

Heuristic Analysis of Score Functions

By Simon Müller

Goal

The goal of this task is to create three different custom score function, evaluate them based on the tournament results, and choose the best one. A main difficulty of this task, is to handle the trade-of between evaluation speed of the score function and the complexity as there is a limited time frame. In detail, a fast evaluation of the score function gives us more time to search deeper in the tree and a more complex score function needs more time for its evaluation, but its result may be more accurate.

Score function 1: custom_score

The first score function is a variation of the already implemented score function from Udacity. First, the length of available moves of both players is calculated. The score is then the difference between the players score and the opponents score multiplied by the factor 3. The player needs to have 3 times more available moves than the opponent to get from that subtree a positive score.

Score function 2: custom_score2

This heuristic is a modification of score function 1, but instead of evaluating just the current move, this score function looks one step ahead and sums up the current plus the future (one step) available moves for both the player and the opponent. Then similarly as in score function 1 the score function calculates the difference between the players available moves and the opponent available moves times 3.

Score function 3: custom_score3

As custom score function 3 I used a variant of variant 1. I changed the penalty from 3 to 5 for the opponent player.

Performance

Playing Matches									

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

Win Rate:		71.4%		65.7%		70.0%		72.9%	

Custom_score3 provides the best performance of all custom my own custom score functions. It performs slightly better than the AB_improved score function (~1.5%). Custom_score and custom_score2 perform worse than the Udacity score function.

Final Recommendation

Custom_score3 function is recommended on the following bases:

- The overall winning rate in the simulation
- It is simple (linear in the number of tiles) -> fast computation (similar to custom_score)

There may be some parameter optimization necessary, but this step was not performed in this analysis.