

# Heuristic Analysis for Isolation Project

By

Ajay U. Makode

## 1. Introduction

In this paper, we will be analyzing different heuristic functions for isolation project and will be proposing the best heuristic function with reasons. As a part of this project we have implemented below mentioned algorithms:

1. MINMAX algorithm and
2. Alpha-Beta Pruning

For any algorithm to work properly very important thing is the input data to the algorithm. If your input data is not up to the mark then probably you may not get the desired result to which algorithm is capable of. Here the input data to above two algorithms is provided by heuristic function. So, Heuristic function becomes the key factor in isolation project.

We have created the 3 different heuristic functions based on different parameters and ran the tournament between different agents. We will be providing detailed analysis of each heuristic function in the upcoming sections.

## 2. Heuristic Analysis

I have implemented the different heuristic considering different parameters like difference in moves, difference in position, player near to wall etc. Analysis of these heuristic functions is explained in more detail in the below section.

### 2.1 Heuristic One

```
return float((my_legal_moves - 2 * opponents_legal_moves) * filled_spaces)
```

This is a very simple heuristic function based on the number of legal moves remaining for each player. This function simply returns the difference in number of legal moves left between the players. Here we are multiplying the opponents moves by 2 to reward a player with more legal moves. Here we can think of 3 different scenarios:

1. If 'player' has less than 2 times number of legal moves than opponent, then the score will be negative.
2. If 'player' has 2 times more number of legal moves as compared to opponent, score will be Positive.

3. Number of filled spaces will give the idea about the stage of the game. Initial stages, score will be treated normally. But towards the end of the game score value will matter a lot. We will make negative more negative and positive more positive. This will give the benefit who has more number of moves towards the end of the game.

#### Analysis:

This heuristic showing the best results amongst all other heuristic functions in terms of results. but during the multiple run's this heuristic is not performing good against AB players.

## 2.2 Heuristic Two

```
manhattan_distance = abs(position_of_player[0]-position_of_opponent[0]) + abs(position_of_player[1]- position_of_opponent[1])  
  
return(float(difference_in_moves/float(manhattan_distance)))
```

This heuristic is based on both difference in number of legal moves as well as position of the two players also. Here, we are using the Manhattan distance to penalize difference of number of legal moves. We will be inclined to choosing the board states in which the opponent is fairly limited in its movements.

#### Analysis:

With this heuristic we were able to see better results as compared the ID\_Improved. We can consider this heuristic as one of best heuristic. During the multiple runs, results of this heuristic were good against AB player. So, I will be considering this as the best heuristic amongst all.

## 2.3 Heuristic Three

```
return float(len(own_moves) - len(own_v_wall) - len(opp_moves) + len(opp_v_wall))
```

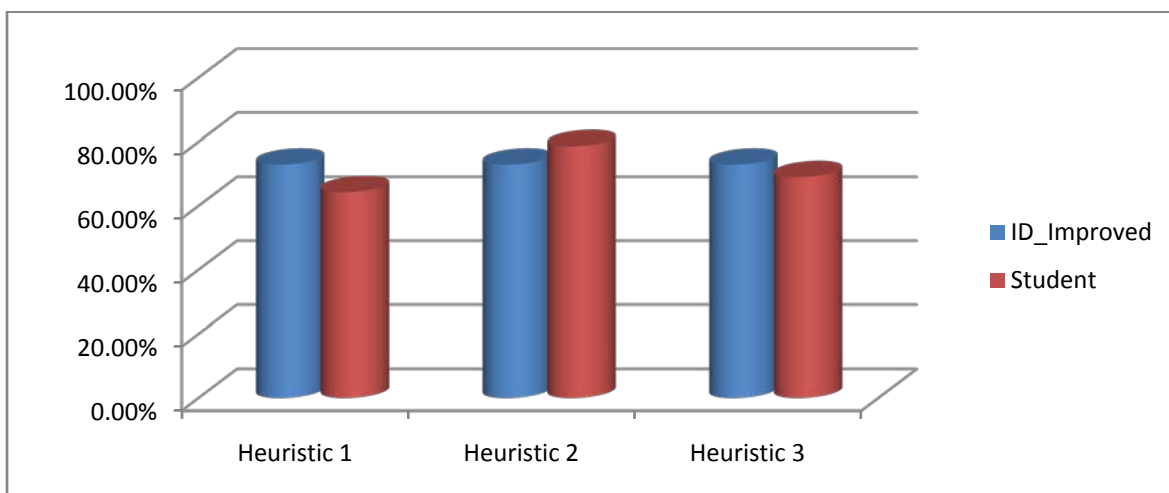
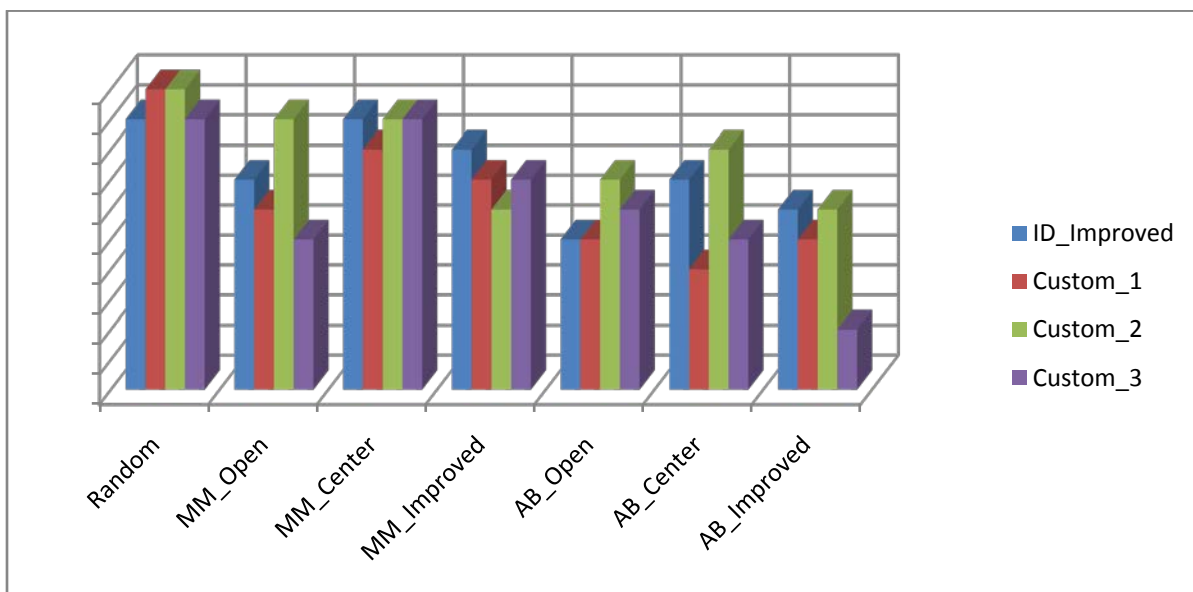
This heuristic is based on the observation that if the player is near to any of the walls of the board, the player has less chance to win. The results of this heuristic were average during the multiple runs of tournament.py.

#### Analysis:

This heuristic shown average and stable results against all of the opponents. We can consider this heuristic function as an average heuristic function.

### 3. Comparison

***** Playing Matches *****									
Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	10	0	9	1
2	MM_Open	7	3	6	4	9	1	5	5
3	MM_Center	9	1	8	2	9	1	9	1
4	MM_Improved	8	2	7	3	6	4	7	3
5	AB_Open	5	5	5	5	7	3	6	4
6	AB_Center	7	3	4	6	8	2	5	5
7	AB_Improved	6	4	5	5	6	4	2	8
Win Rate:		72.9%		64.3%		78.6%		61.4%	



If we see the results, we can see that all **custom\_2** heuristic functions are performing better than AB\_Improved with the difference of **5.7%**. Looking at the results we easily figure out that AB\_Custom\_2 is performing far better against AB opponents.

## 4. Conclusion

As per the results obtained in the tournament, I will go ahead with heuristic 2 because of the following reasons :

1. Out of all heuristics, heuristic-two was the one which performed the best against all the other heuristics including ID\_Improved.
2. Out of all heuristics, heuristic-three will go deeper because of the calculations involved in this function are less as compared to heuristic-one and heuristic-three.
3. In heuristic-two, I am just calculating manhattan distance which involves very less overhead.
4. Heuristic-two out of all the heuristics in the tournament, defeated the Id\_Improved with the largest margin of 5.7%.
5. In order for the heuristic to perform well and search to the greater depths, it is necessary that the calculations involved in evaluating a board are not computationally exhaustive. Heuristic-two seems to be incorporating the core of the game of Isolation, that is to block available move to the opponent and always choosing the board state where there is a possibility for the player to block the opponent.