

Heuristic Analysis

Markus Meier

October 2, 2017

1 Heuristic 1

The idea behind this heuristic is that it is advantageous for the player to choose board states that allow him more moves in his next turn.

2 Heuristic 2

The idea behind this heuristic is that it is advantageous for the player to limit the freedom of his opponent. That means the less moves the opponent has the better for the player. This might be scaled with a certain aggressive factor.

3 Heuristic 3

This heuristic tries to combine the first two approaches.

We count for both players the available moves. However, moves that lead close to the border of the game board are penalized, since they will have less freedom in the following steps of the game.

4 Evaluation

On the following pages, there are some screenshots from the evaluations in three tournaments. The results are not consistent for all the scores, so it is hard to tell if heuristic 3 is really better. More samples could solve the issue, but require more time.

```
(aind) mmeier@archlinux ~/git/AIND-Isolation master python3 tournament.py
```

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.

Playing Matches

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	8	2	10	0	10	0
2	MM_Open	9	1	5	5	9	1	8	2
3	MM_Center	10	0	7	3	9	1	10	0
4	MM_Improved	10	0	6	4	8	2	9	1
5	AB_Open	7	3	2	8	4	6	6	4
6	AB_Center	4	6	6	4	5	5	5	5
7	AB_Improved	3	7	1	9	4	6	5	5
Win Rate:		75.7%		50.0%		70.0%		75.7%	

```
(aind) mmeier@archlinux ~/git/AIND-Isolation master python3 tournament.py
```

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.

Playing Matches

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	8	2	10	0	10	0
2	MM_Open	9	1	7	3	8	2	8	2
3	MM_Center	10	0	6	4	10	0	10	0
4	MM_Improved	6	4	6	4	8	2	8	2
5	AB_Open	6	4	4	6	3	7	7	3
6	AB_Center	5	5	3	7	6	4	7	3
7	AB_Improved	5	5	4	6	4	6	5	5
Win Rate:		72.9%		54.3%		70.0%		78.6%	

```
(a) mmeier@archlinux ~/git/AIND-Isolation master python3 tournament.py

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.

*****
Playing Matches
*****

Match #   Opponent   AB_Improved   AB_Custom   AB_Custom_2   AB_Custom_3
              Won | Lost   Won | Lost   Won | Lost   Won | Lost
1         Random     10 | 0       8 | 2       10 | 0       10 | 0
2         MM_Open     8 | 2       2 | 8       10 | 0       9 | 1
3         MM_Center    9 | 1       9 | 1       10 | 0       10 | 0
4         MM_Improved  8 | 2       5 | 5       9 | 1       9 | 1
5         AB_Open     5 | 5       5 | 5       5 | 5       6 | 4
6         AB_Center    6 | 4       3 | 7       6 | 4       4 | 6
7         AB_Improved  6 | 4       3 | 7       5 | 5       3 | 7
-----
Win Rate:   74.3%       50.0%       78.6%       72.9%
```