Heuristic Analysis

Five heuristics I tried in this project.

1. Linear ratio improved score

Intuition

Less aggressive at the beginning and more aggressive move closer to the end.

Implementation

`(blank_ratio*own_moves - C*(1-blank_ratio)*opp_moves)`

2. Nonlinear ratio improved score

Intuition

Same as the above, but be aggressive more quickly.

Implementation

`blank_ratio = (1+C) / (g.width*g.height+1 - len(g.get_blank_spaces()))`
`(blank_ratio*own_moves - (1-blank_ratio)*opp_moves)`

3. Second Moves Score

Intuition

In ID_Improved, we use the number of boxes player can move, this take each move have the same utility. But this may not be true. In this heuristic, I use the number of boxes player can move to in second move. Return the difference between player and opponent.

4. Second Moves In Middle Game Score

Intuition

As above, but at the beginning of the game, there's only little difference between player and opponent, so use ID_Improved for first a few moves, and use Second Moves Score for rest of moves. Return the difference of player and opponent.

5. All Boxes Can Move Score

Intuition

As Second Moves Score, instead of only count second moves, this one count all boxes could be moved to in the board. Return the difference of player and opponent.

Program results

The result from `tournament.py` are organized in below:

	ID_Improved	Linear_Improved	Nonlinear_Improved	Second_Move	Second_In_Mid_Game	All_Boxes
Random	86/100	86/100	86/100	87/100	**92/100**	82/100
MM_Null	71/100	72/100	80/100	81/100	**84/100**	79/100
MM_Open	61/100	74/100	67/100	**76/100**	72/100	63/100
MM_Improved	55/100	68/100	69/100	**73/100**	65/100	62/100
AB_Null	70/100	70/100	74/100	84/100	**85/100**	78/100
AB_Open	64/100	61/100	60/100	**71/100**	69/100	69/100
AB_Improved	62/100	67/100	55/100	**72/100**	59/100	57/100
Total winning rate	70.00%	71.14%	70.14%	**77.71%**	75.14%	70.00%

Analysis and Recommendation

According to the total winning rates presented above, two heuristics(Second_Move and Second_Move_In_Middle_Game) perform better than the ID_improved agent, other three heuristics slightly better or equal to ID_improved agent.

Linear_Improved nearly win every round vs. ID_Improved, except with AB_Open. Noninear_Improved also perform well, except AB_Open & AB_Improved. This may indecate first-nonaggressive-then-aggressive strategy perform slightly better than just aggressive.

All_Boxes_Can_Move didn't come up to my expectations. I think it caused by the complexity of computing this heuristic. Heuristic should be static and easy to compute, for it can search more deeper in search-tree.

I recommend choosing Second Moves Score:

1. It has the best performace in these heuristics
2. It give a weighted-value to every first move, that's why it can always outperfome ID_Improved which take every move equaly.
3. It can be easy compute without to many CPU time or storages
4. About Second_Move_In_Middle_game or Third_Moves? the increase of precision can not make up for the increase of the complexity of the heuristic
5. The perfomance of these heuristics comparing themselves shows below, also Second_Moves is the best one.

	ID_Improved	Linear_Improved	Nonlinear_Improved	Second_Move	Second_In_Mid_Game	All_Boxes
ID_Improved	-	48/100	53/100	55/100	58/100	53/100
Linear_Improved	52/100	-	50/100	59/100	50/100	53/100
Nonlinear_Improved	52/100	51/100	-	62/100	57/100	39/100
Second_Move	41/100	41/100	43/100	-	48/100	38/100
Second_In_Mid_Game	48/100	40/100	48/100	48/100	-	45/100
All_Boxes	47/100	48/100	51/100	61/100	61/100	-
Total winning rate	48.00%	45.60%	49.00%	**57.00%**	54.80%	45.60%

Raw result: https://github.com/zouyu9631/Isolation-Project/blob/master/raw_results