Heuristic Analysis of Isolation Game

by Yanfei Wu

For the isolation game, custom agents ('AB_Custom') using alpha-beta search and iterative deepening with three custom Heuristic functions were created. By competing against a baseline agent also using alpha-beta search and iterative deepening called 'AB_Improved' in a round-robin tournament, the strength rating of each custom Heuristic was evaluated. The results of the tournament and the performance of each Heuristic function is discussed in this report.

- **Heuristic 1**

The first Heuristic evaluates the **ratio** between the **difference in number of moves left** between the player and the opponent and their **Manhattan distance**. The more moves the player has left than the opponent (more advantageous board state) and the closer the player is to the opponent (more easily to block the opponent), the higher the heuristic score is for the player in the current game state. Below is the tournament result:

Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	14	6	17	3
2	MM_Open	12	8	13	7
3	MM_Center	14	6	15	5
4	MM_Improved	13	7	12	8
5	AB_Open	11	9	13	7
6	AB_Center	11	9	11	9
7	AB_Improved	9	11	11	9
	Win Rate:	60.0%		65.7%	

The winning rate of 'AB_Custom' is higher than 'AB_Improved' by 5.7% in this tournament.

- **Heuristic 2**

The second Heuristic evaluates the difference in **total number of moves left** in both **current and future** board states between the player and the opponent. The more total available moves the player has, the higher the heuristic score is for the player. This Heuristic takes into account how the player will potentially limit the future moves of the opponent. The tournament result is shown below:

Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	17	3	18	2
2	MM_Open	13	7	10	10
3	MM_Center	18	2	16	4
4	MM_Improved	9	11	14	6
5	AB_Open	11	9	10	10
6	AB_Center	11	9	12	8
7	AB_Improved	10	10	10	10
	Win Rate:	63.6%		64.3%	

'AB_Custom' using this second Heuristic function performs slightly better than the baseline agents 'AB_Improved' with only 0.7% higher winning rate.

- **Heuristic 3**

The third Heuristic evaluates the **difference in the number of moves left** or the **difference in the Manhattan distance to the board center** between the player and the opponent. This is an even simpler Heuristic function. When the player has more moves left than the opponent, the Heuristic function assigns higher score to the player. When they both have the same number of moves left, their Manhattan distances to the board center are compared. The closer to the board center, the more advantageous as this player is more likely to do better than the one closer to the edge of the board. The table below shows the tournament result using this Heuristic:

Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	16	4	18	2
2	MM_Open	13	7	12	8
3	MM_Center	15	5	15	5
4	MM_Improved	13	7	12	8
5	AB_Open	10	10	10	10
6	AB_Center	11	9	10	10
7	AB_Improved	9	11	11	9
	Win Rate:	62.1%		62.9%	

This Heuristic function, similar to the second one, only leads to a marginal win for 'AB_Custom' compared to 'AB_Improved' in the winning rate, by only 0.8%.

Overall, these three Heuristic functions are all easily interpretable and can be easily implemented. They all achieved a winning rate comparable or higher than the baseline agents. These Heuristic functions evaluate the board state by combinations of simple positional advantages, including available moves (current, future), Manhattan distance (to the board center, to the opponent). Among these three, the first Heuristic function performs best with a much larger winning rate difference than the baseline agent 'AB_Improved'. The rest of the two Heuristic functions only lead to marginal wins in winning rate. Therefore, out of these three Heuristic functions, the first one should be preferred.

Note that none of these three Heuristic functions takes account into the specific rule of the isolation game (the allowed 'L-shape' moves like a knight in chess). Therefore, there should potentially be better Heuristic functions that can be implemented. But as far as both simplicity and performance are concerned, the proposed Heuristic function is a good option.