Artificial Intelligence Nanodegree

Assignment 2: Build A Game Playing Agent

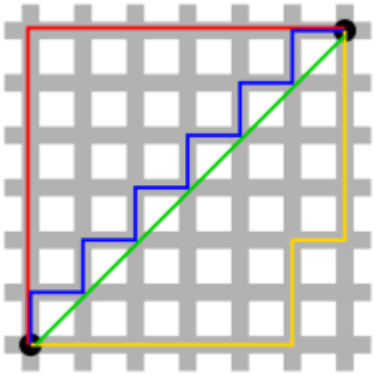
Ollie Graham
25th September, 2017

Heuristic Analysis

Heuristic 1: Manhattan Distance

This heuristic attempts to maximize the moves which keep the agent closer to the center of the board than its opponent.

The Manhattan distance refers to the grid like structure of Manhattan Island, New York City and is shown by the blue line the in the following diagram. It is the shortest distance between the two points, whilst adhering to the grid structure (the green line is the Euclidean distance and is the shortest distance between the points):



The intention of this heuristic is to gain an advantage over the computer agent by valuing moves that keep the player agent near the center of the board, away from the edge of the board where number of available moves are more limited.

Here are the results of the tournament using this heuristic:

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

Match #   Opponent    AB_Improved   AB_Custom_2
                      Won | Lost    Won | Lost
   1       Random      9  |   1     9  |   1
   2       MM_Open     8  |   2     7  |   3
   3      MM_Center   10  |   0    10  |   0
   4     MM_Improved   8  |   2     7  |   3
   5       AB_Open     6  |   4     3  |   7
   6      AB_Center    5  |   5     7  |   3
   7     AB_Improved   6  |   4     4  |   6
--------------------------------------------
          Win Rate:      74.0%        67.0%
```

By using this heuristic the agent performed well against the MM player, but did not perform as well against the AB player. It's strongest performance against both the MM and AB players was against the _Center heuristic which was expected due to the pattern of play that this heuristic was attempting to maximize.

The agent also did not outperform the AB_Improved agent which used the simple own moves – opponent moves heuristic.

Heuristic 2: Weighted Moves

This heuristic builds on the performance of the AB_improved heuristic by weighting both the agent and opponent moves.

The AB_improved heuristic is:

$$own\ moves - opponent\ moves$$

The weighted moves heuristic I applied here is:

$$(w * own\ moves) - ((1 - w) *\ opponent\ moves)$$

where:

- own moves = number of remaining legal moves for agent
- opponent moves = number of remaining legal moves for opponent
- w = weight

I referred to the work done by fellow Udacity student Ryan Shrott in training a genetic algorithm to converge on the optimal weight to use for this heuristic. Details can be found at this link.

Ryan determined that the optimal weight to use for this heuristic is 1/e, or 0.3678.

The heuristic gives more weight to the number of moves the opponent player has remaining. This causes the agent to value moves which reduce the number of remaining moves for the opponent more highly, which should be beneficial in early game states when its not possible to search the whole tree in the allotted time.

Here are the results of tournament play with this heuristic:

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

Match #   Opponent    AB_Improved  AB_Custom_3
                       Won | Lost   Won | Lost
   1       Random       9  |   1    10  |   0
   2       MM_Open      8  |   2     8  |   2
   3      MM_Center    10  |   0     6  |   4
   4     MM_Improved    8  |   2     7  |   3
   5       AB_Open      6  |   4     4  |   6
   6      AB_Center     5  |   5     4  |   6
   7     AB_Improved    6  |   4     4  |   6
----------------------------------------------
          Win Rate:      74.0%       61.0%
```

Although this heuristic performed reasonably well against the MM player, it lost 60% of its matches against the AB player regardless of the heuristic used.

Overall the agent still won 61% of the matches played, but it didn't outperform the more simple AB_Improved heuristic. In this case the 'simpler' heuristic performed better than the weighted heuristic likely due to the weighted heuristic adding no more information about the best move in any state than simply by having more moves available.

Perhaps playing a larger number of games would cause the winning percentages to be closer for these metrics and is something I would like to investigate with more time to run the games.

Heuristic 3: Dual Heuristic

In this heuristic I combined two heuristics which are applied according to how far the game has progressed.

A simple rule checks the amount of free spaces remaining on the game board. If there is >= 30% of the spaces remaining the following heuristic is applied:

$$own\ moves - 2\ *\ opponent\ moves$$

If <30% of spaces remain then the AB_improved heuristic is applied instead:

$$own\ moves - opponent\ moves$$

Here are the results of play using this heuristic:

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

Match #   Opponent    AB_Improved   AB_Custom
                       Won | Lost   Won | Lost
   1       Random       9  |   1    10  |   0
   2       MM_Open      8  |   2     9  |   1
   3      MM_Center    10  |   0     8  |   2
   4     MM_Improved    8  |   2    10  |   0
   5       AB_Open      6  |   4     5  |   5
   6      AB_Center     5  |   5     7  |   3
   7     AB_Improved    6  |   4     5  |   5
-----------------------------------------------
          Win Rate:      74.0%        77.0%
```

The heuristic seeks to apply moves which greatly reduce the opponent's moves early in the game when the depth of the search tree is large (as implied by subtracting a multiple of the opponents moves from the agent's moves) and reverts to the more simple 'improved' heuristic later in the game when favoring the move which leaves the agent with more moves than the opponent towards the end game.

The results show that the heuristic outperforms the single AB_Improved heuristic on its own, although again, I would like to see if this relationship holds over a larger number of trials given more time.

I recommend that this heuristic be used by the agent because:

1. Its win rate was well above the 60% threshold considered as acceptable for this assignment
2. This heuristic outperformed the other heuristics I tested and the simple but powerful AB_Improved heuristic in my trials
3. It's still a relatively simple heuristic which doesn't limit being able to search deep into the search tree.