**Heuristics Strategy:**

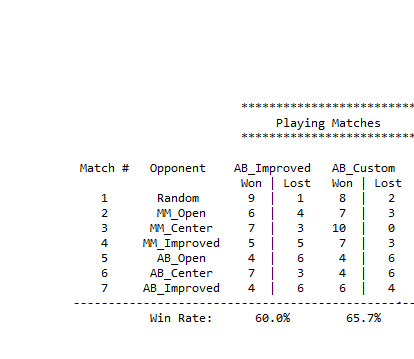
Three different heuristics have been implemented based on look ahead logic, each one is different in terms of strategy but all of them score based on the sum of available moves in the look ahead moves.

The strategies that I choose to implement are quite general in game playing, I implemented and aggressive, defensive and a balanced strategy which aims to be in the middle ground rather than being aggressive or defensive.

**Heuristic 1:**

The first heuristic calculates the difference between the sum of available look ahead moves for current player and opponent player as the score. This strategy seemed like a good one as it takes the characteristics of both aggressive agent and defensive agent into consideration unlike the second and third heuristics which are overly aggressive and defensive respectively.

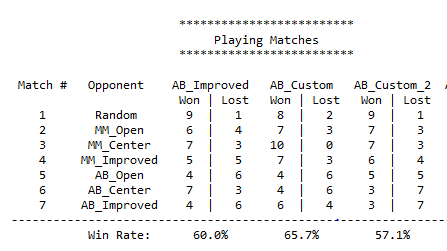
The first heuristic has higher win ratio at 67% which is 7% more than the AB\_Improved in overall performance, the first heuristic performs very well against the MiniMax MM\_Random, MM\_Open, MM\_Center and MM\_improved opponents, I was under the impression that the MM\_Improved will perform a little better than the remaining opponents because the heuristics compiled by MM\_Improved seemed a little more intelligent than the remaining opponents but it is on par with MM\_open opponent. The win ratio gradually decreases when played with AlphaBeta players, the win ratio of AB\_custom is better when played against the improved player in comparison with open and center opponents this clearly explains that the first heuristic is costly in computation and is not able to traverse a lot of depth with iterative deepening.



**Heuristic 2:**

The second heuristic is an aggressive strategy which maximizes on the sum of available moves in the look ahead moves of the current player. This strategy seems to work fairly well with 57% win ratio overall, even without considering any information about the opponent player.

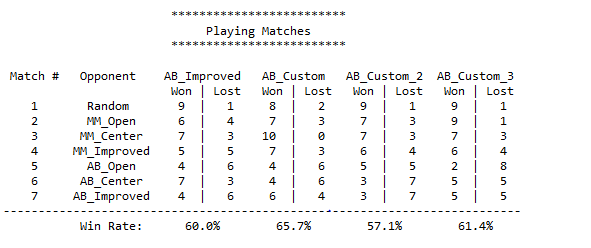
The second heuristic (AB\_Custom\_2) outperforms the AB\_improved for all the Minimax opponents and it slightly underperforms when compared to AB\_custom heuristic. The win ratio degrades as we move toward AlphaBeta Opponents especially with the AB\_Center and AB\_improved the performance is very poor because AB\_Custom\_2 is also computationally costly which limits the depth it traverses and it also does not consider any intelligence from the opponent’s moves which makes the win ratio even worse. At an overall level the win ratio of AB\_custom\_2 falls slightly short in comparison to AB\_Improved.



**Heuristic 3:**

The third heuristic is a defensive strategy which minimizes on the sum of available moves in the look ahead moves for the opponent player. This strategy is quite opposite to the second heuristic and works fairly well, the overall win ratio is at 61.4%.

The third heuristic performance is also somewhat similar to the second heuristic which is expected it starts of performing very well against the Minimax opponents in fact the performance is on par with the first heuristic and then the performance degrades for the AlphaBeta opponents for the same reasons as the second heuristic, computationally expensive which limits the depth of the search tree and not taking the current player information into consideration.



**Conclusion and Recommendations:**

From the below results we can see that the first heuristic performed very well at 65.7% win ratio followed by third heuristic at 61.4% and second heuristic at 57.1%, this is as expected because the first heuristic does not try to maximize on the current player moves or minimize on the opponent moves which are a little extreme to score, but instead computes a nice balanced heuristics by taking the difference between them.

1. I recommend the first heuristic (AB\_Custom) as the win ration in the below table suggest(65%), because it is more optimal in terms of strategy when compared to the other two heuristics as it considers intelligence from both the current player and the opponent player.
2. The first heuristic is naturally a better choice because of its complexity, it considers more than just the possible next moves. The heuristic calculates the difference between sum of all the look ahead moves of current player and opponent player, this implies it considers all the possible weights of the next two moves for the current player and opponent player given the board state.
3. The first heuristic will yield even better results if the search timeout of the alpha beta pruning is increased or when executed on a better hardware.

