

The 3 heuristics I have implemented are:

1. Custom: chooses next move closest to the centre of the board
 - a.
2. Custom 2: more aggressive version of 'improved score' heuristic by changing the weight
 - a. $\text{own_moves} - 2 * \text{opp_moves}$
3. Custom 3: one that only aims to reduce the opponent's mobility

I tested the heuristic functions using tournament.py, and all 3 heuristics I built outperformed the AB_Improved.

Below is the summary table for the results. My first heuristic beat all the computer players more than half of the time, apart from AB_Open where the result was 50-50. The second heuristic achieved the highest overall winning rate, but only won 40% of the matches against AB_improved. The third one also did won less than half the matches against AB_Improved.

Based on the results, I have chosen 'closest to centre' as my main heuristic function, because it has at least 50% winning rate against all 7 players, defeated AB_Improved more than half of the time, and it makes intuitive sense that if you stay close to the middle, you are less likely to be isolated.

To obtain a more reliable result, I could increase the number of matches played. However, running the test on 100 matches itself took about an hour to finish. Therefore, I thought I would leave it as 100 for now, and increase it if necessary.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	73	27	86	14	86	14	81	19
2	MM_Open	70	30	54	46	59	41	62	38
3	MM_Center	71	29	59	41	73	27	73	27
4	MM_Improved	48	52	59	41	64	36	56	44
5	AB_Open	48	52	50	50	58	42	45	55
6	AB_Center	53	47	55	45	55	45	51	49
7	AB_Improved	50	50	53	47	40	60	46	54
Win Rate:		59.0%		59.4%		62.1%		59.1%	