

Heuristic Analysis

Martin Lopez

My Results:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	8	2	10	0	9	1
2	MM_Open	9	1	8	2	9	1	7	3
3	MM_Center	10	0	8	2	8	2	9	1
4	MM_Improved	6	4	7	3	8	2	7	3
5	AB_Open	5	5	7	3	5	5	4	6
6	AB_Center	3	7	6	4	6	4	5	5
7	AB_Improved	2	8	3	7	5	5	5	5
Win Rate:		62.9%		67.1%		72.9%		65.7%	

Custom_score(): I used $\text{float}(\text{own_moves} - (\text{weight})\text{opp_moves})$ with weight set to 1.6 on the opponent moves. It almost consistently beat AB_Improved in most matches except the first three. I noticed weights between 1 and 2 did best. I believe giving an emphasis to the opponents moves makes the program more defensive.

Custom_score_2(): I used $\text{return } -\text{float}(\text{diff_in_y}^2 + \text{diff_in_x}^2)$ which relies on the distance between the two opponents. I negate it so that the smaller the distance between the opponents the better.

Custom_score_3(): I used $\text{float}(\text{len}(\text{game.get_legal_moves}(\text{player})) + \text{len}(\text{game.get_blank_spaces}()))$ to calculate and return the sum of the number of the player's moves and the remaining open spaces. It did do better in the matches that Custom_score() did bad on. I'm guessing there might be a better heuristic that uses the benefits of both Custom_score() and Custom_score_2()

Final thoughts: I believe I could make a better heuristic that takes into account the strengths of all 3 of my custom heuristics but simpler runs faster. Since a better heuristic would not let the program go as deep in the search tree, I would recommend my second heuristic. After running tournament.py multiple times it has consistently been the better performing heuristic. It consistently wins match 1 100% of the time, the stats are consistently better than AB_Improved on multiple matches, and it is simple enough that it doesn't take much computational time letting it run through many game states. Making the distance between the two opponents matter is a strong simple heuristic that always beat AB_Improved and my other two heuristics on overall performance on multiple test runs.