

Report – Heuristic Analysis

In this task, another 5 addition evaluation functions are implemented and compared. For my code, I added 2 functions to compare with others. (my_heuristic1 and my_heuristic2).

My Custom Heuristics:

1) my_heuristic1: Minimizing opponents moves and maximizing our moves

This heuristic is based on the notion that at any time, the number of moves by the opponent are minimized and try to maximize our moves. The mathematical expression for this can be represented as:

$$\text{len}(\text{my_moves}) * \text{len}(\text{my_moves}) - \alpha * \text{len}(\text{opponent_moves}) * \text{len}(\text{opponent_moves})$$

where α has been chosen as 5.0.

2) my_heuristic2: Aggressive to Defensive

For this heuristic, during the first half of the game where the occupied space of the board is less than half, the player plays aggressively by attempting to minimize and limit the opponent's available moves at weighted cost against the player's available moves. When the occupied space of the board is more than half, the player switches to defensive strategy by maximizing available moves of the player at a weight cost against the opponent's moves.

It can be mathematically expressed as:

$$\begin{aligned} \text{ratio} &= \text{game.move_count} / (\text{game.width} * \text{game.height}) \\ \text{len}(\text{my_moves}) - \alpha * \text{len}(\text{opponent_moves}) &\text{ if ratio} \leq 0.5 \\ \text{else } \alpha * \text{len}(\text{my_moves}) - \text{len}(\text{opponent_moves}) \end{aligned}$$

where α has been chosen as 3.

Group Members' custom Heuristics:

3) haoshan_heuristic: Point of View Strategy

For this heuristic, the value is calculated from the point of view of the given player. It was composed of two types of rules, the progression of the game which is the percentage of the game board filled and whether the player or opponent crash with the boundary. It can be expressed as:

The percentage of unoccupied spaces:

$$(\text{len}(\text{current blank spaces}) / (\text{board width} * \text{board height})) * 100$$

Rules:

When the player agent is far from the boundary and percentage is low, get a higher score(my score), vice versa. Same rules applied to the opponent agent(opponent score).

Heuristic:

$(\text{my score} - \text{opponent score} * 1.5)$

4) sinmei_heuristic: Difference between average scores for future moves of player and opponent

This heuristics technique required pre-computation of the value of each space to evaluate each move in the game. In the Isolation board game, there are 8 possible movements, this heuristic will check if the movement is valid and calculate the values for the spaces according to their positions and possibilities to get to the space from the center with a discount of 0.9. The space at the center point of the board has the maximum value of 10, if the space is nearer to the center and has higher possibility to access, then higher score will be given and vice versa. With the precomputed values of spaces, the difference between average scores of the players and opponents can be calculated as below:

$\text{my_score} = \text{len}(\text{my available move}) + \text{value of my last move} + \text{mean}(\text{value of my available move})$

$\text{opponent_score} = \text{len}(\text{available opponent move}) + \text{value of opponent last move} + \text{mean}(\text{value of opponent available move})$

$\text{difference between average scores} = \text{my_score} - \text{opponent_score}$

5) jiaqi_heuristic: Move differences with weight factor

This heuristics technique is calculating the difference between the moves of the player and opponent, different weight factors are applied to both players and opponents to optimize the performance of the player. Weight factor for the player is set to a constant of 1, for opponent is based on the number of open spaces remaining in the game. This heuristics technique can be expressed as:

$\text{player weight factor} * \text{len}(\text{my available moves}) + ((\text{total number of spaces}) / (\text{number of space occupied}) - \text{opponent factor}) * \text{len}(\text{available opponent moves})$

This evaluation function is depending on the number of occupied spaces in the game, so the player will perform differently from time to time according to the number of blank spaces in the game. From the experiment, this evaluation function outperforms the offensive to defensive strategy.

The tournament opponents are listed below:

- Random: An agent that randomly chooses a move each turn.
- MM_Null: CustomPlayer agent using fixed-depth minimax search and the null_score heuristic
- MM_Open: CustomPlayer agent using fixed-depth minimax search and the open_move_score heuristic
- MM_Improved: CustomPlayer agent using fixed-depth minimax search and the improved_score heuristic
- AB_Null: CustomPlayer agent using fixed-depth alpha-beta search and the null_score heuristic
- AB_Open: CustomPlayer agent using fixed-depth alpha-beta search and the open_move_score heuristic
- AB_Improved: CustomPlayer agent using fixed-depth alpha-beta search and the improved_score heuristic

- ID_Improved: CustomPlayer agent using iterative alpha-beta search and the improved_score heuristic
- Student1: CustomPlayer agent using iterative alpha-beta search and the heuristic 1
- Student2: CustomPlayer agent using iterative alpha-beta search and the heuristic 2
- Student3: CustomPlayer agent using iterative alpha-beta search and the heuristic 3
- Student4: CustomPlayer agent using iterative alpha-beta search and the heuristic 4
- Student5: CustomPlayer agent using iterative alpha-beta search and the heuristic 5
- Student6: CustomPlayer agent using iterative alpha-beta search and the heuristic 6
- Student7: CustomPlayer agent using iterative alpha-beta search and the heuristic 7

ADDED

- CHEAH JO YEN: CustomPlayer agent using iterative alpha-beta search and my_heuristic1
- CHEAH JO YEN-2: CustomPlayer agent using iterative alpha-beta search and my_heuristic2
- WONG HAO SHAN: CustomPlayer agent using iterative alpha-beta search and haoshan_heuristic
- CHONG SIN MEI: CustomPlayer agent using iterative alpha-beta search and sinmei_heuristic
- LIM JIA QI: CustomPlayer agent using iterative alpha-beta search and jiaqi_heuristic

Result

Agent	Performance	Ranking
CHEAH JO YEN	67.50%	3
CHEAH JO YEN-2	65.89%	5
WONG HAO SHAN	65.54%	6
CHONG SIN MEI	66.79%	4
LIM JIA QI	68.75%	1
ID_Improved	60.00%	11
Student1	64.64%	8
Student2	67.86%	2
Student3	65.89%	5
Student4	62.14%	10
Student5	64.82%	7
Student6	63.21%	9
Student7	68.75%	1

The table above shows the performance and ranking of agents using different heuristics:

This result obtained by running 20 matches with 150ms as the time limit. Deepening iterative is implemented by setting True for iterative in `CUSTOM_ARGS`

```
CUSTOM_ARGS = {"method": 'alphabeta', 'iterative': True}
```

From this experiment, all our custom heuristics agents outperforms the ID_Improved agent. Besides, for my custom heuristic both ranked at 3 and 5 which is good.

The raw evaluation result can be found on the next page.

Evaluating: CHEAH JO YEN

Playing Matches:

Match 1:	CHEAH JO YEN	vs Random	Result: 72 to 8
Match 2:	CHEAH JO YEN	vs MM_Null	Result: 56 to 24
Match 3:	CHEAH JO YEN	vs MM_Open	Result: 45 to 35
Match 4:	CHEAH JO YEN	vs MM_Improved	Result: 45 to 35
Match 5:	CHEAH JO YEN	vs AB_Null	Result: 60 to 20
Match 6:	CHEAH JO YEN	vs AB_Open	Result: 53 to 27
Match 7:	CHEAH JO YEN	vs AB_Improved	Result: 47 to 33

Results:

CHEAH JO YEN 67.50%

Evaluating: CHEAH JO YEN-2

Playing Matches:

Match 1:	CHEAH JO YEN-2	vs Random	Result: 70 to 10
Match 2:	CHEAH JO YEN-2	vs MM_Null	Result: 56 to 24
Match 3:	CHEAH JO YEN-2	vs MM_Open	Result: 48 to 32
Match 4:	CHEAH JO YEN-2	vs MM_Improved	Result: 44 to 36
Match 5:	CHEAH JO YEN-2	vs AB_Null	Result: 56 to 24
Match 6:	CHEAH JO YEN-2	vs AB_Open	Result: 46 to 34

Match 7: CHEAH JO YEN-2 vs AB_Improved Result: 49 to 31

Results:

CHEAH JO YEN-2 65.89%

Evaluating: WONG HAO SHAN

Playing Matches:

Match 1: WONG HAO SHAN vs Random Result: 73 to 7
Match 2: WONG HAO SHAN vs MM_Null Result: 62 to 18
Match 3: WONG HAO SHAN vs MM_Open Result: 46 to 34
Match 4: WONG HAO SHAN vs MM_Improved Result: 40 to 40
Match 5: WONG HAO SHAN vs AB_Null Result: 52 to 28
Match 6: WONG HAO SHAN vs AB_Open Result: 45 to 35
Match 7: WONG HAO SHAN vs AB_Improved Result: 49 to 31

Results:

WONG HAO SHAN 65.54%

Evaluating: CHONG SIN MEI

Playing Matches:

Match 1: CHONG SIN MEI vs Random Result: 75 to 5

Match 2: CHONG SIN MEI vs MM_Null Result: 69 to 11
Match 3: CHONG SIN MEI vs MM_Open Result: 41 to 39
Match 4: CHONG SIN MEI vs MM_Improved Result: 42 to 38
Match 5: CHONG SIN MEI vs AB_Null Result: 64 to 16
Match 6: CHONG SIN MEI vs AB_Open Result: 49 to 31
Match 7: CHONG SIN MEI vs AB_Improved Result: 34 to 46

Results:

CHONG SIN MEI 66.79%

Evaluating: LIM JIA QI - 17134267

Playing Matches:

Match 1: LIM JIA QI - 17134267 vs Random Result: 69 to 11
Match 2: LIM JIA QI - 17134267 vs MM_Null Result: 63 to 17
Match 3: LIM JIA QI - 17134267 vs MM_Open Result: 47 to 33
Match 4: LIM JIA QI - 17134267 vs MM_Improved Result: 43 to 37
Match 5: LIM JIA QI - 17134267 vs AB_Null Result: 59 to 21
Match 6: LIM JIA QI - 17134267 vs AB_Open Result: 49 to 31
Match 7: LIM JIA QI - 17134267 vs AB_Improved Result: 55 to 25

Results:

LIM JIA QI - 17134267 68.75%

Evaluating: ID_Improved

Playing Matches:

Match 1: ID_Improved vs Random Result: 68 to 12
Match 2: ID_Improved vs MM_Null Result: 58 to 22
Match 3: ID_Improved vs MM_Open Result: 38 to 42
Match 4: ID_Improved vs MM_Improved Result: 33 to 47
Match 5: ID_Improved vs AB_Null Result: 49 to 31
Match 6: ID_Improved vs AB_Open Result: 47 to 33
Match 7: ID_Improved vs AB_Improved Result: 43 to 37

Results:

ID_Improved 60.00%

Evaluating: Student1

Playing Matches:

Match 1: Student1 vs Random Result: 67 to 13
Match 2: Student1 vs MM_Null Result: 61 to 19
Match 3: Student1 vs MM_Open Result: 48 to 32
Match 4: Student1 vs MM_Improved Result: 40 to 40
Match 5: Student1 vs AB_Null Result: 52 to 28
Match 6: Student1 vs AB_Open Result: 47 to 33
Match 7: Student1 vs AB_Improved Result: 47 to 33

Results:

Student1 64.64%

Evaluating: Student2

Playing Matches:

Match 1: Student2 vs Random Result: 71 to 9
Match 2: Student2 vs MM_Null Result: 60 to 20
Match 3: Student2 vs MM_Open Result: 36 to 44
Match 4: Student2 vs MM_Improved Result: 41 to 39
Match 5: Student2 vs AB_Null Result: 60 to 20
Match 6: Student2 vs AB_Open Result: 63 to 17
Match 7: Student2 vs AB_Improved Result: 49 to 31

Results:

Student2 67.86%

Evaluating: Student3

Playing Matches:

Match 1: Student3 vs Random Result: 70 to 10
Match 2: Student3 vs MM_Null Result: 51 to 29
Match 3: Student3 vs MM_Open Result: 43 to 37
Match 4: Student3 vs MM_Improved Result: 44 to 36
Match 5: Student3 vs AB_Null Result: 57 to 23
Match 6: Student3 vs AB_Open Result: 52 to 28
Match 7: Student3 vs AB_Improved Result: 52 to 28

Results:

Student3 65.89%

Evaluating: Student4

Playing Matches:

Match 1: Student4 vs Random Result: 67 to 13
Match 2: Student4 vs MM_Null Result: 54 to 26
Match 3: Student4 vs MM_Open Result: 42 to 38
Match 4: Student4 vs MM_Improved Result: 40 to 40
Match 5: Student4 vs AB_Null Result: 52 to 28
Match 6: Student4 vs AB_Open Result: 48 to 32
Match 7: Student4 vs AB_Improved Result: 45 to 35

Results:

Student4 62.14%

Evaluating: Student5

Playing Matches:

Match 1:	Student5	vs	Random	Result: 73 to 7
Match 2:	Student5	vs	MM_Null	Result: 59 to 21
Match 3:	Student5	vs	MM_Open	Result: 46 to 34
Match 4:	Student5	vs	MM_Improved	Result: 39 to 41
Match 5:	Student5	vs	AB_Null	Result: 51 to 29
Match 6:	Student5	vs	AB_Open	Result: 50 to 30
Match 7:	Student5	vs	AB_Improved	Result: 45 to 35

Results:

Student5 64.82%

Evaluating: Student6

Playing Matches:

Match 1:	Student6	vs	Random	Result: 68 to 12
Match 2:	Student6	vs	MM_Null	Result: 58 to 22
Match 3:	Student6	vs	MM_Open	Result: 45 to 35
Match 4:	Student6	vs	MM_Improved	Result: 42 to 38
Match 5:	Student6	vs	AB_Null	Result: 53 to 27

Match 6: Student6 vs AB_Open Result: 45 to 35

Match 7: Student6 vs AB_Improved Result: 43 to 37

Results:

Student6 63.21%

Evaluating: Student7

Playing Matches:

Match 1: Student7 vs Random Result: 72 to 8

Match 2: Student7 vs MM_Null Result: 57 to 23

Match 3: Student7 vs MM_Open Result: 51 to 29

Match 4: Student7 vs MM_Improved Result: 38 to 42

Match 5: Student7 vs AB_Null Result: 66 to 14

Match 6: Student7 vs AB_Open Result: 51 to 29

Match 7: Student7 vs AB_Improved Result: 50 to 30

Results:

Student7 68.75%