Heuristic Analysis

Pietro Rancati

Heuristics

Different heuristics utilizing different combination of the available features of the board states (i.e. Available player moves, available opponent moves, moves in common, distance of players from the center etc.) have been implemented and tested. Below are reported and described the three heuristics that performed the best during the tournament (the number of matches has been increased from 10 to 100 to have more stability of the results).

Custom score 1:

The first custom score is a modification of the improved score discussed in the lecture plus the consideration of the distance of the player position and the opponent position to the center of the board. It is defined as:

$$player\_moves - opponent\_moves - centrality(player) * \alpha + centrality(opponent) * \alpha$$

Where player_moves defines the available moves the current player has in the current board state, opponent_moves defines the available moves the opponent player has in the current board state, centrality defines the distance from a position to the centre of the board (value range between 0 [centre of the board] and 1 [corner of the board]), α is a factor to weight the centrality based on the state of the board (α=(player_moves + opponent_moves) / 2).

Custom score 2:

The second custom score is a modification of the improved score discussed in the lecture. It is defined as:

$$player\_moves - \beta * opponent\_moves$$

Where player_moves define the available moves the current player has with the current board state, opponent_moves define the available moves the opponent player has with the current board state, and β is a multiplicative factor (empirically set to 1.5) to make the agent play in a more "aggressive" way, trying to reduce the moves the opponent has.

Custom score 3:

The third custom score is based on the logic that the player should have more moves in comparison to the opponent and that the opponent should have less moves in comparison to the player. It is defined as:

```
α * (player_moves / opponent_moves) - β * (opponent_moves / player_moves)
```

Where player_moves define the available moves the current player has with the current board state, opponent_moves define the available moves the opponent player has with the current board state, and the α and β multiplicative factors (empirically set to 1. and 2. respectively) represent the importance of the two ratios in the calculation of the score.

Results and conclusions

The results achieved by the custom scores in the tournament are the following:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	90	10	94	6	96	4	94	6
2	MM_Open	74	26	74	26	74	26	75	25
3	MM_Center	87	13	86	14	83	17	84	16
4	MM_Improved	72	28	74	26	74	26	67	33
5	AB_Open	46	54	52	48	56	44	50	50
6	AB_Center	56	44	59	41	55	45	63	37
7	AB_Improved	50	50	44	56	52	48	54	46
	Win Rate:	67.9%		69.0%		70.0%		69.6%	

As we can see, all the custom scores achieved slightly better results that the improved score discussed in the lecture.
We can derive by the results of custom score 2 and 3, that make the agent playing in a more aggressive way (trying to reduce the number of moves the opponent has) usually pays off and result in a bigger chance of winning.
Also, the score where the centrality factor has been taken into consideration achieve better results, denoting that assuming a more central position during the game is a winning factor.

However, multiple improvements could be introduced:

- Combination of the different heuristics proposed and the introduction of new factors.
- Better selection of the empirical factors (α and β) thought some more advanced techniques (i.e. utilize genetic algorithms to find the optimal value for each factor).
- Apply different heuristic based on the advancement of game:
  - utilizing a fast early-game heuristic to position the player in a optimal position when entering in the middle/late-game.
  - utilizing a more complex but slow middle/late-game heuristics.

The heuristic I would recommend to use is the custom_score_2. Its simplicity is his strength. Although it is pretty simple compared to the other two heuristics, it still achieves the best results. The main reason is that, due to its simplicity, the search tree is explored deeper (1 level deeper than custom_score and 1/2 levels deeper than custom_score_3) and on slow

computer, like the one utilized for the project, where only few levels are explored, it makes the difference. My guess on the isolation game is that the player needs to keep a decent position in the early stages of the game and try to survive the longer during the last stages. The heuristic seems to have these properties, exploring deeper the search tree with more chances to find the "longest survivor" branch.
Moreover the heuristic seems quite stable, achieving consistently good result against all the other players.