Result Analysis of the custom heuristics function

In this project we have experimented with three different heuristic functions, these three functions are the following:

- Custom_score(), in this heuristic I made the same calculation as in heuristic 2 but if the result of the difference is 0 I return the difference of the distances of both player to the center calculated with the Manhattan distance.
- Custom_score_2(), in this heuristic I made the same calculation as in heuristic 2 but if the result of the difference is 0 I return the difference of the distances of both player to the center calculated with the Euclidean distance.
- Custom_score_3(), this heuristic return the difference of legal moves between my player and the opponent and take in count the distance of my move from the center of the gameboard.

Here is a summary of my results:

****

Playing Matches

****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	17	3	18	2	15	5	14	6
2	MM_Open	14	6	13	7	13	7	12	8
3	MM_Center	16	4	18	2	16	4	16	4
4	MM_Improved	11	9	12	8	14	6	14	6
5	AB_Open	11	9	10	10	13	7	12	8
6	AB_Center	10	10	10	10	10	10	12	8
7	AB_Improved	10	10	12	8	10	10	9	11
	Win Rate:	63.6%		66.4%		65.0%		63.6%	

The general results are very similar for the three functions. But, looking at this results and the performance of each heuristic, I would recommend the use of the Custom_Score() heuristic for this reasons:

- His performance is better than the other two.
- Using the Manhattan distance rather than euclidean distance makes the function be better in terms of computing cost.
- In terms of implementation, the three functions has the same level of difficulty.

I changed the NUM_MATCHES parameter setting it to have 20 plays for each round in order to have a better understanding of the performance of the functions.

Below we provide a description of each function and how each perform against every type of opponent provide by Udacity.

Heuristic 1 – Implemented in function Custom_score()

With this heuristic function, we return the difference in number of available legal moves left between the players. If both players have the same number of moves, then the returned value is the scale difference between the Manhattan distance of each player to the center of the board. If the returned value is positive (negative), then the student player is doing better (worse) than its opponent. If the returned value is "inf" ("-inf"), then the student has won (lost) the game.

This heuristic is the best that performs between the three I implemented.

Heuristic 2 – Implemented in function Custom_score_2()

With this heuristic function, we return the difference in number of available legal moves left between the players. If both players have the same number of moves, then the returned value is the scale difference between the Euclidean distance of each player to the center of the board. If the returned value is positive (negative), then the student player is doing better (worse) than its opponent. If the returned value is "inf" ("-inf"), then the student has won (lost) the game.

The performance of this function is better than in the third case.

Heuristic 3 – Implemented in function Custom_score_3()

With this heuristic function, we return the difference in number of available legal moves left between the players adding the distance of the move to the center of the board. So the evaluation value take in count and corrects the difference. If the move is near the center will add a higher value than is far. If the returned value is "inf" ("-inf"), then `player` has won (lost) the game.

This heuristic performs the worst of the three but not so far from them.