HEURISTIC ANALYSIS

By Vinicius Ribeiro Machado da Silva

Objective

As the AB_improved score is defined by (#player_moves - #opponent_moves), I tried to develop some heuristics that could beat this one but playing around with the same two variables only. Below you can see the heuristics developed, the results obtained and then the critical analysis.

Heuristic 1: custom_score

```
"if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    player_moves = len(game.get_legal_moves())
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float( player_moves  -  opponent_moves*(np.random.rand()*9+1) )"
```

This score focus on using a stochastic value for the offensive strategy of pretending the higher value of the remaining opponent moves. A broad band of factors is used ranging from 1 up to 10.

Heuristic 2: custom_score_2

```
"if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    player_moves = len(game.get_legal_moves())
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float( player_moves  -  opponent_moves*5.5)"
```

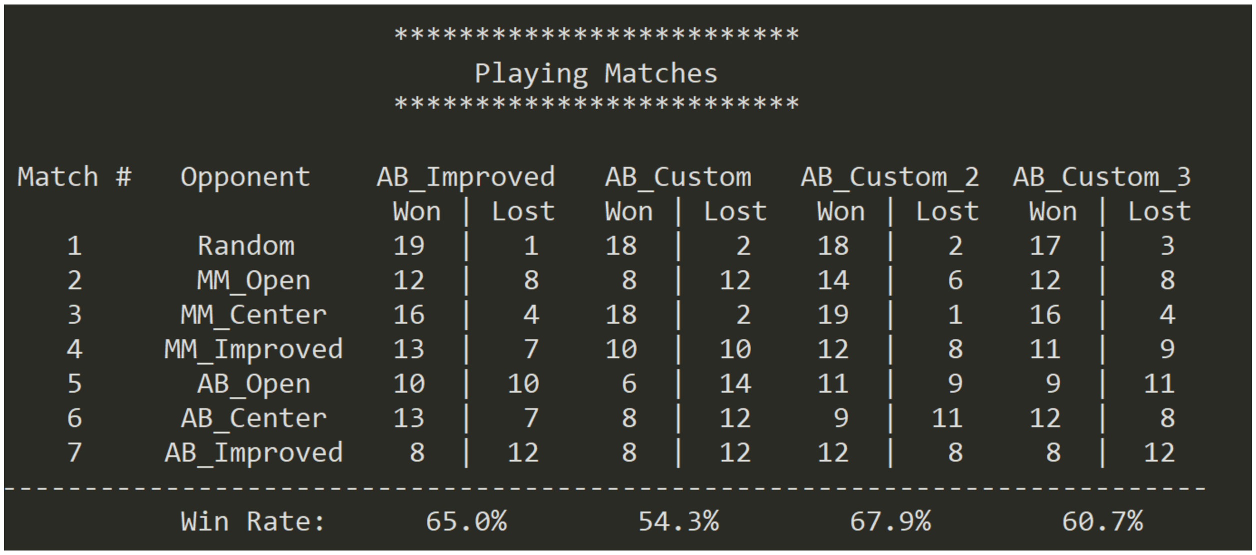
This score focus on using a deterministic value for the offensive strategy of pretending the higher value of the remaining opponent moves. One can observe that the fator of 5.5 is the mean value of the fator's distribution used in custom score (***(np.random.rand()*9+1)***) This aims to contrast with the random offensive strategy adopted before.

Heuristic 3: custom_score_3

```
"if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    player_moves = len(game.get_legal_moves())
    opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))
    if opponent_moves == 0:
        opponent_moves = 0.01
    return float( player_moves * (1/opponent_moves) )"
```

This score focus on using a penalty value for the remaining player moves based on the inverse of the number of the opponent moves. As the number of the opponent moves decrease, the number of player moves is increased, becoming more relevant for the move choice.

Results



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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      19  |   1    18  |   2    18  |   2    17  |   3
   2       MM_Open     12  |   8     8  |  12    14  |   6    12  |   8
   3      MM_Center    16  |   4    18  |   2    19  |   1    16  |   4
   4     MM_Improved   13  |   7    10  |  10    12  |   8    11  |   9
   5       AB_Open     10  |  10     6  |  14    11  |   9     9  |  11
   6      AB_Center    13  |   7     8  |  12     9  |  11    12  |   8
   7     AB_Improved    8  |  12     8  |  12    12  |   8     8  |  12
--------------------------------------------------------------------------
          Win Rate:      65.0%        54.3%        67.9%        60.7%
```

Critical Analysis

It turns out that the **custom_score_2**, which uses a deterministic offensive strategy, had a slight improvement regarding the overall results. If one applies a statistical test to evaluate whether exists a significative improvement between the mentioned score and the AB_improved, it is likely to find that the difference of the performance is not significative.

The use of a broad band random offensive strategy in custom_score seems not to provide a good choice. Alternative random strategies such as exponential decay of the randomness can also be tried, starting with a random factor and converging to a deterministic one.

In custom_score_3, a different use of #player_moves and #opponent_moves variables tried to formulate a new score. The results indicates that the difference between the mentioned variables keeps providing better results.

With that said, the recommended evaluation function is the custom_score_2, once it provides a small improvement regarding the overall results, including the AB_improved score, without increasing the computational cost. The offensive randomness idea enable the agent to be more aggressive sometimes, but further research need to be done to obtain even better results with the development of band/dispersion control and offensive factor decay.