

Stephane Chong / chong.stephane@gmail.com

We are considering the following heuristics in pseudo code. The statistics provided below are based on 4 rounds of 20 games played per player.

Heuristic 1: the number of legal moves available, #player legal moves

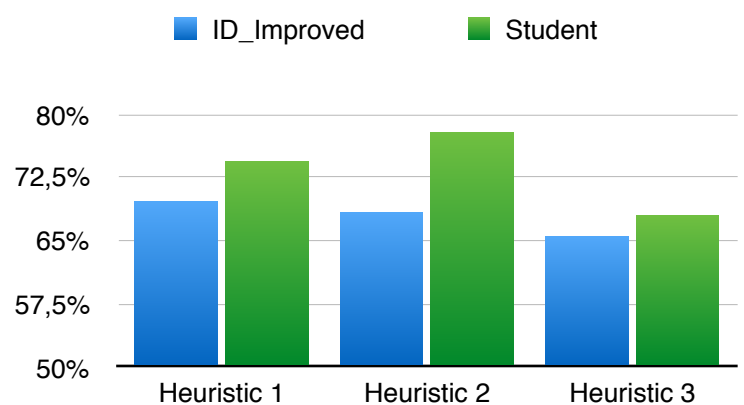
Heuristic 2: If blank tiles > game width + 3, #player's legal moves

else #player legal moves - #opponent legal moves

Heuristic 3: #player's legal moves - $1/2 * \text{euclidean_distance}(\text{player position, opponent position})$

The results are shown below for 80 games per player:

Winning Rate %	Heuristic 1	Heuristic 2	Heuristic 3
ID_Improved	69,5%	68,2%	65,5%
Student	74,3%	77,7%	67,9%



Discussion:

- The baseline results seems to be dependent on the performance of calculating the heuristic as ID_Improved performance decreases with the complexity of the heuristic. Heuristic 3 is expensive to computer and therefore performs poorly on the baseline
- Additionally, a simpler heuristic can be advantageous for both the implementation and the performance of the heuristic, in particular when the board size changes. Heuristic 2 sets a threshold for when the agent become more aggressive by taking into consideration the moves that reduces the opponent's legal moves.
- There is a slight improvement in the performance by changing the heuristic when the board is nearly complete. The heuristic is based on the insight that it may pay off to be offensive in the last few rounds, in hope that the opponent will be trapped in a losing partition of the game. Further enhancement will include testing for an optimal blank tiles threshold.
- Given that it is performing better for the Student agent while not sacrificing too much on the ID baseline as well as simple to implement and scale, I recommend to use the **heuristic 2**