Assignment 6: Minimax Algorithm

Srihari K G 210030035

Tejas Ajit Mhaiskar 210030039

Description of Algorithm

a) Minimax Algorithm :

The minimax algorithm is a decision-making algorithm used in game theory and artificial intelligence. It is based on the concept of maximizing the gain and minimizing the loss. The algorithm works by assuming that the opponent will play optimally and selecting the move that leads to the best possible outcome for the player. In simpler terms, the algorithm works by exploring all possible moves that can be made by the player and the opponent and assigning a score to each move based on the likelihood of winning the game. The player then selects the move with the highest score, assuming the opponent will select the move with the lowest score. The algorithm recursively evaluates each possible move until a terminal state is reached, meaning that the game has ended. The final score is then used to select the best move for the player. The minimax algorithm is commonly used in games such as chess and tic-tac-toe, but it can also be used in decision-making problems where there are two opposing sides.

b) Alpha-Beta Pruning :

Alpha-beta pruning is a search optimization technique used in the minimax algorithm to reduce the number of nodes evaluated. It works by maintaining two values, alpha and beta, to represent the best scores found so far. When a node is visited, its score is compared with alpha and beta, and if it is worse than alpha or better than beta, the node and its subtree are pruned. This eliminates the need to evaluate all possible moves, making the algorithm more efficient and faster, without sacrificing accuracy.

Heuristic Functions :

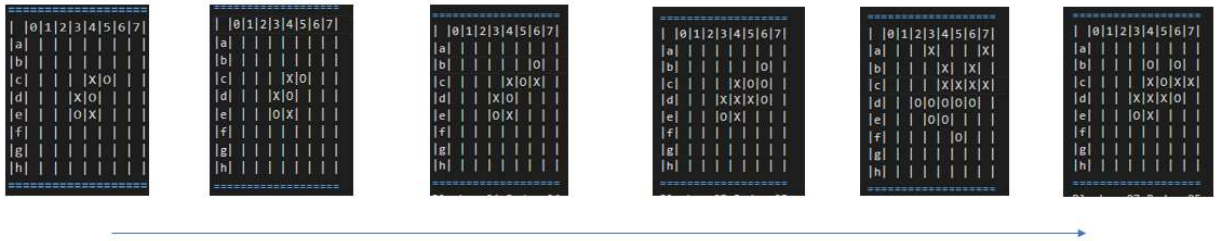
1) Number of Black Coins on board - Number of White Coins on board
2) Number of valid moves for Player 1 - Number of valid moves for Player2
3) Assigning weights where, corner coins are valued higher , and if the coins are on the edges of board there are added higher values for both type of coins and the heuristic is the difference between them.

TREES

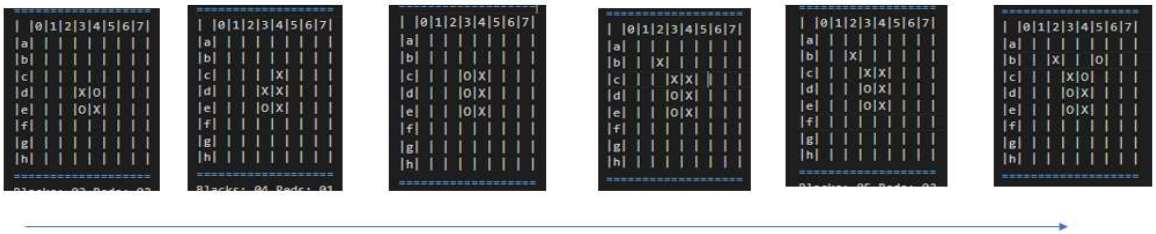
The following moves were recorded for game between various bots and some of them had Time out because decision time were taking more than what was restricted.

The following tree is described as player 1- > player 2- > player 1- > player 2- > player 1- > player 2 and so on.

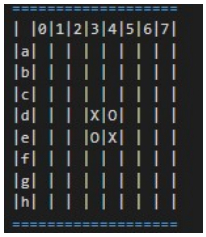
1. **AB bot vs Random bot**



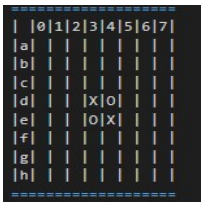
2. **AB vs Mini Max**



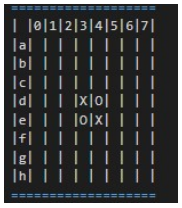
3. **Mini Max Vs Slow (TIMEOUT)**



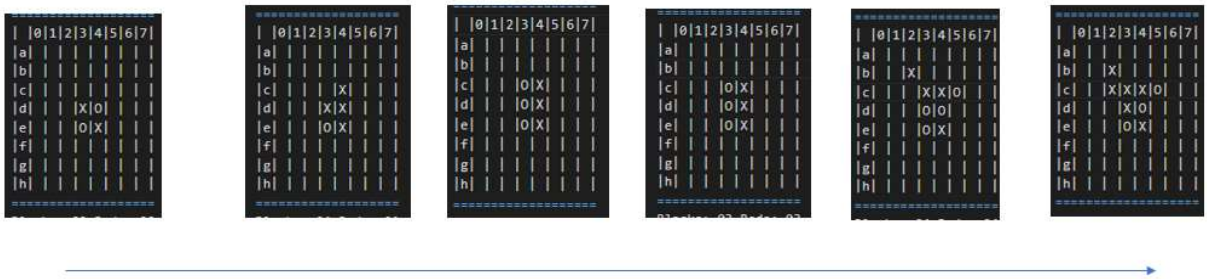
4. **Slow vs Slow (TIMEOUT)**



5. **Minimax vs Turing (TIMEOUT)**



6. **Minimax vs AB**



Comparison between two Algorithms:

a) Space and Time Complexity:

- Both the minimax algorithm and the minimax algorithm with alpha-beta pruning have the same worst-case time complexity of O(b^d), where b is the branching factor and d is the maximum depth of the search tree. However alpha beta pruning on an average has time complexity O(b^d/2).
- However, alpha-beta pruning can significantly reduce the average-case time complexity, as it can eliminate large portions of the search tree. In terms of space complexity, minimax with alpha-beta pruning requires slightly more memory than standard minimax, as it needs to store additional values for alpha and beta. However, the memory overhead is negligible in practice.
- Overall, alpha-beta pruning is a powerful optimization that can greatly improve the efficiency of the minimax algorithm, especially for games with a large search space. It reduces the search time by eliminating unnecessary branches in the search tree, making it a more efficient algorithm than standard minimax.

b) Winning Criteria:

- The winning criteria for both minimax and minimax with alpha-beta pruning are the same. The algorithms aim to find the best possible move for the player by maximizing their chances of winning and minimizing their chances of losing.

- The minimax algorithm evaluates all possible moves and selects the move that leads to the best possible outcome for the player, assuming the opponent plays optimally.

- Minimax with alpha-beta pruning also aims to find the best possible move for the player, but it eliminates unnecessary branches in the search tree, making it faster and more efficient. This allows it to search deeper into the tree, which can result in finding better moves and increasing the player's chances of winning.

In summary, both algorithms have the same winning criteria, but alpha-beta pruning can help the player find the best move more efficiently.

On an average 50 moves happen before the result.

Observations :

Alpha beta wins always against Minimax bot , in about 50 moves.