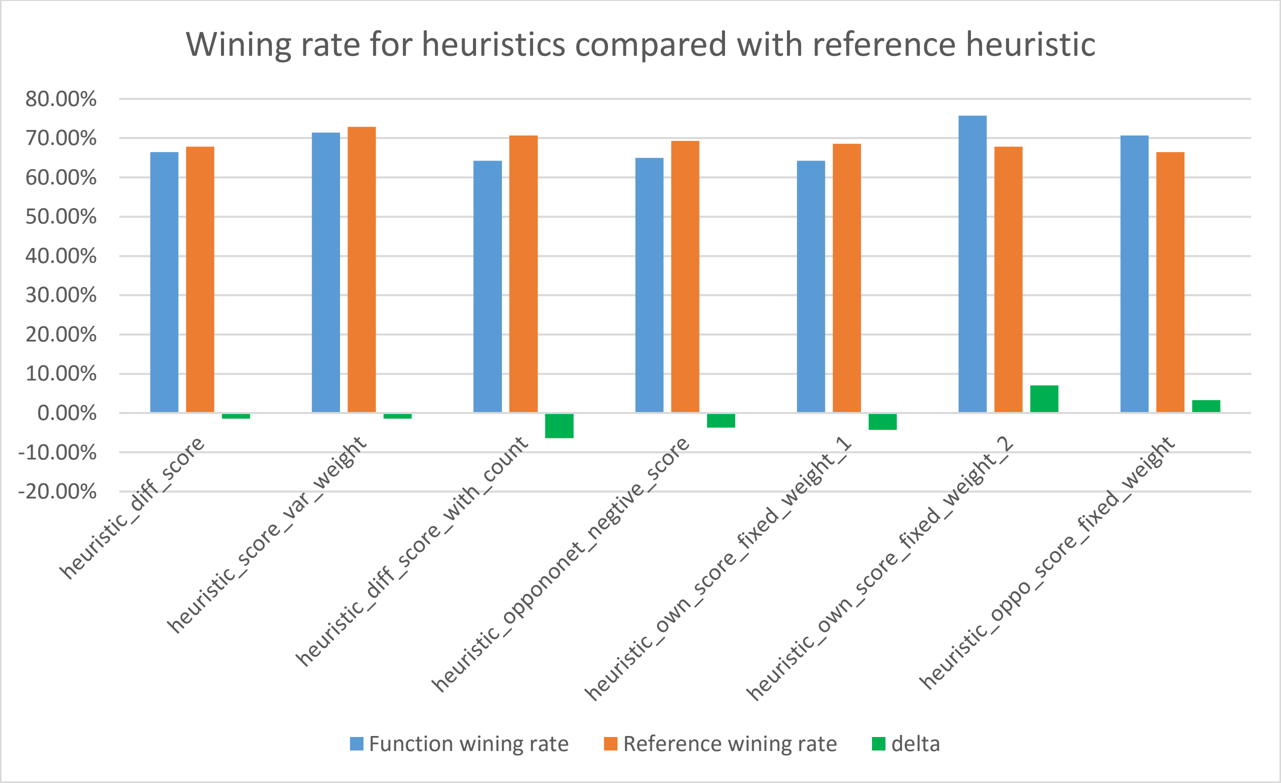
Analysis of heuristic functions in Isolation game

Totally 7 heuristic functions are tested. Listed below:

Function name	Function implementation
heuristic_diff_score	own_moves - opp_moves
heuristic_score_var_weight	own_moves/game.move_count
heuristic_diff_score_with_count	own_moves - opp_moves - game.move_count
heuristic_opponet_negtive_score	-opp_moves
heuristic_own_score_fixed_weight_1	2*own_moves - opp_moves
heuristic_own_score_fixed_weight_2	own_moves/2 - opp_moves
heuristic_oppo_score_fixed_weight	own_moves - 2*opp_moves

And the wining rate data of different heutistic functions listed below:

Function name	Function wining rate	Reference wining rate	delta
heuristic_diff_score	66.43%	67.86%	-1.43%
heuristic_score_var_weight	71.43%	72.86%	-1.43%
heuristic_diff_score_with_count	64.29%	70.71%	-6.42%
heuristic_opponet_negtive_score	65.00%	69.29%	-3.71
heuristic_own_score_fixed_weight_1	64.29%	68.57%	-4.26%
heuristic_own_score_fixed_weight_2	75.71%	67.86%	7.05%
heuristic_oppo_score_fixed_weight	70.71%	66.43%	3.28%



The first function is actually the same with the ID_improved function, but we can see the wining rate is somehow different. This means the wining rate is not stable even for the same heuristic in the same tournament. So if your function cannot over-perform the reference more than like 2 percent delta constantly, it is not a valid prof that your function is better.

heuristic_score_var_weight and heuristic_diff_score_with_count introducing the move_count variable which doesn't help and it improves the calculation time, affecting the search depth. heuristic_oppononet_negtive_score simply uses the negative score of the opponent's move but its results is better than expected. I think the reason is when the heuristic function takes place, it is actually the opponents turn to move, so my_move is actually not a correct number (opponent's move is the correct number) so will introduce some error. Moving on, let's improve the weight of my_move using heuristic_own_score_fixed_weight_1, the performance becomes worse. Then moving on to lower my_move's weight using heuristic_own_score_fixed_weight_2， the performance finally out-performs the ID_improved. With these 3 functions, we can see that know yourself and know your opponent's better(which means take opponent's with more weight) is a good strategy. Let's use another heuristic heuristic_oppo_score_fixed_weight to improve the weight of the opponent's moves , it also out performs the ID_improved, which validates our point.

Since heuristic_own_score_fixed_weight_2 which lower's my_move's weight to 0.5 has the best performance, so I choose this function in the task. From other unsatisfactory heuristics, we can learn that 1. It is not easy to out perform the ID_improved heuristic. 2. Sometimes more variables and thus more calculation is not worthy it, because it will case the search depth shallow. 3. Know yourself and know your enemy better applies to this problem, sometimes we can apply some life wisdom to solving AI issues.