

Heuristic analysis

Jinxiang Ruan

In my custom evaluation functions, basically I take the below factors into account:

- my_moves
- opponent_moves
- my_centrality (distance from the centre of the board)
- opponent_centrality
- common_moves (shared space both accessible by me and opponent)

Custom Heuristics

Heuristic 1: Minimizing opponent's moves and play close to the centre

The heuristic is based on the logic that opponent's moves should be minimized, and try to occupy the centre of the board and force the opponent to move towards the border. It can be mathematically expressed as:

$$\text{len(my_moves)} - a * \text{len(opponent_moves)} + \text{my_centrality} - \text{opponent_centrality}$$

The value of a was empirically chosen as 2

Heuristic 2: Heuristic 1 + maximizing common moves

The heuristic is trying to play more offensively, to consider when there are more common moves, there would be more chance for me to occupy the opponents' move. It can be mathematically expressed as:

$$\text{len(my_moves)} - a * \text{len(opponent_moves)} + \text{my_centrality} - \text{opponent_centrality} + \text{len(common_moves)}$$

Heuristic 3: Heuristic 1 + minimizing common moves

The heuristic is trying to play more defensively, to avoid move into a position closed to the opponent, to limit the chance that the opponent could block me. It can be mathematically expressed as:

$$\text{len(my_moves)} - a * \text{len(opponent_moves)} + \text{my_centrality} - \text{opponent_centrality} - \text{len(common_moves)}$$

Performance

The tournament.py script is used to evaluate the effectiveness of your custom heuristics. The script measures relative performance of your agent (named "Student" in the tournament) in a round-robin tournament against several other pre-defined agents. The Student agent uses time-limited Iterative Deepening along with your custom heuristics.

The result of various agents is as below:

Playing Matches					

Match #	Opponent	AB_Improved Won Lost	AB_Custom Won Lost	AB_Custom_2 Won Lost	AB_Custom_3 Won Lost
1	Random	7 3	10 0	7 3	9 1
2	MM_Open	7 3	7 3	8 2	8 2
3	MM_Center	6 4	7 3	9 1	8 2
4	MM_Improved	5 5	8 2	3 7	5 5
5	AB_Open	4 6	6 4	5 5	6 4
6	AB_Center	7 3	7 3	7 3	6 4
7	AB_Improved	6 4	8 2	5 5	6 4

Win Rate:		60.0%	75.7%	62.9%	68.6%

Agent	Win Rate	Rank
ID_Improve	60.0%	4
Heuristics 1	75.5%	1
Heuristics 2	62.9%	3
Heuristics 3	68.6%	2

All the custom heuristics perform better than ID_Improve by a reasonable margin as can be seen in the above table. Compare to the ID_Improve, the custom heuristics play more aggressively by applying a ratio 2:1 to minimizing the opponent's moves, and try to occupy the centre of the board and force the opponent to move towards the border.

The *custom_score* function had been implemented to execute **Heuristic 1** as:

1. It outperforms all other heuristics with win rate 75.5% which is higher than all the other heuristics being considered.
2. It depends on the current state of the board only and no additional game tree search is required
3. It is easier to implement and involves only a few calculations
4. It aligns with the strategy that occupying the centre of the board and forcing your opponent to move towards the border would increase your chance of winning.