Leon Truong – ANID - 2017

Heuristic Analysis

Summary

Heuristics are rules or evaluations to find different ways or methods to improve the efficiency when solving a problem. In this case, if the current position is a good position to be in. This is evaluated using a cost value or score to maximise. To identify if the current position is good in a game of isolation, the program needs to evaluate the position of the player, the opponent and the game state. To achieve an optimal heuristic it needs to be an admissible heuristic that never overestimates the cost of reaching the goal.

Heuristic 3 was chosen as the main heuristic to use as it uses a weighted sum. This weighted sum, weights both heuristic 1 and 2 to form a multi-problem solver. This achieved a 10% win rate over the AB_improved. The implementation of this algorithm is not complex to incorporate, except hand-crafting the weights is required. The efficiency of this weighted sum, even though it is a combination of multiple heuristics, it still executes within the time limit.

Overall Comparison

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	10	0	10	0
2	MM_Open	5	5	5	5	7	3	8	2
3	MM_Center	8	2	5	5	9	1	8	2
4	MM_Improved	6	4	7	3	5	5	6	4
5	AB_Open	5	5	4	6	4	6	4	6
6	AB_Center	3	7	4	6	5	5	6	4
7	AB_Improved	5	5	5	5	3	7	5	5
Win rate		58.6%		55.%		61.4%		67.1%	

In the table above, I have highlighted the areas where the agent has occurred a loss against the opponent. AB_Open shows a recurring theme of losses in all three custom scores. However heuristic three overcomes or performs just as good as the previous scoring methods. Analysis of heuristic is outlined below.

Heuristic 1

This function evaluates the player's position, if the player is closer to the centre of the board, they have a higher probability of winning. I noticed through human to human plays, we were gravitating to the centre of the board as a defensive manoeuvrer. This central position is evaluated against the opponent's position.

If the player's position is closer to the centre than the opponent, the score will result in a positive, else a negative. If both players are equal distance from the centre, it will result in a score of 0.

Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	9	1	9	1
2	MM_Open	5	5	5	5
3	MM_Center	8	2	5	5
4	MM_Improved	6	4	7	3
5	AB_Open	5	5	4	6
6	AB_Center	3	7	4	6
7	AB_Improved	5	5	5	5
Total Win Rate:		58.6%		55.%	

This heuristic resulted in having a lower winning rate than AB_improved. This may suggest always close to the centre of the board may not be the best heuristic to use. It is however still a valid tactic.

Heuristic 2

This function evaluates the player's position and the opponent's position and aggressively chases opponent. This is a variation difference of numbers (improved_score) to assign a higher penalty of being further away from opponent.

Match #	Opponent	AB_Improved		AB_Custom_2	
		Won	Lost	Won	Lost
1	Random	9	1	10	0
2	MM_Open	5	5	7	3
3	MM_Center	8	2	9	1
4	MM_Improved	6	4	5	5
5	AB_Open	5	5	4	6
6	AB_Center	3	7	5	5
7	AB_Improved	5	5	3	7
Total Win Rate:		58.6%		61.4%	

As a result, AB_open performs similar as AB_custom_2. As AB_custom_2 applies a double the penalty of being further away from the opponent. Thus it performed better than AB_improved.

Heuristic 3

This function uses a weighted score to combine both evaluation functions to hopefully form one superior one. This was decided as a heuristic as a person doesn't just use one strategy in a game, it depends on the position of the player. The weighting was decided by testing using different scoring of the win rates. The first heuristic did worse, so it was assigned a lower value, the second heuristic performed best and a third method was included to improve the possibility of the result.

Match #	Opponent	AB_Improved		AB_Custom_3	
		Won	Lost	Won	Lost
1	Random	9	1	10	0
2	MM_Open	5	5	8	2
3	MM_Center	8	2	8	2
4	MM_Improved	6	4	6	4
5	AB_Open	5	5	4	6
6	AB_Center	3	7	6	4
7	AB_Improved	5	5	5	5
Total Win Rate:		58.6%		67.1%	

As a result, heuristic three outperforms the two heuristics. This is a near 10% increase. The weights that I've currently applied are hard coded. A better heuristic should change the weighting of each feature dynamically to cater for various situations.