Heuristic Analysis

First of all, we want the player to have as many moves as possible. Therefore, the open evaluation function gives the player a good heuristic metric. In addition, we want to the player to be aggressive meaning that the player should also consider moves that will limit the opponent's further moves. The idea brings the improved evaluation function that outputs a score equal to the difference in the number of moves available to the two players. Last but not least, the closer to the center, the better chance of the player will get stuck. Thus the center score evaluation is generated.

Open Improved Heuristic
For the first step, I created a revised version for the "Improved" evaluation function discussed in lecture that gives more weight on player's available number of moves. It calculated the score equal to 2 times of the player's move subtract the opponent's move. This is also a combination of open move evaluation and improved evaluation, indicating a conservative approach that player's number of moves are more important than restricting the opponent's moves.

Here is the tournament experiment result (highlight is the discussed score):
Custom: improved evaluation
Custom_2: open evaluation
Custom_3: 2*player's moves - opponent's moves (i.e. open + improved evaluation)

****
Playing Matches
****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	8	2	10	0
2	MM_Open	7	3	7	3	8	2	7	3
3	MM_Center	7	3	8	2	5	5	10	0
4	MM_Improved	7	3	7	3	9	1	7	3
5	AB_Open	4	6	3	7	7	3	6	4
6	AB_Center	7	3	6	4	4	6	7	3
7	AB_Improved	6	4	5	5	5	5	3	7
	Win Rate:	67.1%		64.3%		65.7%		71.4%	

It seems like the created Custom_3 function performs well compared to other three functions. But after another couple of tests, the results were not so consistent and led to a lower win rate for Custom_3 function.

Aggressive and Conservative Heuristics, Random Heuristics

Therefore, I started to adjust the weight to the range of (1,2) and add a less conservative score evaluation. The aggressive heuristic is meant to eliminate the opponent's moves even though this will sacrifice the player's number of legal moves. While the conservative heuristic is the reverse.

I also added a random scored evaluation function which could be either a conservative or aggressive approach by creating a random number between -1 and 1.

Here is the tournament experiment result:

Custom: random evaluation ((1+rand)*player's move - (1-rand)*opponent's move)
rand is a random number between (-1,1)
Custom_2: aggressive evaluation (player's move - 1.5 opponent's move)
Custom_3: conservative evaluation (1.5*player's move - opponent's move)

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

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	9	1	9	1	9	1
2	MM_Open	7	3	7	3	7	3	7	3
3	MM_Center	7	3	10	0	9	1	9	1
4	MM_Improved	6	4	6	4	5	5	6	4
5	AB_Open	8	2	5	5	6	4	6	4
6	AB_Center	6	4	4	6	6	4	6	4
7	AB_Improved	7	3	4	6	3	7	5	5
	Win Rate:	70.0%		64.3%		64.3%		68.6%	

It seems like the conservative approach gives a better win rate of 68.6% compared to the other two scores. However, this is still lower than the AB_improved score.

Squared Heuristic

For the next step, I replaced the conservative approach, instead, calculate a squared evaluation as Custom_2. This idea is similar to the variance calculation that leverages the difference of the 2 players' number of moves by factoring.

Here is the tournament experiment result:
Custom: random evaluation ((1+rand)*player's move - (1-rand)*opponent's move)
Custom_2: squared evaluation (player's move^2 - opponent's move^2)
Custom_3: conservative evaluation (1.5*player's move - opponent's move)

****
Playing Matches
****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	9	1	7	3	9	1
2	MM_Open	6	4	6	4	6	4	5	5
3	MM_Center	10	0	7	3	10	0	8	2
4	MM_Improved	9	1	7	3	6	4	6	4
5	AB_Open	8	2	5	5	7	3	5	5
6	AB_Center	5	5	5	5	5	5	9	1
7	AB_Improved	2	8	4	6	5	5	6	4
	Win Rate:	68.6%		61.4%		65.7%		68.6%	

Now we found the squared evaluation and conservative evaluation are pretty stable and functioning. Still it not improving much compared to the AB_improved evaluation. In addition, with several tests, the created random evaluation (custom) function seems to be the worst and could be replaced.

Tie-2-Center or Conservative Heuristic
Now I realized that there is one key factor that is the distance between the move and the center. As discussed in the lecture, when the move is close to the center, it's likely that the player may have more choices of move in further games. This rule holds especially for the very first several moves when there are less occupied spaces in the board. In addition, for the very first couple of moves in the game, it's likely to have a draw number of moves for both players. This is a critical time if the player could pick the move that is close to the center instead of a random selection.

Therefore, we would like to build a score that when the two players have same number of moves, pick the move closest to the center of the board; otherwise we use the conservative evaluation approach.

Here is the tournament experiment result:

Custom: tie-2-center or conservative

Custom_2: squared evaluation (player’s move^2 - opponent’s move^2)

Custom_3: conservative evaluation (1.5*player’s move - opponent’s move)

****
Playing Matches
****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	8	2	7	3	7	3
2	MM_Open	6	4	7	3	7	3	7	3
3	MM_Center	5	5	9	1	7	3	8	2
4	MM_Improved	7	3	6	4	6	4	6	4
5	AB_Open	5	5	6	4	4	6	6	4
6	AB_Center	6	4	5	5	8	2	5	5
7	AB_Improved	4	6	6	4	5	5	4	6
	Win Rate:	57.1%		67.1%		62.9%		61.4%	

Now the result shows that the Custom heuristic shows a relatively larger win rate compared to the AB_Improved methods. In addition, both Custom_2 and Custom_3 methods did a good evaluation on the game.

Therefore, we choose the Tie-2-Center or Conservative heuristic for the following 3 reasons:

1. The Tie-2-Center intuitive gives the player a good chance of future moves regardless of picking a random move which might be far from the center and limit the future moves.
2. It seems like the designed additional tie scenario adds complexity to the heuristic but meanwhile it actually helped the algorithm from spending more time to the further depth of the tree search. This will be especially helpful for the very beginning couple of moves of the game.
3. For the other scenario when the players’ number of moves are different, we pick the conservative heuristic because of the previous several experiments show that the conservative heuristic is superior than random or aggressive heuristics.