Heuristic Analysis

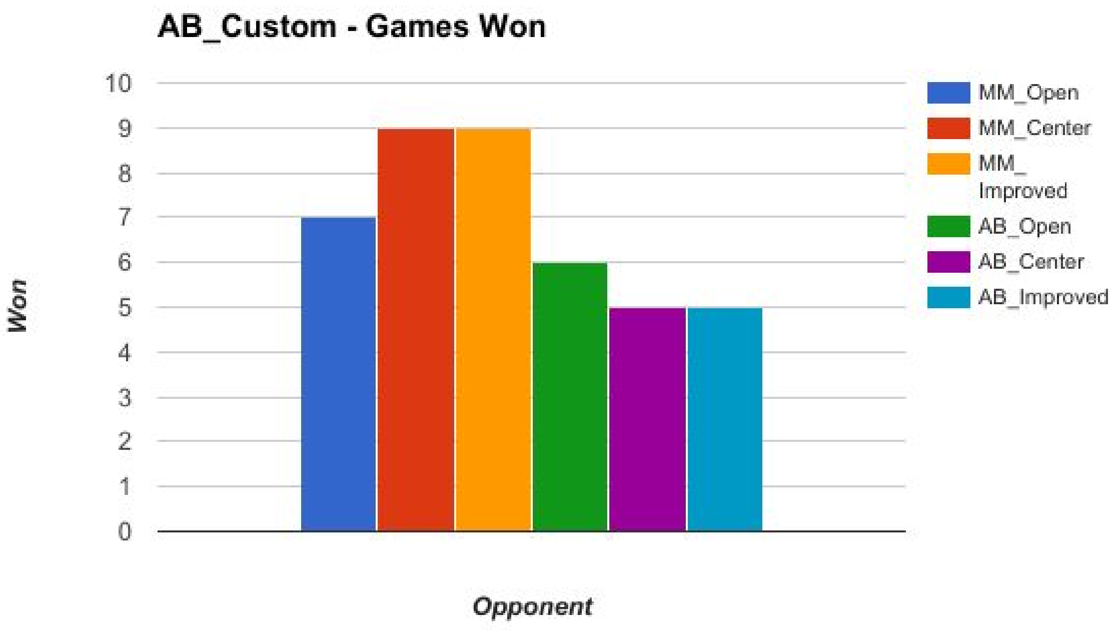
Custom Score 1

This evaluation function is based on the evaluation function mentioned in the lectures.

```
score = my_moves - opponent_moves * 2
```

This evaluation functions tries to find moves which reduces the number of opponent moves more sharp than the normal evaluation function improved_score in sample_players.py, because it multiplies the number of opponent_moves by 2.

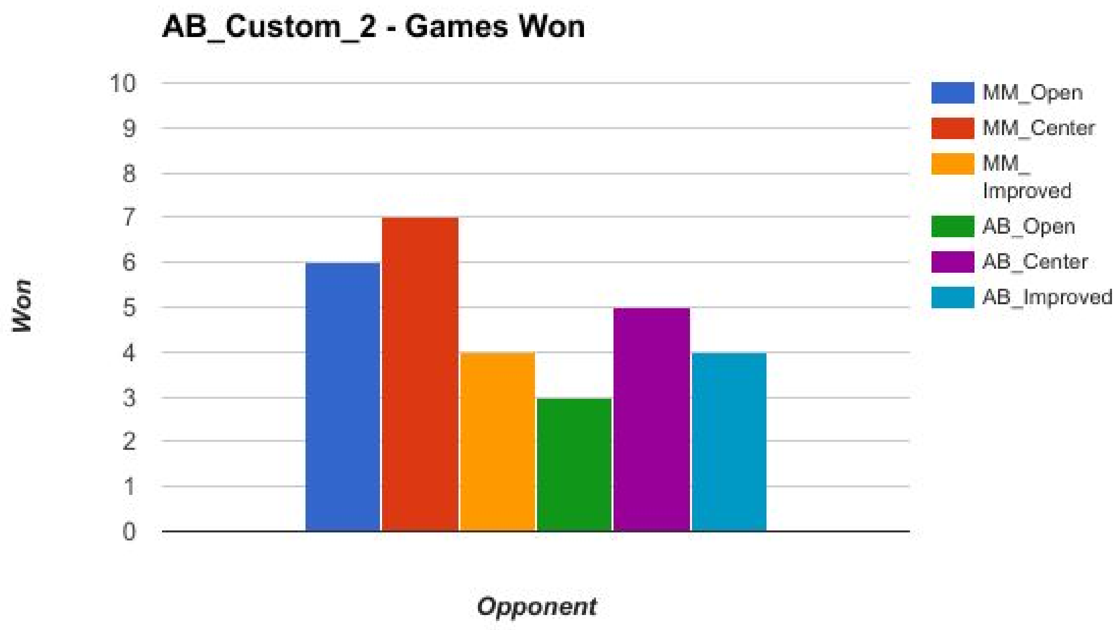
This evaluation function won against each opponent at least 50% of the time, on the weaker opponents it did perform better than against the harder opponents.



Custom Score 2

This evaluation function evaluates the score based on the fraction of my_moves against all possible moves.

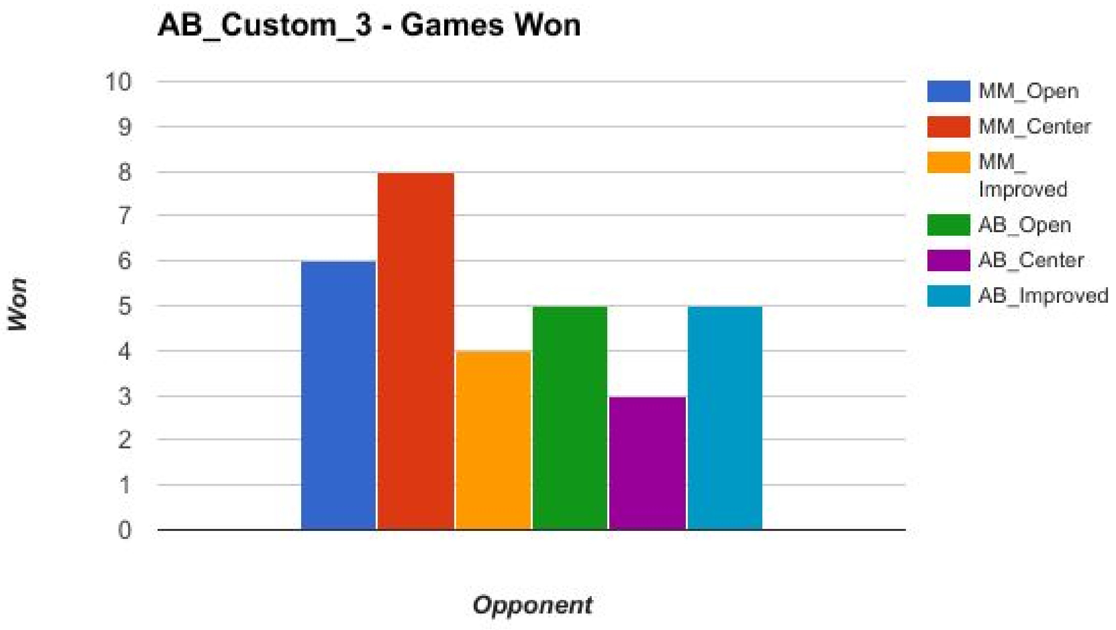
```
score = my_moves / (my_moves + opponent_moves)
```



Custom Score 3

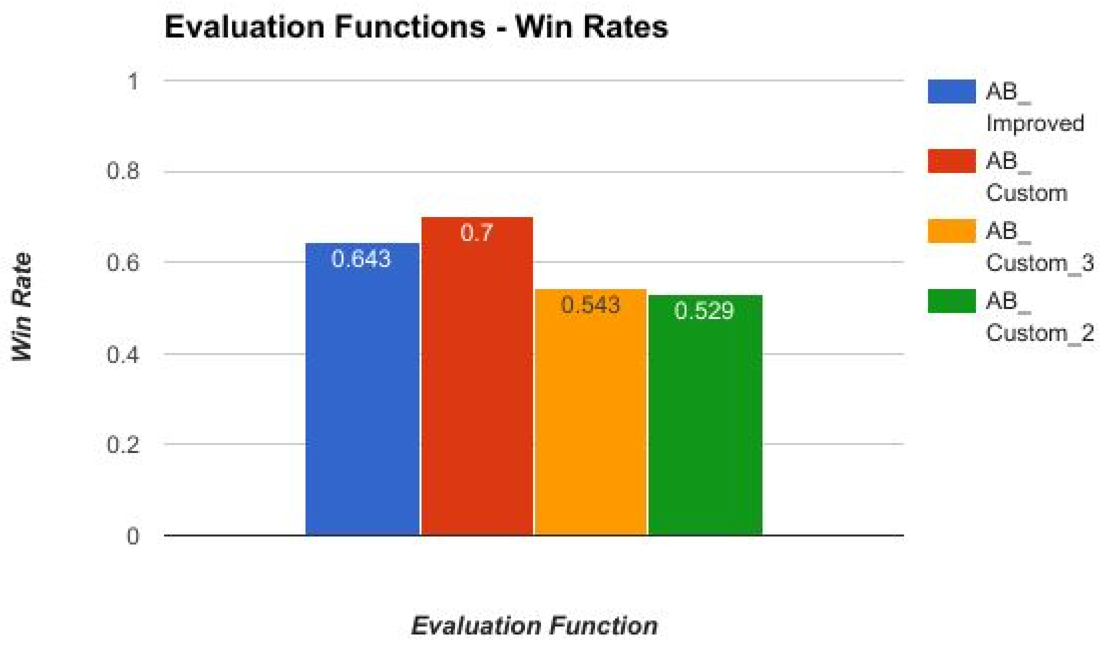
This evaluation function is based on the standard evaluation function improved_score in sample_players.py, but it ranks moves where the player is in a corner lower, and ranks moves where the opponent is in a corner higher.

```
score = my_moves - opponent_moves
if opponent in corner
     score += 1
if player in corner
     score -= 1
```



Scores

On the figure below we can see that AB_Custom was best performing, but we have to mention is this analysis that our sample size is too small that we can say that this algorithm is performing better than the others.



Recommendation

Based on the characteristics mentioned below, I would recommend to use the Custom Score 1, evaluation function.

- Largest win rate compared to the other implementations.
- Very simple implementation.
- Very fast.