# Isolation Heuristic Analysis

In this project we were tasked with creating an Adversarial Search Agent that plays the game of "Isolation", which is a deterministic game. To create this agent minimax, Alpha-Beta Pruning and iterative deepening algorithms were implemented in the project. In this project 3 heuristic scoring functions were developed their brief description is stated below:

**Custom_score:** This heuristic takes difference of percentages of total number of legal moves the player has and twice the number legal moves the opponent player has left, the scoring is done in this way so that the agent is more aggressive in its approach and chases after the opponent player. This heuristic was inspired from the video lectures

**Custom_score_2:** In this heuristic total number of legal moves left for the players and opponent is scaled by a weighing factor which sums to one. This heuristic can be hyper parameter that can define the cautiousness/ aggressiveness of the agent, weights of 0.1 and 0.9 for player and opponent adds a bit of cautiousness to the game.

**Custom_score_3:** This heuristic rewards the game playing agent if the players position is at the centre of the board, and the opponent is penalized if its position is away from the centre of the board. This centrality allows the agent to have more options to manoeuvre its position as the game progresses. The inspiration for this idea was taken from one of the video lectures.

The Snapshot below exhibits the performance of the 3 heuristics against the baseline agent "AB_Improved" taken from Tournament.py file.

```
                    *************************
                         Playing Matches
                    *************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      39  |   1    39  |   1    40  |   0    40  |   0
   2       MM_Open     29  |  11    36  |   4    36  |   4    38  |   2
   3      MM_Center    37  |   3    40  |   0    39  |   1    40  |   0
   4     MM_Improved   34  |   6    35  |   5    27  |  13    30  |  10
   5       AB_Open     21  |  19    24  |  16    22  |  18    22  |  18
   6      AB_Center    24  |  16    21  |  19    25  |  15    19  |  21
   7     AB_Improved   28  |  12    21  |  19    19  |  21    19  |  21
--------------------------------------------------------------------------
          Win Rate:      75.7%        77.1%        74.3%        74.3%
```

**Description of Heuristics Performance:**

A total of 40 matches instead of 10 matches (default choice in tournament.py) were used to evaluate performance of custom heuristics against the the AB_Improved Agent. All three heuristics had a win rate of around 75% or more with AB_custom 1 i.e. Custom_score evaluation providing the best performance.

**Recommendation:**

Custom_Score evaluation function which takes difference of percentages of total number of legal moves the player has and twice the number legal moves the opponent player has left would

be best choice out of the three heuristic functions. The reasons to recommend this heuristic is that a) it had the largest win rate among the other heuristic functions; b) It enables the agent to be more aggressive in its approach by chasing after the opponent, and c) This evaluation function also enables the agent to seek moves with the most options , while getting in the way of the opponent.