Heuristic Analysis

Playing Matches

This script evaluates the performance of the *custom_score* evaluation function against a baseline agent using alpha-beta search and iterative deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use ID and alpha-beta search with the custom_score functions defined in game_agent.py.

Match#	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	9	1	9	1	8	2
2	MM_Open	5	5	7	3	5	5	7	3
3	MM_Center	8	2	9	1	6	4	6	4
4	MM_Improved	6	4	6	4	4	6	7	3
5	AB_Open	4	6	8	2	6	4	4	6
6	AB_Center	5	5	5	5	5	5	6	4
7	AB_Improved	5	5	4	6	5	5	2	8
	Win Rate:	57.1%		68.6%		57.1%		57.1%	

The heuristic functions are used to provide the score of the current player. In general the player that choses a position with more moves have less probability to remain blocked. The heuristic functions are build in order to follow that easy rule.

HEURISTIC FUNCTION 1 (AB_CUSTOM)
The heuristic function 1 tries to maximise the current player moves and minimise the adversary player moves with the follow equation:
return player moves - 2 * opponent moves

HEURISTIC FUNCTION 2 (AB_CUSTOM_2)
The heuristic function 1 tries to maximise the current player moves and minimise the adversary player moves with the follow equation:
return player moves - opponent moves

HEURISTIC FUNCTION 3 (AB_CUSTOM_3)
The heuristic function 3 provide the number of the current player moves.
return player moves

Result

The better function is the heuristic function 1 that provides a combination between the number of player and opponent moves where are rewarded the moves with double possibility than the opponent. The three reasons to support the choice of evaluation function 1 are:

1. Best score; The function 1 beat all the other function in term of win rate (win match / number of match)
2. It is simple and fast heuristic. It provides a linear combination between the number of player and opponent moves where the computer player chase after the opponent.
3. Ease of implementation