# Analysing Heuristic For Isolation game

## Heuristics

Heuristic functions are used as guide in making an informed decision when the game tree cannot be explored all the way to a terminal state. They are used to guide the game play towards the end goal. A good heuristic function would lead to moves that increase the likelihood of a win.

## Exploring Heuristics for Isolation Game

As part of this exercise several heuristic functions were explored.

1. Number of open moves for the player
2. Difference in open moves between player and opponent
3. Distance of player from center of board which has high degree of freedom
4. Difference ins distance of players to center of board
5. Position scores for the first 3 moves based on data collected from multiple runs
6. Knights Tour Heuristic (since out players move in a similar pattern)

## Simple Heuristics

Simple heuristics like number of open moves and difference in open moves lead to a consistenly above average win rate (70-80%). In some cases the win rate even touched 88%.

## Multiple Heuristics

When multiple heuristic functions were combined with weights, the win rate was not consistent. It fluctuated from a low 68% all the way to 78%. But intuitively based on the multiple runs, the performance was not an improvement over simple heuristics.

## Game Position Score Heuristics

Gathered the data on first 3 moves by running the game multiple times and analysed that to get a position score for each of the first 3 moves. When this data was used as a rule-book, the win rate did not improve because the rule-book was applied to both the opponent (CustomPlayer as well) and the player.
So moved the logic to custom_score as a heuristic to calculate the score based on the first 3 moves. To achive this, history of the moves were added to the game object.  But this did not improve the win percentage as was exhibited by multiple runs.

## Search Depth

On an average the search depth on my laptop seemed to be around 6.5 for each

turn.
Optimizing the code by removing log statements, unnecessary function calls, simplifying heuristic function did not improve the search depth.
Intuitively larger search depth with a reasonable heuristic should yield better win percentage.

There seems to be a positive correlation between win percentage and average depth explored.

**Run data**
Student 76.43%
avg search depth: 6.667979741136747

Student 75.71%
avg search depth: 6.814933333333333

Student 79.29%
avg search depth: 6.831578947368421

## Example Run:

```
*************************
  Evaluating: Student
*************************

Playing Matches:
----------
 Match 1:   Student   vs   Random      Result: 20 to 0
 Match 2:   Student   vs   MM_Null     Result: 20 to 0
 Match 3:   Student   vs   MM_Open     Result: 16 to 4
 Match 4:   Student   vs MM_Improved   Result: 15 to 5
 Match 5:   Student   vs   AB_Null     Result: 15 to 5
 Match 6:   Student   vs   AB_Open     Result: 15 to 5
 Match 7:   Student   vs AB_Improved   Result: 14 to 6


Results:
----------
Student          82.14%
```

## Reference:

1. Knight Tour Heuristic: https://support.sas.com/resources/papers/proceedings15/3060-2015.pdf

2. Artificial Intelligence for Games - Ian Millington, John Funge