

Heuristic Analysis

Following are the custom functions considered:

1. ***eval_param_score_fn(game, player,x,y,z)***

2. ***eval_normalize_score_fn(game, player)***

3. ***eval_reduce_opp_score_fn(game, player)***

1. ***eval_param_score_fn(game, player,x,y,z)*** : Parametrized evaluation function. Here x,y,z are tuning parameter.

Bench match comparison:

Student vs ID_Improved Mean winning % of ID_Improved : **72.86%**

Mean winning % of Student : **78.57%**

2. ***eval_normalize_score_fn(game, player)*** : The function evaluates based on the number of occupied space. When board positions are less than half occupied, agent weighs to reduce opponent's legal move. (Aggressive play)

Bench match comparison:

Student vs ID_Improved Mean winning % of ID_Improved **71.42%**

Mean winning % of Student : **72.14%**

3. ***eval_reduce_opp_score_fn(game, player)***: This evaluation function only based on opponent's number of legal moves.

Bench match comparison: Student vs ID_Improved Mean winning % of D_Improved **71.42%**

Mean winning % of Student : **72.86 %**

Hence the relative comparison of heuristic function result:

Mean winning % using ***eval_param_score_fn***: **78.57%**

Mean winning % using ***eval_normalize_score_fn*** : **72.14%**

Mean winning %using ***eval_reduce_opponent_score_fn*** : **72.86 %**

Therefore on above analysis results, ***eval_param_score_fn*** is the better choice compared to other two. It slightly performs better against all opponents. By tweaking the parameter, performance can be further improved and by learning using knowledge base of game results, the parameter can be optimized.

