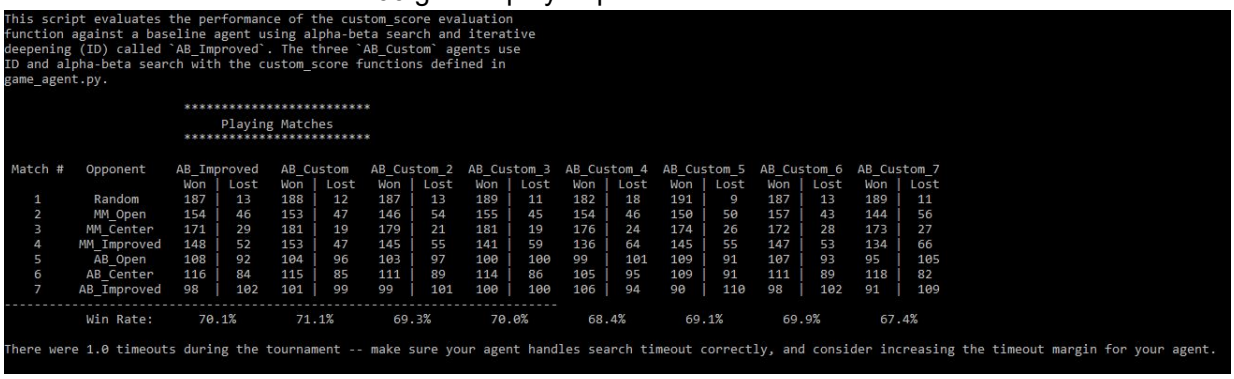
Heuristic Analysis

Below is the overall result of 200 games played per match



```
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.

                    ************************
                         Playing Matches
                    ************************

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3  AB_Custom_4  AB_Custom_5  AB_Custom_6  AB_Custom_7
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost   Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      187 |  13    188 |  12    187 |  13    189 |  11    182 |  18    191 |   9    187 |  13    189 |  11
    2       MM_Open     154 |  46    153 |  47    146 |  54    155 |  45    154 |  46    150 |  50    157 |  43    144 |  56
    3      MM_Center    171 |  29    181 |  19    179 |  21    181 |  19    176 |  24    174 |  26    172 |  28    173 |  27
    4     MM_Improved   148 |  52    153 |  47    145 |  55    141 |  59    136 |  64    145 |  55    147 |  53    134 |  66
    5       AB_Open     108 |  92    104 |  96    103 |  97    100 |  100   99  |  101   109 |  91    107 |  93    95  |  105
    6      AB_Center    116 |  84    115 |  85    111 |  89    114 |  86    105 |  95    109 |  91    111 |  89    118 |  82
    7     AB_Improved   98  |  102   101 |  99    99  |  101   100 |  100   106 |  94    90  |  110   98  |  102   91  |  109
--------------------------------------------------------------------------------
          Win Rate:      70.1%        71.1%        69.3%        70.0%        68.4%        69.1%        69.9%        67.4%

There were 1.0 timeouts during the tournament -- make sure your agent handles search timeout correctly, and consider increasing the timeout margin for your agent.
```

Improved Alpha-Beta Pruning algorithm

The algorith used as a basis of comparison

Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	93.5	(187, 13)
MM_Open	77.0	(154, 46)
MM_Center	85.5	(171, 29)
MM_Improved	74.0	(148, 52)
AB_Open	54.0	(108, 92)
AB_Center	58.0	(116, 84)
AB_Improved	49.0	(98, 102)

`Total Win Ratio (%): 70.14`

Heuristic 1: Difference between each players move, decaying aggressivity, prioratizing center position

This heuristic prioratizes moves that leads to higher number of player moves with lower number of opponents moves. It starts with a higher aggressive focus on opponents moves that decays as the board fills up. It also prioratizes moves closer to the center.

The heuristic formula:

dist = distance between player move and center from 0 being center to 1 being a corner

decay = # of current blank spaces / # number of total blank spaces

(# of player moves - # of opponents moves * 2 * decay) * 1/dist

Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	94.0	(188, 12)
MM_Open	76.5	(153, 47)
MM_Center	90.5	(181, 19)
MM_Improved	76.5	(153, 47)
AB_Open	52.0	(104, 96)
AB_Center	57.5	(115, 85)
AB_Improved	50.5	(101, 99)

```
Total Win Ratio (%): 71.07
```

Heuristic 2: Difference between each players move with an aggressive approach

This heuristic prioratizes moves that leads to higher number of player moves with lower number of opponents moves while putting more value on lower number of opponents moves.

The heuristic formula:

# of player moves - # of opponents moves * 2

Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	93.5	(187, 13)
MM_Open	73.0	(146, 54)
MM_Center	89.5	(179, 21)
MM_Improved	72.5	(145, 55)
AB_Open	51.5	(103, 97)
AB_Center	55.5	(111, 89)
AB_Improved	49.5	(99, 101)

```
Total Win Ratio (%): 69.29
```

Heuristic 3: Difference between each players move

This heuristic prioratizes moves that leads to higher number of player moves with lower number of opponents moves.

The heuristic formula:

# of player moves - # of opponents moves

Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	94.5	(189, 11)
MM_Open	77.5	(155, 45)
MM_Center	90.5	(181, 19)
MM_Improved	70.5	(141, 59)
AB_Open	50.0	(100, 100)
AB_Center	57.0	(114, 86)
AB_Improved	50.0	(100, 100)

```
Total Win Ratio (%): 70.0
```

Heuristic 6: Difference between each players move, decaying aggressivity, prioratizing outer position

This heuristic is the same as heuristic 1 except that it prioratizes position further from the center.

The heuristic formula:

dist = distance between player move and center from 0 being center to 1 being a corner

decay = # of current blank spaces / # number of total blank spaces

(# of player moves - # of opponents moves * 2 * decay) * dist

Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	93.5	(187, 13)
MM_Open	78.5	(157, 43)
MM_Center	86.0	(172, 28)
MM_Improved	73.5	(147, 53)
AB_Open	53.5	(107, 93)
AB_Center	55.5	(111, 89)
AB_Improved	49.0	(98, 102)

```
Total Win Ratio (%): 69.93
```

Heuristic 4,5,7: Different Variance of the distance factor

These heuristics are different variants of heuristic 1 and 6. They also use a distance factor that either values center or outer positions. However, these are applied to either the player or the opponent. I was curious to see whether it would drastically affect the outcome.

The heuristic 4 formula:

dist = distance between player move and center from 0 being center to 1 being a corner

decay = # of current blank spaces / # number of total blank spaces

(# of player moves * 1/dist - # of opponents moves * 2 * decay)

The heuristic 5 formula:

dist = distance between player move and center from 0 being center to 1 being a corner

decay = # of current blank spaces / # number of total blank spaces

(# of player moves * dist - # of opponents moves * 2 * decay)

The heuristic 7 formula:

dist = distance between player move and center from 0 being center to 1 being a corner

decay = # of current blank spaces / # number of total blank spaces

(# of player moves - # of opponents moves * 2 * decay * 1/dist)

Heuristic 4 Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	91.0	(182, 18)
MM_Open	77.0	(154, 46)
MM_Center	88.0	(176, 24)
MM_Improved	68.0	(136, 64)
AB_Open	49.5	(99, 101)
AB_Center	52.5	(105, 95)
AB_Improved	53.0	(106, 94)

```
Total Win Ratio (%): 68.43
```

Heuristic 5 Results:

	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	95.5	(191, 9)
MM_Open	75.0	(150, 50)
MM_Center	87.0	(174, 26)
MM_Improved	72.5	(145, 55)
AB_Open	54.5	(109, 91)
AB_Center	54.5	(109, 91)
AB_Improved	45.0	(90, 110)

```
Total Win Ratio (%): 69.14
```

Heuristic 7 Results:

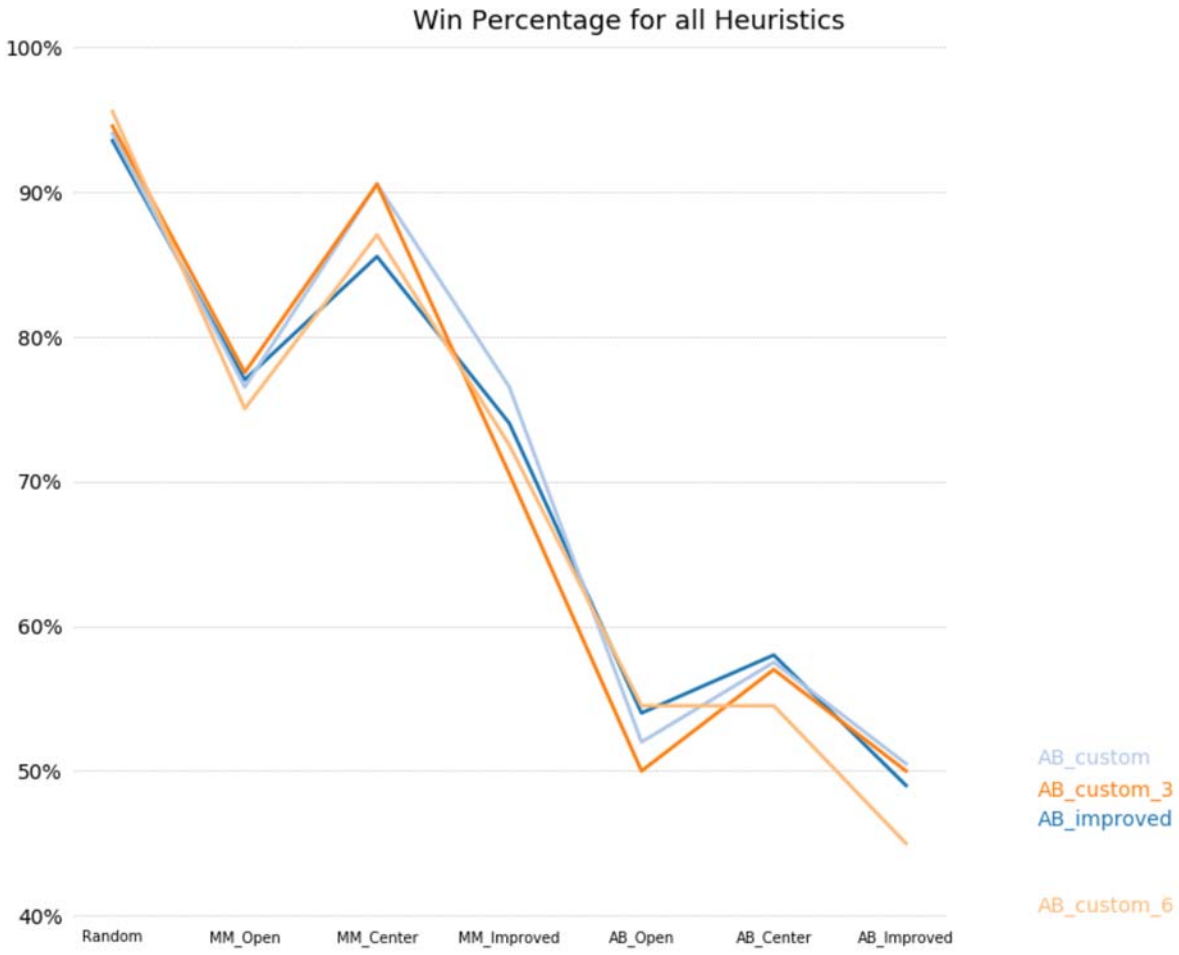
	AB_Improved Win Ratio (%):	AB_Improved Wins/Losses:
Random	94.5	(189, 11)
MM_Open	72.0	(144, 56)
MM_Center	86.5	(173, 27)
MM_Improved	67.0	(134, 66)
AB_Open	47.5	(95, 105)
AB_Center	59.0	(118, 82)
AB_Improved	45.5	(91, 109)

```
Total Win Ratio (%): 67.43
```

Conclusion:

Overall the results are fairly close to each other. A variant of Heuristic 3 (difference between number of moves) seem to be preferable. A decaying aggressivity by having the number of opponent moves being more valuable at the beginning and less valuable towards the end seem to also help. However, a distance factor, as currently implemented doesn't improve the results significantly.

The graph below show the results of the three best heuristics and shows that the results are very close and it is hard to tell which is better.


Win Percentage for all Heuristics
100%
90%
80%
70%
60%
50%
40%
Random
MM_Open
MM_Center
MM_Improved
AB_Open
AB_Center
AB_Improved
AB_custom
AB_custom_3
AB_improved
AB_custom_6

In [ ]: