Jonathan S Kim
KJ1646@gmail.com
AI Nanodegree - Udacity

Heuristic Analysis

Background

The purpose of this project was to come up with three heuristics for a variation of the game isolation, the variation being that players could only move their pieces in a L-shape motion, like knights in chess. The three chosen heuristics have a common theme: count the number of possible agent moves subtracted by the number of my possible opponents moves, which is the heuristic used by the MM/AB Improved Score heuristic. The heuristics chosen, however, takes this to another level by adding one extra step.

The theme was incorporated because while in development it was observed that the MM/AB Improved Score heuristic performed much better than any novel method. This may be due to a lack of imagination, experience or a combination of both, but the thought process was to not break what is working and to just improve on it.

There are two figures below. All data used to populate said figures were produced by tournaments running on a Windows 10 machine with i7-7500U CPU (2-cores), 16 GBs of RAM, with at most 2 tournaments running simultaneously in separate processes.

Heuristic 1 – Custom Score 1

The best way to describe this heuristic is the number of the possible moves the agent can perform subtracted by the number of the possible moves the opponent can perform n-ply ahead.

To be specific, for the first 80% of the game—determined by the number of blank spaces on the board—this heuristic tries all available moves and then tries all the available moves 1-ply ahead, forming a tree with a depth of max 3. The heuristic does this for the agent's perspective and the opponents, counting the number of the agent own respective moves and subtracting the number of opponents moves, which are the nodes of the search tree.

For the last 20% of the game this heuristic takes the same idea, but as far as it can. Instead of just going 1-ply ahead, it performs a depth first search and finds all possible leaf nodes at varies depths. And again, like the prior, it sums up the agents moves and sums up its opponents and subtracts the two totals for the score.

The rationale behind this heuristic is that counting the number of immediate available moves works relatively well. Therefore, knowing the number of moves 1-ply ahead shouldn't hurt, especially if the complexity of doing so isn't too terrible in practice. This rational for the last 20% is the same, but with the added assumption that the branching factor is low enough by this point of the game, that the overall complexity would remain within a reasonable margin.

The result is a **79.29%** win rate averaged over 10 tournaments. Look at figure 1.

Heuristic 2 – Custom Score 2

This heuristic just takes what Custom Score 1 during the first 80% of the game, which was to count the number of moves 1-ply ahead. Custom Score 1 was an incremental change of Custom Score 2 with the hopes of improvement.

The result is a **75.57%** win rate averaged over 10 tournaments.

Heuristic 3 – Custom Score 3

Custom Score 3 takes the idea of the improved score heuristic and a variant of the center score heuristic and combines them. To be specific, takes the difference of available moves between the two players and subtracts 25% of the euclidean distance from it.

The rationale behind this heuristic is that for a n-by-n game board, the number of possible moves is higher when positioned away from the corners and the edges. Therefore, taking the already established heuristic of counting the difference of available moves and subtracting a weighted amount of the distance of the players current position from the center of the board will cause the heuristic to lean away from the corners and edges. Many other weights were tried, including values over 1, but values around 25% seemed to perform the best.

The result is a **73.78%** win rate averaged over 10 tournaments.

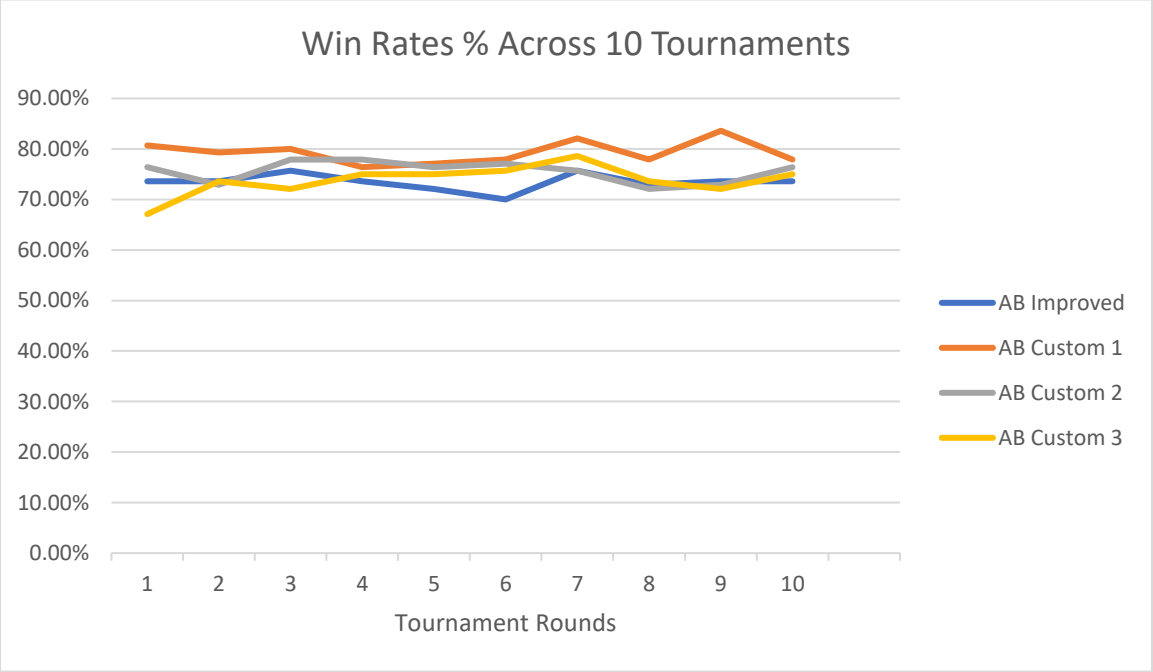

Figure 1.
All tournaments matches were 20 games each for all agent pairings, which each agent getting equal games going first.

Results

Out of all the proposed heuristics, the one that performed the best, consistently, and is the one recommended is the first Custom Score 1 heuristic. This makes sense for the three following reasons. First, in this variant of isolation, considering the number of moves only 1 ply ahead could be quite limiting for there are many situations where you might have immediate move 1 to 2 ply ahead, but from that position on there is none. Therefore, this heuristic improves on that by counting a 1 ply further. Second, it does this in a way that isn't too time consuming, just 2 times more complex than improved score. This allows for more iterations of Iterative Deepening, thus more search depth. And finally, this heuristic accounts for the end game, when moves really matter and one wrong move could lead you to a loss. It does this by counting the total number of possible moves via a depth first search for both players and computing the difference. And since it is endgame, the branching factor would have reduced by quite a bit, thus proved to be not so time consuming.

This can be seen in the data. Compared to the **73.44%** average win rate of the Improved Score heuristic, the Custom Score 1 heuristic proved to be consistently better with a **79.29%** average win rate, albeit just by a little. A final tournament was run to verify the results, this time with 100 matches for each agent pairing. The results, seen below, of this final tournament proved to hold consistent.

Opponent	**AB Improved**	**AB Custom 1**
Random	96 won | 4 lost	99 won | 1 lost
MM Open	81 won | 19 lost	84 won | 16 lost
MM Center	98 won | 2 lost	97 won | 3 lost
MM Improved	83 won | 17 lost	93 won | 7 lost
AB Open	52 won | 48 lost	58 won | 42 lost
AB Center	57 won | 43 lost	58 won | 42 lost
AB Improved	52 won | 48 lost	56 won | 44 lost
Win Rate %	**74.1%**	**77.9%**

Figure 2.

Tournament with 100 games each for all agent pairings. Results displayed are that of the AB Improved heuristic and that of the best and most consistently performing proposed heuristic: AB Custom 1.