to a draw. Late in the game, it's easy to see what the "best" move is. The Minimax algorithm helps find the best move, by working backwards from the end of the game. At each step it assumes that player A is trying to maximize the chances of A winning, while on the next turn player B is trying to minimize the chances of A winning (i.e., to maximize B's own chances of winning).

What we will be doing in our project

This proposed work presents an intuitive method to implement the Alpha-Beta pruning followed by minimax algorithm, a back-pedal algorithm that is used in option choosing from combinations of several alternatives, to achieve better understating of how Artificial Intelligence (AI) works. By combining complex algorithms with regular paper-and-pencil game Tic-tac-toe leads toward a clear and intuitive learning. Such basic games are deeply melted in our instincts thus the concept of pruning can easily be approached. Minimax is used in game playing to find the best efficient move for a player assuming that the rival player will play to win too. Later Alpha-Beta pruning will equalize the minimax algorithm. It returns the same move but it removes all the branches that will not be affecting the final part of the game. For output, a standard grid of nine squares (3x3) is used, consisting of 3 rows and 3 columns. However, this grid can further be extended to 4x4 or 5x5 to increase the complexity of the algorithm.

OBJECTIVES

- Implement Tic-Tac-Toe using min-max and alpha-beta pruning.
- Using OpenMP to parallelize the algorithm thereby increasing efficiency.
- Display the output result.

Literature Survey

- Paper [1] : Designing and Implementation of Tic-Tac-Toe-4X4 based Artificial Intelligence using Python Programming:

The primary goal for this project is to create a computer artificial intelligent based on Tic-Tac-Toe 4x4 game that show two players on who will win and who will lose the game accordingly, using the standard Minimax algorithm, it was adopted and modified as a subset of rules from best gameplay practices:

(1) attempt to win,
(2) endeavor to keep a misfortune,
(3) make a key move, and
(4) make an irregular move.

To make the game more fun and more winnable at easier difficulty levels, probabilities are introduced that the computer would find a valuable move and ignore it.

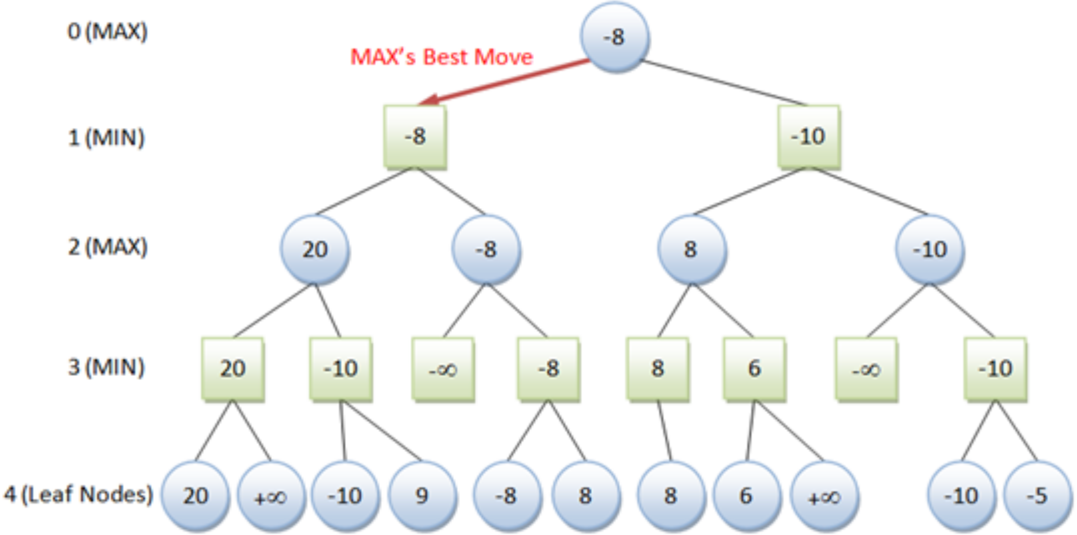
Methodology

Minimax:

Minimax is a kind of back pedal decision taking algorithm used in game trees to find the best and efficient option or the choice for the player considering that the counter player will also choose the best move. The authors of have proposed a optimally modified MINIMAX

An Intuitive Implementation Of Alpha-Beta Pruning Using Tic-Tac-Toe algorithm as Rminimax with improved results. This type of algorithms are used in board games such as TicTacToe, Chess etc. In Minimax algorithm the two players are known as maximizer(+1) and minimizer(-1). The maximizer tries to obtain the higher value while the minimizer tries to get the opposite value i-e low value.

MINMAX


0 (MAX)
1 (MIN)
2 (MAX)
3 (MIN)
4 (Leaf Nodes)
-8
MAX's Best Move
-8
-10
20
-8
8
-10
20
-10
-∞
-8
8
6
-∞
-10
20
+∞
-10
9
-8
8
8
6
+∞
-10
-5

Methodology:

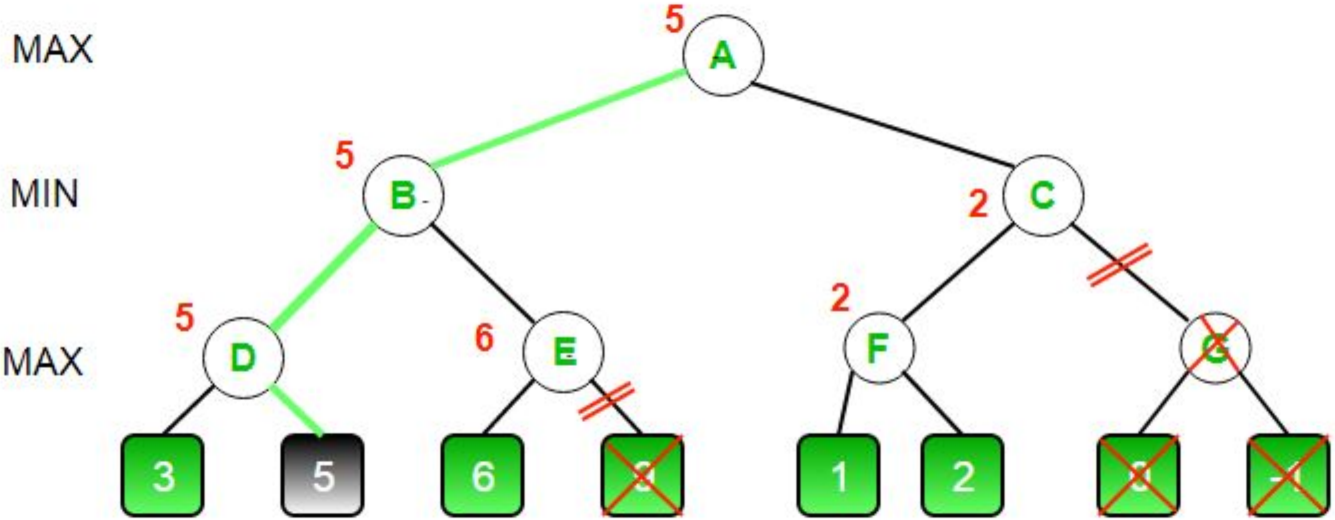
Alpha-Beta Pruning

Alpha: It is the best choice for the player having max part. In this case we want to get the higher value.

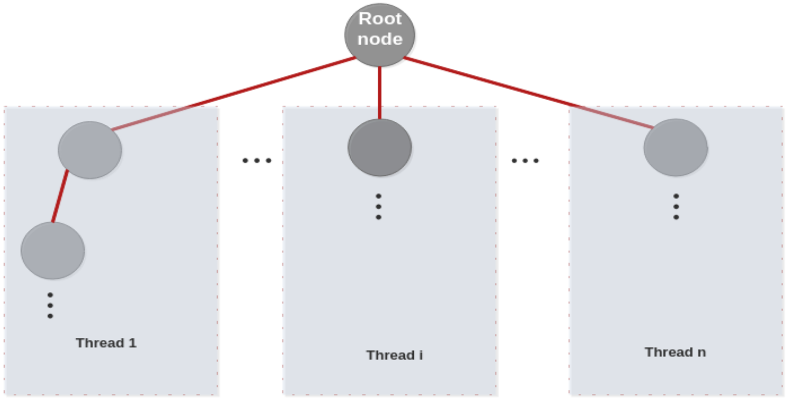
Beta: It is the best choice for min part. In this case the low value is go-to-go.

AlphaBeta pruning is applied to equalize the Minimax. This will return the best available move from the rest in other words it will trim the rest of combinations for that particular move unaffecting the game

Alpha-Beta Pruning



Parallelization



Conclusion:

We have succesfully implemented this tic tac toe project using alpha beta pruning and parallel computing. The end result is that the program can never lose , it either leads to a draw or the AI wins.