

Heuristic Analysis

June 28, 2017

0.1 Heuristics

This section outlines the three heuristics used in the Isolation game.

0.1.1 Heuristic 1 (Custom)

This heuristic is based on player's move imbalance and keeping distance from the opponent. Heuristic calculates the move imbalance by finding total moves available for the player and it's opponent. Imbalance is calculated as below:

Move Imbalance = $\text{my_moves} / (\text{my_moves} + \text{opponent_moves})$

Distance from opponent is calculated by measuring the Manhattan distance from opponent's location.

Heuristic score is based on the move that maximises the imbalance in the favour of current player by maintaining maximum possible distance from the opponent.

heuristic score = imbalance * distance

0.1.2 Heuristic 2 (Custom_2)

This heuristic is based on player's move imbalance and tracking the opponent closely. Heuristic calculates the move imbalance by finding total moves available for the player and it's opponent. Imbalance is calculated as below:

Move Imbalance = $\text{my_moves} / (\text{my_moves} + \text{opponent_moves})$

Distance from opponent is calculated by measuring the Euclidean distance from opponent's location.

Heuristic score is based on the move that maximises the imbalance in the favour of current player by maintaining nearest possible distance from the opponent.

heuristic score = imbalance * (1 - distance / max_euclidean_distance)

where max_euclidean_distance is the diagonal of the game board.

0.1.3 Heuristic 3 (Custom_3)

This heuristic is a variant of the first heuristic.

It measures the distance from opponent by calculating Euclidean distance from opponent's location, instead of Manhattan distance.

Heuristic score is based on the move that maximises the imbalance in the favour of current player by maintaining maximum possible Euclidean distance from the opponent.

heuristic score = imbalance * distance

0.2 Results

Results are analysed over 30 tournaments, with each tournament consisting of 7 games with different opponents. All three heuristics are performing better than the heuristic that maximises the differences in the move between the player and the opponent.

On average, heuristic #1 is performing better than other heuristics. This heuristic is aiming to maximise the available moves for the player by maintaining maximum possible distance from the opponent. Keeping maximum distance from the opponent results in higher chances of win as two players will not be in a competition to occupy the positions around them. A high favourable imbalance indicates player has more moves available than the opponent which gives higher probability of winning the game.

Below is the summary of tournament results.

```
In [3]: import pandas as pd
        from IPython.display import display

        result = pd.read_csv("logs/results.csv")
        summary = pd.read_csv("logs/results_summary.csv")
```

0.3 Tournament Results

```
In [4]: display(result)
```

	Tournament	AB_Improved	AB_Custom	AB_Custom_2	AB Custom_3
0	1	55.7	78.6	77.1	75.7
1	2	67.1	67.1	71.4	74.3
2	3	60.0	71.4	74.3	70.0
3	4	65.7	74.3	72.9	81.4
4	5	62.9	72.9	75.7	64.3
5	6	55.7	70.0	71.4	80.0
6	7	60.0	71.4	70.0	64.3
7	8	64.3	77.1	74.3	77.1
8	9	60.0	77.1	70.0	74.3
9	10	65.7	77.1	77.1	74.3
10	11	60.0	80.0	75.7	77.1
11	12	58.6	84.3	77.1	75.7
12	13	54.3	78.6	74.3	72.9
13	14	62.9	74.3	74.3	75.7
14	15	55.7	75.7	77.1	68.6
15	16	57.1	74.3	70.0	75.7
16	17	68.6	75.7	77.1	80.0
17	18	57.1	75.7	77.1	75.7
18	19	54.3	77.1	74.3	70.0
19	20	55.7	74.3	77.1	70.0
20	21	55.7	70.0	70.0	70.0
21	22	62.9	72.9	72.9	75.7
22	23	55.7	81.4	74.3	70.0
23	24	68.6	70.0	77.1	65.7
24	25	55.7	74.3	80.0	74.3

25	26	58.6	75.7	68.6	78.6
26	27	58.6	72.9	74.3	80.0
27	28	64.3	77.1	75.7	77.1
28	29	61.4	77.1	72.9	78.6
29	30	65.7	64.3	71.4	74.3

0.4 Tournament Results Summary

In [5]: display(summary)

	Summary	AB Improved	AB Custom	AB Custom_2	AB Custom_3
0	Min	54.30	64.30	68.60	64.30
1	Max	68.60	84.30	80.00	81.40
2	Median	60.00	75.00	74.30	75.00
3	Avg	60.29	74.76	74.18	74.05