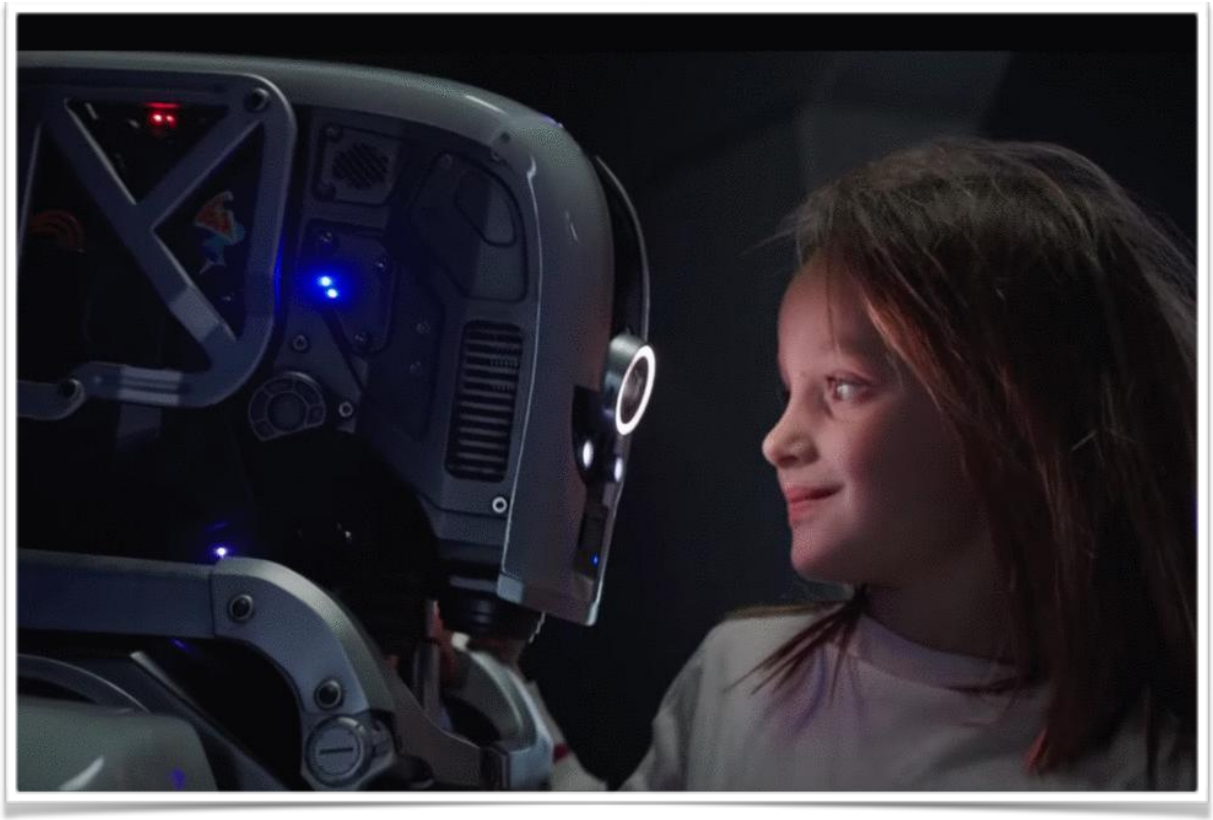


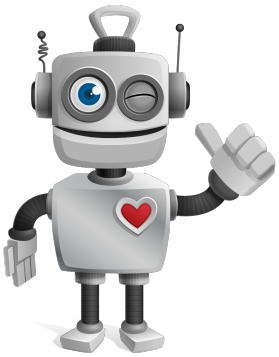
Human Intelligence VS Artificial Intelligence

CMPT-310 Project Report

Raman Sehmbi Student ID: 301326893

Sidharth Miglani Student ID: 301326138

Introduction

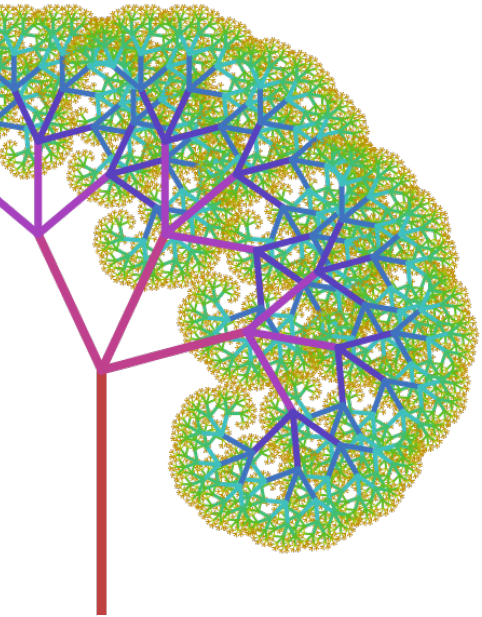


Artificial Intelligence is the science of getting machine to think and make decisions like humans do. Playing board games also require thinking and making decisions for the best possible move. In this project, we implemented Monte Carlo Search and weighted heuristic algorithm for the game called Reversi.

Monte Carlo Search

Monte Carlo Search is an algorithm which picks the best possible move from the set of available nodes using : Selecting, Expanding, Simulating and Updating.

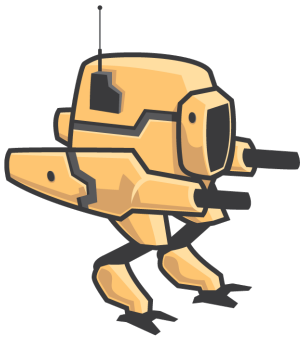
In our project, we implemented these four steps for decision making process. We have a node and tree class. The node is searched from current tree state, and the one with the highest possibility of winning is selected. The decision is calculated using the formula $w_i/s_i + c\sqrt{(ln_{sp}/s_i)}$. After expansion, we select the one with highest number of wins and update the game state.



Weighted Search

This algorithm is assigning weights to the play-outs. This algorithm plays the complete game and assigns low score to the play-out which yields a win and high negative score to the payout which looses.

The result is list of scores, and the one with the highest score is selected and the state is changed according to the move associated with the highest score. Note, with different scores assigned there will only be one state with max score.



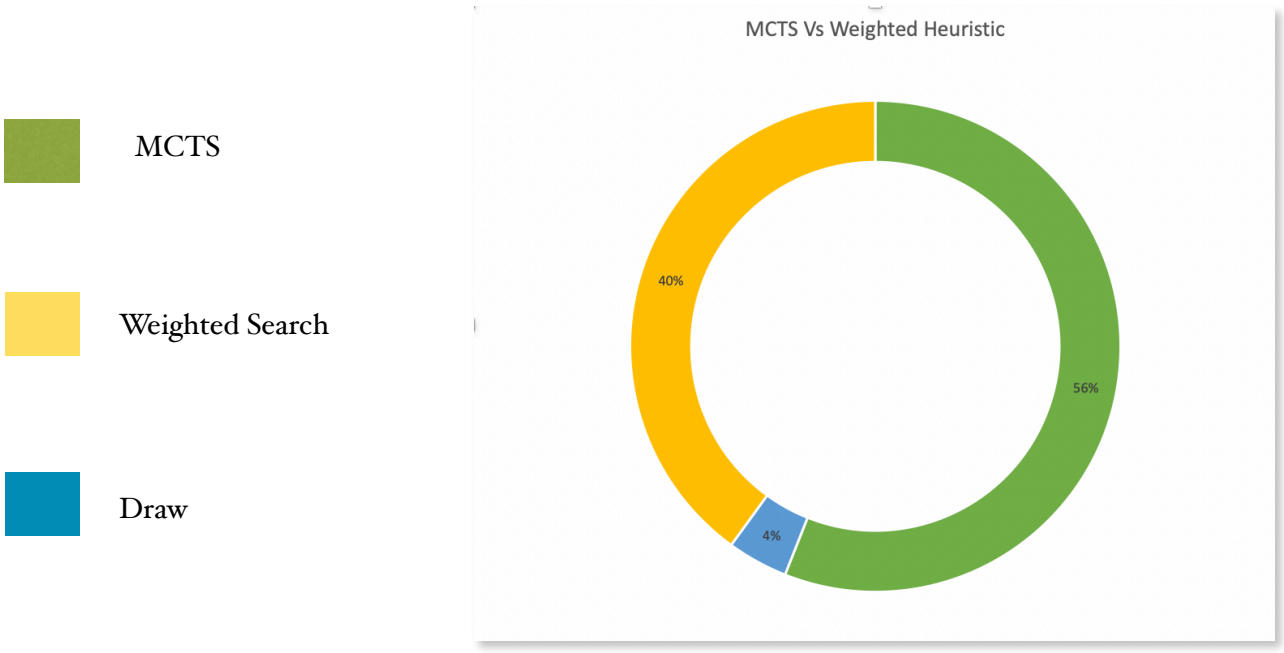
Key Differences between algorithms:

1. MCTS uses time to iterate while Weighted Search uses loop.
2. Weighted Search (WS) has negative weights attached whereas MCTS uses Upper Confidence Bound.
3. MCTS stops as soon as the time is up while WS keeps iteration for 800 loops for the play-outs.

In our project, these two algorithms plays against each other. MCTS tries to beat Weighted search and vice-versa.

Let's explore the results from the play-outs:

Out of the 100 play-outs we had, MCTS won 56% of the time where as weighted search won 40% of the time.



Comparing Running Time:

One of the interesting things to note here is the time taken by each algorithm to make a move.

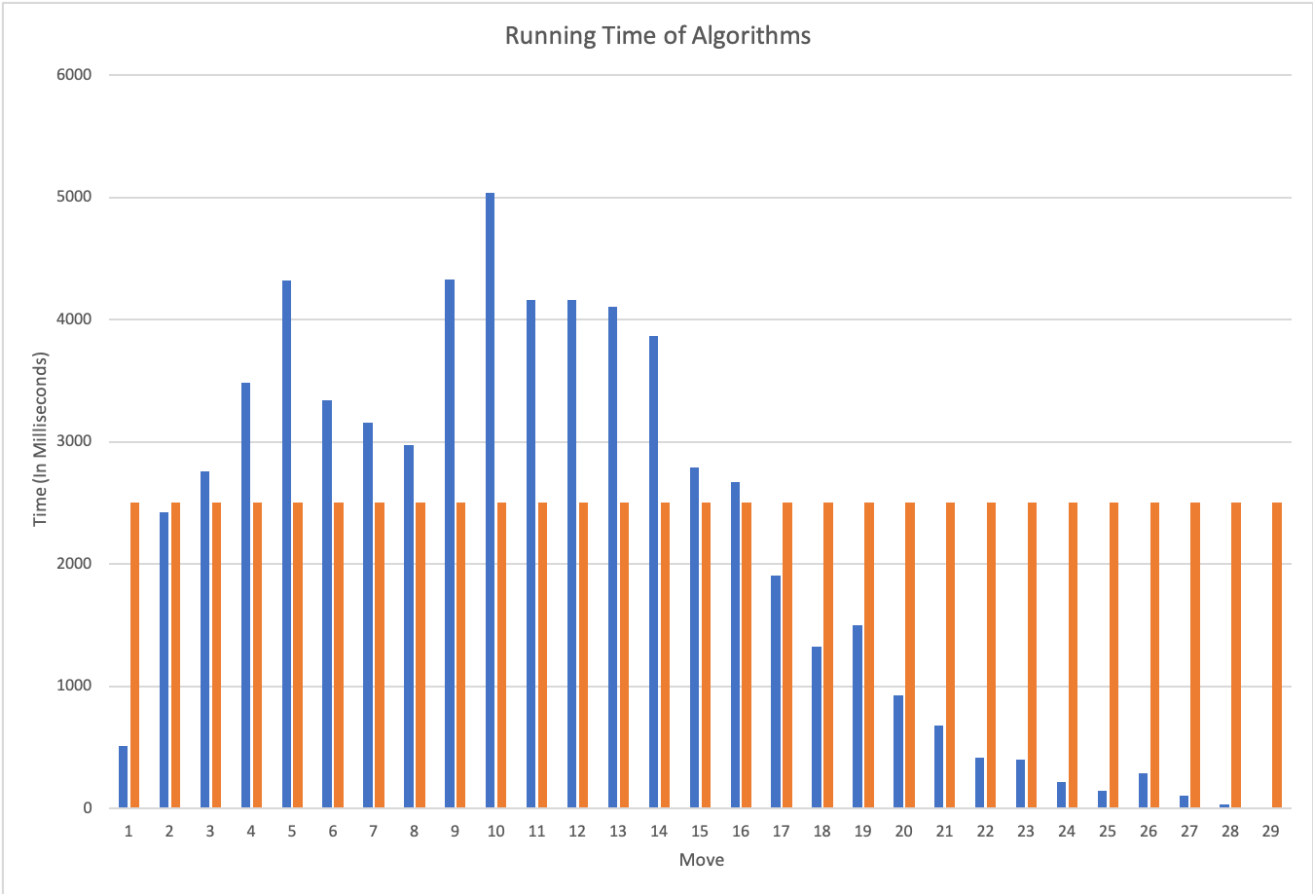
It's important since we want an algorithm which gives us fast and better results. Waiting too long and for a better move for the algorithms in today's world would be of less importance than making a slightly less better move in less time.

```
long mstartTime = System.currentTimeMillis();
reversiBoard = monteCarloSearch.findNextMove(reversiBoard,
long mendTime = System.currentTimeMillis();
long mtimeElapsed = mendTime - mstartTime;
MCTSTime.add(mtimeElapsed);
```

```
long startTime = System.currentTimeMillis();
WeightHeuristic wH = new WeightHeuristic();
reversiBoard = wH.findNextMove(reversiBoard, playerNo: 2);
long endTime = System.currentTimeMillis();
long timeElapsed = endTime - startTime;
WHeurisctics.add(timeElapsed);
```

Weighted Search

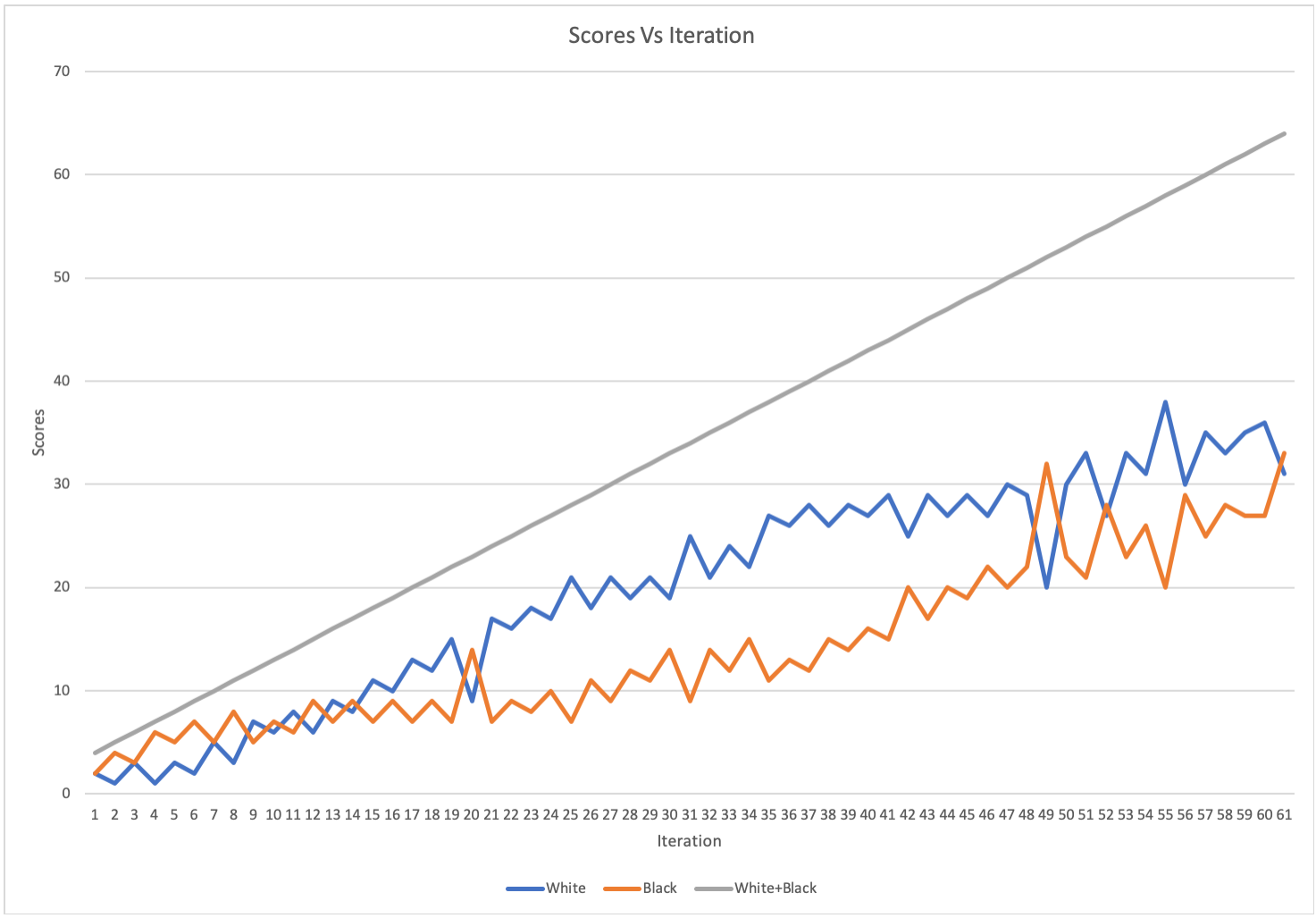
MCTS



The running of MCTS is constant as it is timed to 2500 milli-seconds for each iteration, where as running time of weighted search is variable as it depends on available states and possible moves.

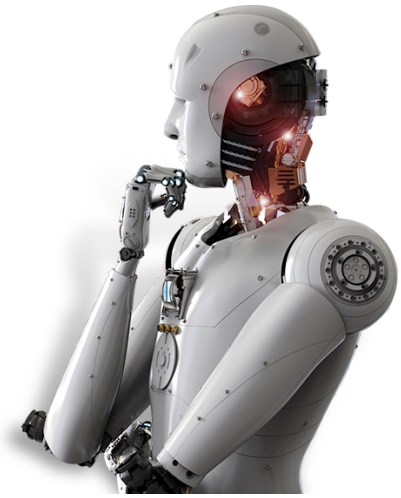
Comparing Scores of Iteration

The graph below demonstrates the score of moves made by Blue(Weight Heuristic) and Orange (MCTS). One interesting thing about the graph is that till last turn, most of time the scores were in favour of white(Blue on graph) team. If we compare iteration 20 to 47, white team clearly looks like a winner but in the end, black wins. Since the average wins of black team are more than the average wins of the white team, it is evident that MCTS is playing the game more strategically.



Who is Better ?

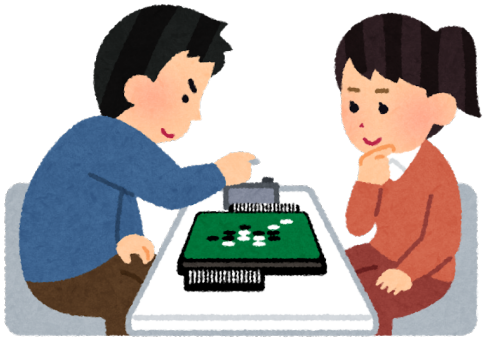
The running time of MCTS is constant, that means for every move it will take 2500 milliseconds, where as for Weighted Search time decreases as less positions on the Reversi board were available. Also, the scores were in favour of Weight Heuristic most of the times, considering this, one can easily get fooled that weighted Search is better which is clearly not the case.



Also,

The average win rate of MCTS is more than Weighted search but the total running time is more. One has to make sacrifice of one of the feature in order to choose algorithm from the given two algorithms.

Project Structure for Reversi



The implementation of Reversi and algorithms are done in Java for this project.

The project has 3 major java classes:

Main.java to select if you want to play against AI or want to see how the 2 algorithms compete against each other.

ManualHeuristic.java class which contains the all the elements for weighted Search algorithm.

MonteCarloSearch.java which contains elements for Monte Carlo Tree Search.

You can either import the project in **IntelliJ Idea** and just click the play button in order to run the program.

Or,

You can also run the project using the **Command Line**. Navigate to project source code. The source code of this project is in the directory:

Project/src/"all the source files"

In order to compile all the java code.

Navigate to project source code directory using

cd [path]/Project/src

Now, you will be in src folder

Use javac *.java

The above command will compile all the java code.

Just run: java Main

And the program will run.

Project Screenshots and how to run it.

Once you ran **java Main** command, you will see the following screen on the terminal. Enter the numbers 1,2 or 0 to choose from options.

If you enter wrong number, the program will ask you to enter the correct input again.

```
****************** Welcome to Reversi ******************


~ For Playing with the computer enter 1 ~
~ For Algorithms to play with each other enter 2 ~
~ To Exit, press 0 ~

Enter your Input:
```

```
  A B C D E F G H
1 _ _ _ _ _ _ _ _
2 _ _ _ _ _ _ _ _
3 _ _ _ * _ _ _ _
4 _ _ * W B _ _ _
5 _ _ _ B W * _ _
6 _ _ _ _ * _ _ _
7 _ _ _ _ _ _ _ _
8 _ _ _ _ _ _ _ _

White: 1 Black: 4

  A B C D E F G H
1 _ _ _ _ _ _ _ _
2 _ _ _ _ _ _ _ _
3 _ _ * _ * _ _ _
4 _ _ B B B _ _ _
5 _ _ * B W _ _ _
6 _ _ _ _ _ _ _ _
7 _ _ _ _ _ _ _ _
8 _ _ _ _ _ _ _ _

Place User Move White:
E3
```

If you enter 1,

You will see the screenshots shown on the left. * represents empty valid locations and W represents White team, B represents Black team. In our case,

B = Computer

W = Human

and Computer takes the first turn!

Enter the Input in the form of E3, C3 or C5 to place your symbol on the board.

If you enter 2,

The algorithms will play with each other and you can see them playing on the screen. Once they are finished, the result will be printed on the screen along with the running times.

You can compare the outputs shown on the report with the Command Line outputs in order to compare them.

```
  A B C D E F G H
1 B B B B B B B W
2 B B B B B B W W
3 B W W B B W B W
4 B B B B W W B W
5 B B W W W B B W
6 B W W W B W B W
7 B W W B B B B W
8 W W W W B B B W


White: 27 Black: 37
Black Wins


MCTS Time (in Milliseconds)
2501 2500 2500 2500 2500 2500 2500 2500 2500 2500 2500

Weight Heuristic Time (in Milliseconds)
1530 2880 3257 2286 893 4673 1632 2323 2635 3209 3465

~ For Playing with the computer enter 1 ~
~ For Algorithms to play with each other enter 2 ~
~ To Exit, press 0 ~

Enter your Input:
```