Vaishak Kumar

Artificial Intelligence Nanodegree

Heuristic Analysis

The two heuristics that I first considered in two player isolation were:

1. Heuristics that focus on increasing the moves of the player
2. Heuristics that focus on decreasing the moves of the opponent

Increasing the moves of the player:

This strategy is effective and yields 41-44% win rate in the tournament.

Decreasing the moves of the opponent:

This strategy makes the player more aggressive. It has on average one percentage point higher performance than the heuristic above.

Hybrid heuristic:

Then, I experimented with a weighted average of the above two heuristics. With equal weights, it performed better than the first heuristic but worse than the second one. After some trial and error, I found that 2*player_moves – opponent_moves outperforms both the above heuristics, and other combinations of them.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	7	3	8	2	9	1
2	MM_Open	4	6	1	9	3	7	3	7
3	MM_Center	6	4	9	1	7	3	6	4
4	MM_Improved	2	8	3	7	1	9	4	6
5	AB_Open	7	3	8	2	5	5	3	7
6	AB_Center	5	5	4	6	4	6	3	7
7	AB_Improved	5	5	5	5	6	4	6	4
	Win Rate:	51.4%		52.9%		48.6%		48.6%	

AB_Custom: 2*player_moves – opponent_moves

AB_Custom_2: player_moves

AB_Custom_3: – opponent_moves

The recommended heuristic is 2*player_moves − opponent_moves. The reasons for the performance of this heuristic is:

1. It recognizes the importance of both maximizing the player's moves and minimizing the opponent's moves.
2. It avoids the self-inflicted harm that is a result of reckless behavior of agents that emphasize minimizing opponent's moves.
3. It places a premium on game configurations where the player has more moves. Therefore, at each step it choses those branches where the player has a lot of choices.