Heuristic Analysis

Three different heuristics have been implemented in this assignment.

Heuristic 1:

This heuristic uses a simple one policy across the game.

1. maximise own possible moves
2. minimise opponent possible moves with less weight

The motivation behind this heuristic is that it favours a position that our agent have more possible moves. On top of it, it prefers that opponent has fewer possible moves but not as important as the first metric. So a 0.5 parameter is used to lower down the weight of the second metric.

Heuristic 2:

This heuristic divides the game into three stages.

- Early stage: <10% occupied rate on game board
- Mid stage: >=10% and <30% occupied rate on game board
- Final stage: >= 30% occupied rate on game board

In early stage of the game,

1. maximise own possible moves
2. minimise opponent possible moves
3. On top the first two policies, minimise the distance from game board center

In mid stage of the game,

1. maximise own possible moves
2. minimise opponent possible moves

In final stage of the game,

1. maximise own possible moves
2. minimise opponent possible moves with less weight

The motivation for this heuristic is same as the first heuristic: we want to have more moves than our opponent. The special part is that at the beginning of a game, we prefer to go closer to the center.

Heuristic 3:

This heuristic divides the game into two stages.

- Early stage: <40% occupied rate on game board
- Final stage: >=40% occupied rate on game board

In early stage of the game

1. maximise own possible moves
2. minimise opponent possible moves but with a smaller weight

In final stage of the game, which a lots less possible moves exist, search one step deeper.

1. For each possible moves, check the difference of own move and opponent move assuming each move is applied
2. Use the the max difference value as the heuristic

The motivation for this heuristic is that in the final stage of a game, we go one step deeper in each iteration considering the moves left is much fewer than the beginning of a game. However, this will in turn reduce the level of Iterative Deepening.

Experiment and Results

The three custom heuristics are benchmarked against AB_Improved against 7 different game agents. The test result is listed below.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	17	3	20	0	17	3	19	1
2	MM_Open	13	7	15	5	12	8	16	4
3	MM_Center	20	0	19	1	18	2	19	1
4	MM_Improved	15	5	17	3	13	7	13	7
5	AB_Open	9	11	9	11	9	11	9	11
6	AB_Center	8	12	13	7	10	10	11	9
7	AB_Improved	10	10	10	10	10	10	8	12
	Win Rate:	65.7%		73.6%		63.6%		67.9%	

Winning rate is calculated as

	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
Random	85.00%	100.00%	85.00%	95.00%
MM_Open	65.00%	75.00%	60.00%	80.00%
MM_Center	100.00%	95.00%	90.00%	95.00%
MM_Improved	75.00%	85.00%	65.00%	65.00%
AB_Open	45.00%	45.00%	45.00%	45.00%
AB_Center	40.00%	65.00%	50.00%	55.00%
AB_Improved	50.00%	50.00%	50.00%	40.00%

When playing with random agent, all three custom heuristics have high winning rate compared with the benchmark AB_Improved. This is expected as random agent does not have logic in choosing moves.

In games with Min-max agents, all three custom heuristics has over 50% winning rate. This is not surprising as all custom agents has alpha-beta search which means they prune nodes that are not useful and therefore can go deeper in Iterative Deepening. Among three custom heuristics, heuristic 1 has higher winning rate with two types of Min-max agents out of three. Heuristic 1 performed close to or higher than benchmark AB_Improved heuristic.

In games with alpha-beta agents, The winning rate of custom agents start dropping as the advantage in alpha-beta pruning no longer exists. The games are solely depends on heuristics. In games against AB_Open and AB_Center, all custom agents performs similar as or better than benchmark AB_Improved heuristic. Custom 1 heuristic performs the best against AB_Center. In games with AB_Improved, both heuristic 1 and 2 perform same as AB benchmark AB_Improved heuristic.

Comparison and Analysis

Heuristic 1 is very simple. It allows the search to go deeper in Iterative Deepening. In our experiment, it produces higher winning rate that the AB_Improved benchmark heuristic overall (73.6% vs 65.7%).

Heuristic 2 and 3 are more complex. They uses different policies for different stages in a game. Heuristic 2 favours close-to-center moves at the beginning of a game. However, in the experiment, it does not produce higher winning rate than Heuristic 1 (63.6% vs 73.6%) and a little worse than benchmark heuristic overall (65.7%).

Heuristic 3 searches one step deeper in each iteration in the final stage of a game. As a result, it reduces the level of Iterative Deepening. In the experiment, it does not produce higher winning rate than Heuristic 1 (67.9% vs 73.6%), but better than benchmark heuristic overall (65.7%).

Recommendation

Based on the comparison and analysis, heuristic 1 is the recommended the heuristic. The reasons for this recommendation are

1. Heuristic 1 has the best winning rate in the experiment. Heuristic achieved an overall winning rate of 73.6%, which is the highest among benchmark heuristic 65.7%, heuristic 2 63.6% and heuristic 3 67.6%.
2. Heuristic 1 has a single policy during the entire game. It does not require additional branching and different calculation like heuristic 2 and 3. Therefore it is very easy to implement and easy to reason about.
3. Because heuristic 1 has the lowest time complexity, it allows the Iterative Deepening to search deeper than more complex heuristic like heuristic 3.