Heuristic Analysis

By: Anamika Sharma, Udacity AI Nanodegree, Term 1 (10 Nov 2017)

Introduction

We are developing an AI agent to solve the game of isolation. Isolation is a board game played by 2 players wherein each player take their turns to move their piece one after another. Location of previous moves will be blocked for all future moves. Another restriction is, player should move his piece in L-shaped steps only. So, the first player who runs out of available moves loses the game.

Things to be implemented:

1. Minmax algorithm for searching the game tree.
2. Alphabeta pruning to increase the efficiency of minmax game tree search.
3. Iterative deepening is used with alphabeta search to get next best move within given span of time.
4. Design heuristic evaluation functions to compare the outcomes of the game.

Heuristic Evaluation Function

Heuristic evaluation functions are used by AI game playing agents to estimate the value or goodness of the position in minmax and related algorithms. Here, I have used three different heuristics to evaluate the performance:

1. Penalizing the player to move to corner positions.
2. Player has moved his piece far away from the opponent's location.
3. Determine the further legal moves after execution of the current legal move.

Heuristic One: It calculates the difference between legal moves of player and opponent while penalizing the player when its current location is the corner of board. This is done because corners decrease the no. of legal moves and henceforth decreases the winning chances.

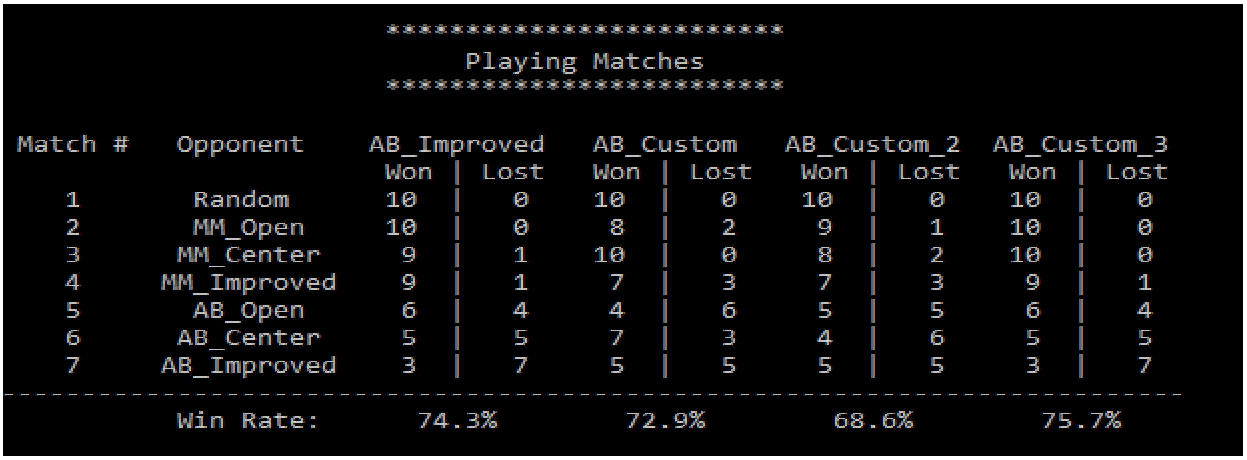
Heuristic Two: First, we will calculate Euclidian distance between player and opponent's current location. Now, we will add this to legal moves of player. After that we will calculate difference between legal moves of player and opponent. If the player moves at greater distance from opponent's location can get a higher score.

Heuristic Three: First we will calculate the available legal moves of both player and opponent after executing current legal move. Then we will calculate difference between legal moves of player and opponent. Basically, we have access to future legal moves in advance and chances of win can be improved.

Results

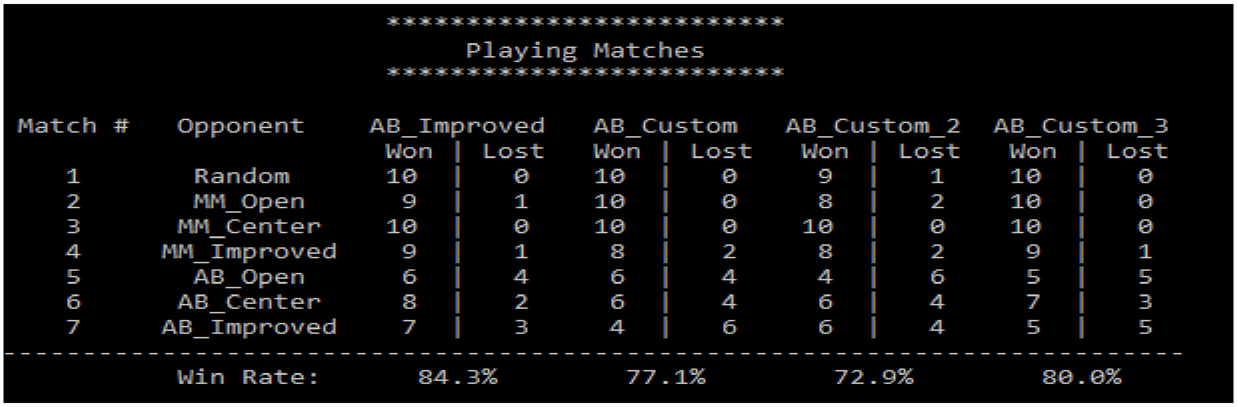
The performance of AI agent for Isolation game has been determined by tournament.py file. This script evaluates the performance of the custom_score evaluation function against a baseline agent using alpha-beta search and iterative deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use ID and alpha-beta search with the custom_score functions defined in game_agent.py. It displays outcomes for all the heuristic evaluation functions implemented by us.

Outcome 1

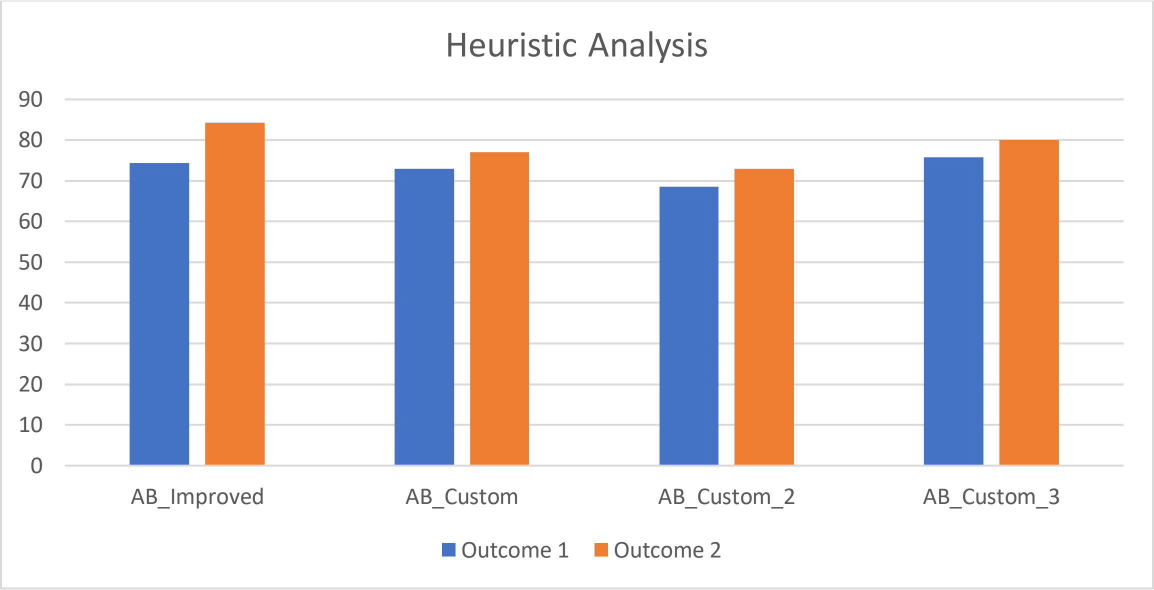


```
                        *************************
                             Playing Matches
                        *************************

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      10  |   0    10  |   0    10  |   0    10  |   0
    2       MM_Open     10  |   0     8  |   2     9  |   1    10  |   0
    3      MM_Center     9  |   1    10  |   0     8  |   2    10  |   0
    4     MM_Improved    9  |   1     7  |   3     7  |   3     9  |   1
    5       AB_Open      6  |   4     4  |   6     5  |   5     6  |   4
    6      AB_Center     5  |   5     7  |   3     4  |   6     5  |   5
    7     AB_Improved    3  |   7     5  |   5     5  |   5     3  |   7
--------------------------------------------------------------------------
           Win Rate:      74.3%        72.9%        68.6%        75.7%
```

Outcome 2



```
                        *************************
                             Playing Matches
                        *************************

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      10  |   0    10  |   0     9  |   1    10  |   0
    2       MM_Open      9  |   1    10  |   0     8  |   2    10  |   0
    3      MM_Center    10  |   0    10  |   0    10  |   0    10  |   0
    4     MM_Improved    9  |   1     8  |   2     8  |   2     9  |   1
    5       AB_Open      6  |   4     6  |   4     4  |   6     5  |   5
    6      AB_Center     8  |   2     6  |   4     6  |   4     7  |   3
    7     AB_Improved    7  |   3     4  |   6     6  |   4     5  |   5
--------------------------------------------------------------------------
           Win Rate:      84.3%        77.1%        72.9%        80.0%
```



Recommendation

I will choose heuristic approach third because:

1. As per outcome percentage, it performs better.
2. It is simple and does not affect the ability of algorithm to search deep.
3. It considers the future legal moves once current move has been executed and it tends to increase the winning chance.