Heuristics Analysis

Three heuristics are described in this summary. Following there is a description of each heuristic and the results obtained against the opponent agents (Random, MM_Null, MM_Open ,MM_Improved ,AB_Null ,AB_Open, AB_Improved) and the challenger agent (ID_Improved). The goal is to maximize the value returned by the heuristic for the maximizing player and minimize its value for the opponent.

1) Positions closer to the center are preferred since they give more freedom to the player

This heuristic is implemented using euclidean distance based on the center of the board and the position of the player for the current board state. The return value can be mathematically represented using the following expression

$$-\sqrt{(position(x) + 1 - \frac{board.height}{2})^2 + (position(y) + 1 - \frac{board.width}{2})^2)}$$

2) Positions closer to the borders should be avoided since they give less freedom to the player

This heuristic takes the minimum distance to any of the borders of the border and tries to maximize it to use the most distant position its 4 borders. The following expression illustrates the heuristic:

$$min(pos(x)+1, board.height-(pos(x)+1), pos(y)+1, board.width-(pos(y)+1))$$

3) Finally, a combination of these two plus number of available moves is used in a finally heuristic which is matched agains the challenger agent (ID_IMPROVED). Here is the expression:

$$-\sqrt{(position(x) + 1 - \frac{board.height}{2})^2 + (position(y) + 1 - \frac{board.width}{2})^2)} + legalMoves()$$

Results

Setting: 100 games against each opponent. Results are the ones shown by the tournament.py script.

1) This heuristic gives poor results. The student agent is almost 10% below ID_Improved. All of the opponent players are using fixed depth search and the student is not able to win in the total against MM_Improved (50%) and AB_Open (48%). As expected for both test agents (ID_Improved and student) it is more difficult to win against an agent with a more elaborated heuristic (such as Improved) than versus one with a simpler expression (Null) for both search techniques (Minimax and alpha beta pruning).

Heuristic: Avoid moves closer to the edge		
Evaluating	ID_Improved	Student
Opponent	Winning rate (%)	Winning rate (%)
Random	83	77
MM_Null	78	60
MM_Open	65	64
MM_Improved	61	50
AB_Null	69	58
AB_Open	64	48
AB_Improved	59	56
Results	68.43	59

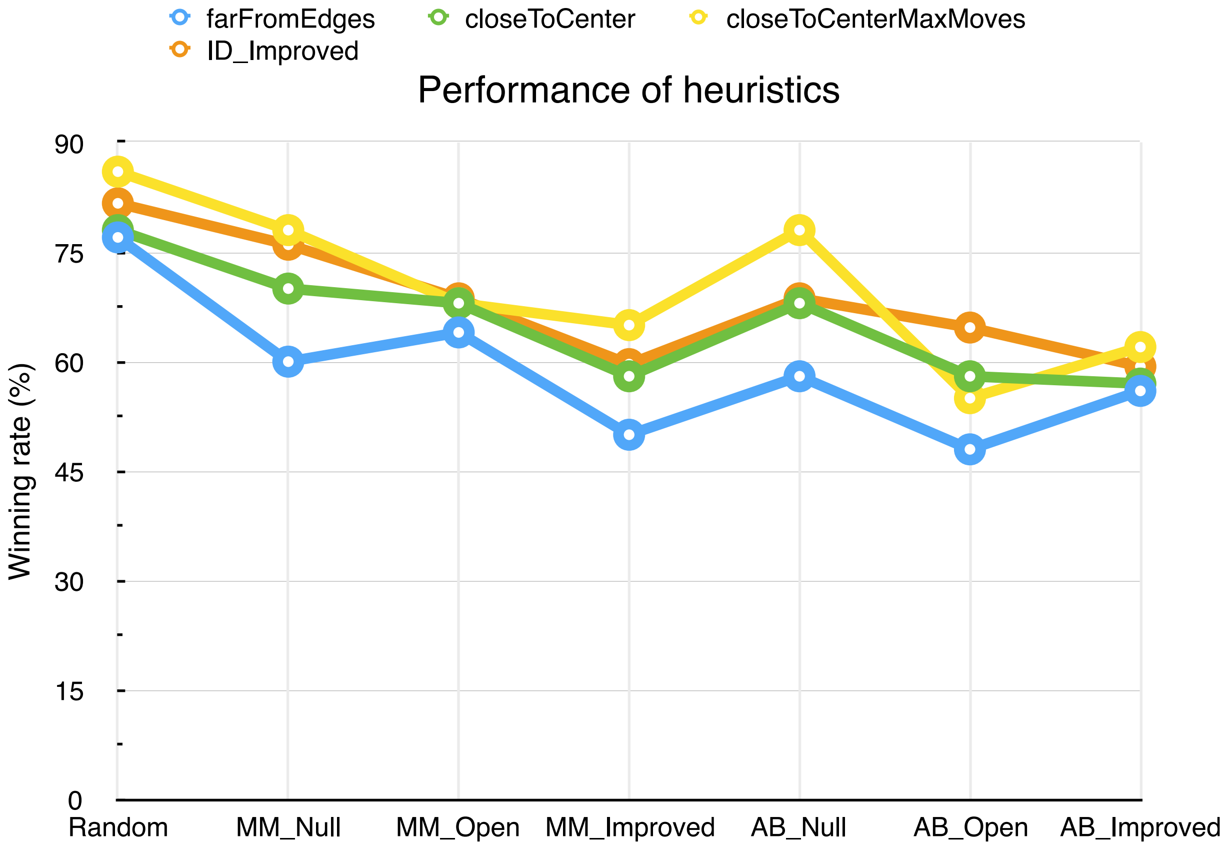
2) This heuristic offers a better match against the Improved one. However the student agent is behind by 3% approximately. Again the result versus all the opponents are as expected with the exception of ID_Improved winning with a better winning rate versus AB_OPEN (73%) than versus AB_NULL (68%). Probably a higher number of matches would allow us to see a clear difference in this regard.

Heuristic: moves closer to the center of the board are better		
Evaluating:	ID_Improved	Student
Opponent	Winning rate (%)	Winning rate (%)
Random	79	78
MM_Null	77	70
MM_Open	67	68
MM_Improved	56	58
AB_Null	68	68
AB_Open	73	58
AB_Improved	60	57
Results	68.57	65.29

3) This is the heuristic with the best results. A combination of favoring those moves closer to the center of the board and maximizing the number of allowed moves given the current state. One interesting observation is that the result of wins for ID_Improved was quite similar in all three tests, which indicates that the intention of having a hardware agnostic base reference of performance to compare with the student agent may be working. The result is clear, this heuristic beats the improved one. Random moves loses most of the matches against both test agents. And most of the times Null performs worse than Open, and Open performs worse than Improved. And since all of them are using fixed depth search, they perform worse against variants using iterative deepening.

Heuristic: moves closer to the center of the board are better and maximize number of legal moves		
Evaluating:	ID_Improved	Student
Opponent	Winning rate (%)	Winning rate (%)
Random	83	86
MM_Null	73	78
MM_Open	74	68
MM_Improved	62	65
AB_Null	69	78
AB_Open	57	55
AB_Improved	59	62
Results	68.14	70.29

The following graph compares the performance of the agent using each custom heuristic and the agent ID_Improved:



We can see that heuristic (1), defined as 'farFromEdges' in the chart performs poorly, getting a score below any other achieved by other agents against their opponents (Random, MM_Null, MM_Open, MM_Improved, AB_Null, AB_Open, AB_Improved). The agent with Heuristic (2) 'closeToCenter' performs pretty similar to the agent ID_Improved. However ID_Improved clearly beats him in the winning rate against agents Random, MM_Null and AB_Open. Finally the combination of heuristics (2) 'closeToCenter' and the maximization of the allowed moves creates the agent with the best performance. Its performance is above the one by any other agent in the graph except in the total of games against AB_Open. In the general result this agent gets an average winning rate of 70.29%, which is 2% above the one by the second best performer (average winning rate 68%), ID_Improved. This heuristic is also easy to implement since only requires to retrieve the current position, calculate the euclidean distance to the center of the board and get the allowed moves. It is also computationally time efficient since it doesn't require multi-depth calculation or nested computations, just one level below is enough. **These three reasons make the heuristic 'closeToCenterMaxMoves' the recommended strategy in this project**