Report of Game Playing Algorithm

Name: Rahul Vijayvargiya, 245784

To conduct experiments and formulate the algorithm. I have made the Reverse Game. In the Game Playing Algorithms, we have used Mini-Max algorithm in which we are having a game tree and we are checking the possible moves and getting the best move out of it.

Short Description of the Rules of the Games:

Rules of reverse are simple:

1. We have a 8 x 8 Board and Human player is denoted by 1 and AI by number 2.
2. Player and AI take 1 turn each time and have as many places as you can.
3. Player can flip as many places as possible on the board in one or two moves.
4. Player must make at least 1 move each time.
5. Game finishes when either player cannot make any more move or board is completely full.

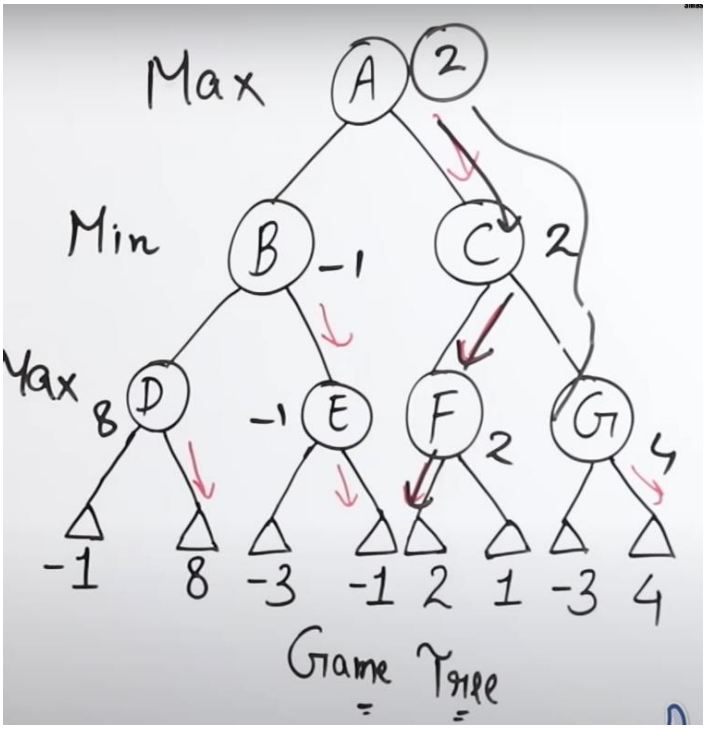
Scores are displayed in the end of the game.

Note: I would like to Explain the algorithms Briefly in my report as per the rules of the report.

Mini-Max Algorithm: Min-Max Algorithm is a recursive algorithm for choosing next move in a 2 or n-player game. It is used most often in games decision theory, statistics etc. for minimizing the possible loss for a worst-case scenario. When dealing with gains it refers to max min to maximum gain.

Time Complexity

O(b^d) Here b = number of legal moves at each point, d= depth of Tree.



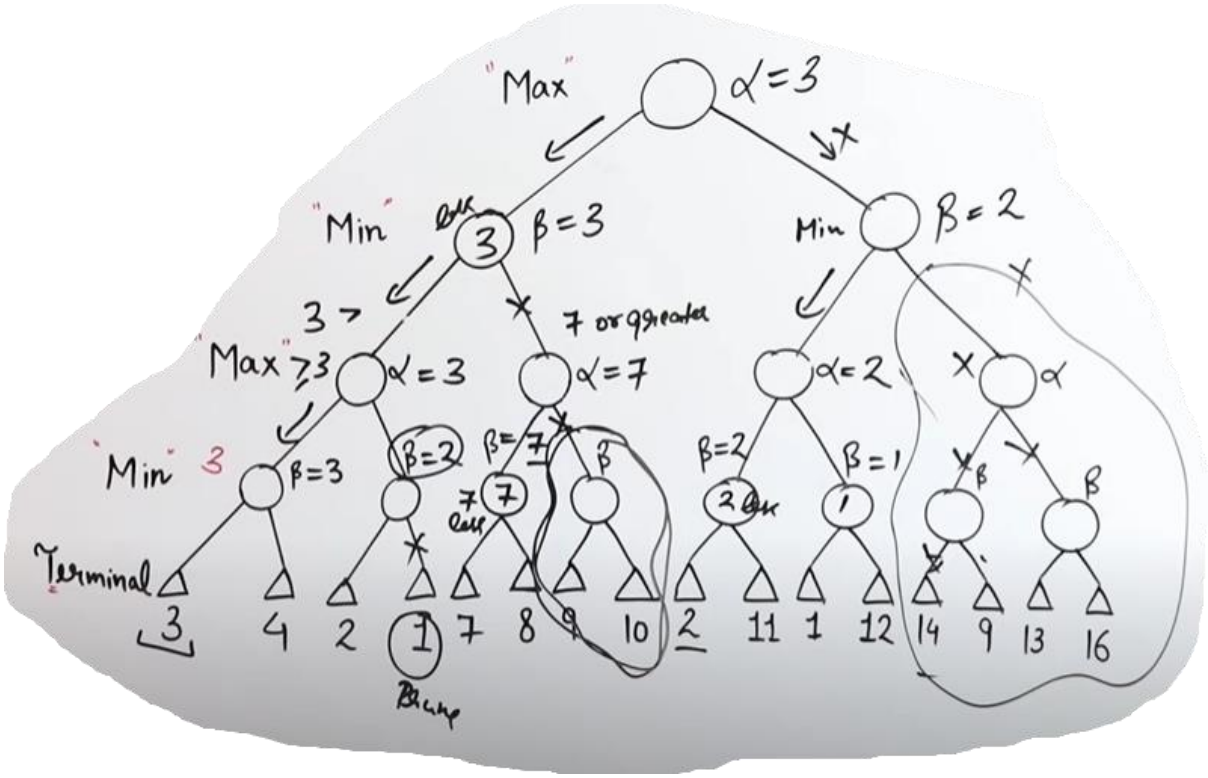
As you can see in the diagram next of game tree at the stages we are choosing the Max and min value alternatively and making the move accordingly.

Alpha Beta Pruning Algorithm: Alpha-Beta Algorithm is search algorithm that seeks to decrease the numbers of nodes that are evaluated by the Min-Max Algorithm in its search tree.it stops evaluating a move when at least one possibility has been found that proves move to be worse than an earlier move. In such cases moves needs not be evaluated further, thus reducing the numbers of nodes to be evaluated and this algorithm is faster than the Min-Max as the time complexity reduces in this one.

Time Complexity

O(b^(d/2)) Here b = number of legal moves at each point, m= depth of Tree.

In the diagram Below you can see how we are eliminating the nodes when we get the minimum and maximum values at the alternative stages, and we are just not visiting the nodes when it is not making sense as we already got the maximum and the minimum values.



Evaluation Function: In my evaluation Function I have given scores in such a way that if the opponent is able to occupy all the corners, then they will get the 4 points as the corners are the most important places in this game and if the opponent is able to occupy the sides then they will get the 2 points and for all other places other than corners and sides of the board they will get 1 point. And in the end whose score will be more that will be the winner of the game.

Analysis Function: Analysis is done according to the time taken tin completed the AI vs AI and time taken in completing the game is 109.57s

Time Taken: 109.57814478874207

Game Over!

Score Minimax: 38

Score Alpha Beta : 11

Bibliography

https://www.youtube.com/watch?v=Ntu8nNBL28o

https://www.youtube.com/watch?v=dEs_kbvu_0s

https://medium.com/javarevisited/alpha-beta-pruning-an-optimization-technique-for-the-minimax-algorithm-in-game-theory-of-6c850c0e462f

https://github.com/milorue/Min-Max-Reversi/tree/master/reversi