Group no: 31

CS6380: Artificial Intelligence (Assignment 2)

Chiranjeevi N (CS17M063)

Rajib Das Bhagat (CS17M034)

1. Problem Statement

To code a bot to play the game of Othello, and win.

2. Description

Othello, most commonly known as reversi is a two players strategy board game. The game is played on an 8 X 8 unchecked board. There are sixty-four identical discs, which are light on one side and dark on the other. Players take their turns placing disks on the board with their assigned color facing up. During a play, any disks of the opponent's color that are in a straight line such as horizontally, vertically or diagonally are bounded by the disk just placed and another disk of the current player's color are turned over to the current player's color. The object of the game is to have the majority of disks turned to display your color when the last playable empty square is filled. In this assignment, we code our bot to win the game of Othello. Given a board configuration and a turn, our bot returns a valid move. The game ends when neither of the players can make a valid move. The player with maximum number of coins is the winner.

3. Algorithm

Here, we have implemented the Othello bot playing assignment using MiniMax algorithm. The algorithm searches in a depth first search manner to the depth of four. The algorithm retuns the minimax value, but not the best move. The best move algorithm is also depicted which keeps tracks of all the best moves and best value present in the board while playing. The best move algorithm accepts a board position and retuns the best move for MAX turn.

Algorithm 1 Minimax algorithm

procedure MINIMAX(j):
If Terminal V(j):
then return $V(j) \leftarrow e(j)$
else for $i \leftarrow 1$ to b
do
Generate j(i) the i^{th} successor of j
$V(j[i]) \leftarrow Minimax(j[i])$
If i = 1
then $CV(j) \leftarrow V(j[i])$
else if j = MAX
then $CV(j) \leftarrow Max(CV(j), V(j[i]))$
else $CV(j) \leftarrow Min(CV(j), V(j[i]))$
return $V(j) \leftarrow CV(j)$

Algorithm 2 BestMove algorithm

```
procedure BestMove(j):
b ← NIL
value ← −INFINITY
for i ← 1 to b
        do V(j[i]) ← Minimax(j[i])
        if V(j[i]) > value
                then value ← V(j[i])
                        b ← (j[i])
return b
```

4. Complexity

The time complexity of minimax is O(bm) which is the sum of all the levels and the space complexity is O(bm); where b is the branching factor.