Heuristic Analysis

Three different heuristics were implemented in order to evaluate the performance of each of them with the difference between the legal moves of the player vs the opponent. This heuristic is referred to as 'ID_Improved' throughout the analysis

This paper describes the results achieved by comparing the custom heuristics described below against the 'ID_Improved' heuristic.

Heuristic 1: The first heuristic implemented ('penalize_corners_heuristic') elaborates the idea of 'ID_Improved' by penalizing the player when the current state of the board includes locations in corners. This is done because corners decrease the number of available moves, but even more important, corners truly decrease the opportunity of winning a game in the future (the player can easily be 'trapped' in a corner). Thus is addition to calculating the difference in legals moves of the player v.s. the opponent, the player is penalized harder when its current location in the board is a corner.

'ID_Improved' v.s. 'penalize_corners_heuristic' results:

ID_Improved Evaluation		Result
ID_Improved	Random	17 to 3
ID_Improved	MM_Null	17 to 3
ID_Improved	MM_Open	14 to 6
ID_Improved	MM_Improved	13 to 7
ID_Improved	AB_Null	16 to 4
ID_Improved	AB_Open	12 to 8
ID_Improved	AB_Improved	9 to 11
ID_Improved		70.00%

Student Evaluation		Result
Student	Random	19 to 1
Student	MM_Null	15 to 5
Student	MM_Open	15 to 5
Student	MM_Improved	13 to 7
Student	AB_Null	15 to 5
Student	AB_Open	13 to 7
Student	AB_Improved	14 to 6
Student		74.29%

Heuristic 2: The second heuristic implemented ('favor_run_away_heuristic') also elaborates on 'ID_Improved', but this time it favors player moves which are farther away from its opponent. This is done by adding the Euclidian distance from player to opponent to the number of legal moves of the player. As a result, player moves which have a greater distance from the opponent's location will have a higher score and thus be favored by this heuristic.

'ID_Improved' v.s. 'favor_run_away_heuristic' results:

ID_Improved Evaluation		Result		Student Evaluation		Result
ID_Improved	Random	19 to 1		Student	Random	17 to 3
ID_Improved	MM_Null	15 to 5		Student	MM_Null	15 to 5
ID_Improved	MM_Open	12 to 8		Student	MM_Open	13 to 7
ID_Improved	MM_Improved	13 to 7		Student	MM_Improved	14 to 6
ID_Improved	AB_Null	16 to 4		Student	AB_Null	17 to 3
ID_Improved	AB_Open	11 to 9		Student	AB_Open	12 to 8
ID_Improved	AB_Improved	13 to 7		Student	AB_Improved	13 to 7
ID_Improved		70.71%		Student		72.14%

Heuristic 3: The last and best heuristic implemented ('look_ahead_heuristic') elaborates on the 'ID_Improved' heuristic as well. The main difference between the 'ID_Improved' and the 'look_ahead_heuristic' is that in addition to calculating the difference between legal moves of player v.s. the opponent, it calculates how many moves does each of those next legal moves have. As a result, this heuristic favors legal moves which have a larger number of moves in the future.

'ID_Improved' v.s. 'look_ahead_heuristic' results:

ID_Improved Evaluation		Result		Student Evaluation		Result
ID_Improved	Random	18 to 2		Student	Random	19 to 1
ID_Improved	MM_Null	16 to 4		Student	MM_Null	18 to 2
ID_Improved	MM_Open	11 to 9		Student	MM_Open	12 to 8

ID_Improved	MM_Improved	14 to 6		Student	MM_Improved	18 to 2
ID_Improved	AB_Null	14 to 6		Student	AB_Null	17 to 3

ID_Improved	AB_Open	13 to 7		Student	AB_Open	15 to 5
ID_Improved	AB_Improved	13 to 7		Student	AB_Improved	16 to 4
ID_Improved		70.71%		Student		82.14%

As a result of the previous experiments, the ‘look_ahead_heuristic’ was selected as the best heuristic implemented. Some of the reasons why the ‘look_ahead_heuristic’ performs consistently better than others are:

1. It builds on ‘ID_Improved’, which is an already good and simple heuristic
2. It is still a very simple heuristic, and thus it does not affect the ability of the algorithm to search deep in the tree
3. It takes into consideration what the set of legal moves holds available for the upcoming play