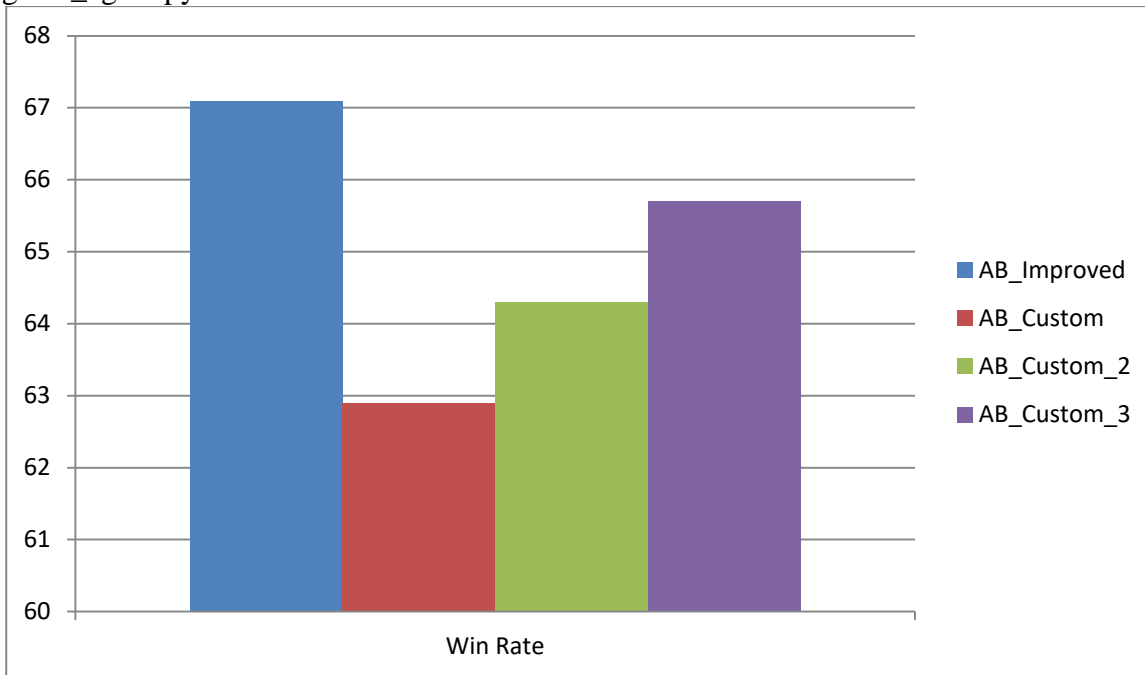


Heuristic Analysis

By Akash Chauhan

This script evaluates the performance of the custom_score evaluation function against a baseline agent using alpha-beta search and iterative deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use ID and alpha-beta search with the custom_score functions defined in game_agent.py.



Playing Matches

#	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	10	0	9	1	8	2
2	MM_Open	6	4	6	4	7	3	7	3
3	MM_Center	9	1	9	1	6	4	8	2
4	MM_Improved	7	3	5	5	6	4	7	3
5	AB_Open	7	3	5	5	6	4	5	5
6	AB_Center	6	4	6	4	8	3	5	5
8	AB_Improved	5	5	3	7	3	7	6	4
Win Rate:		67.1%		62.9%		64.3%		65.7%	

Your ID search forfeited 249.0 games while there were still legal moves available to play.

1. custom_score:

This heuristic is a combination of two heuristic function

a. center heuristic the player which is closer to the center can do better comparing to the case when he is far from the center

```
if game.is_loser(player):
    return float("-inf")

if game.is_winner(player):
    return float("inf")

# combination of two heuristics

own_position = game.get_player_location(player)
opp_position = game.get_player_location(game.get_opponent(player))

# Heuristic one : center heuristic
own_distance_from_center = math.sqrt((own_position[0] - game.width/2)**2 + (own_position[1] - game.height/2)**2)
opp_distance_from_center = math.sqrt((opp_position[0] - game.width/2)**2 + (opp_position[1] - game.height/2)**2)

if int(game.width - own_distance_from_center) > int(game.width/2):
    return float(game.width - own_distance_from_center)

# Heuristic two : left legal moves heuristic
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))

return float(own_moves - 2 * opp_moves)
```

b. left legal moves: that states the condition of player comparing to the opponents 50% more moves. Heuristic change point: when player is closer to the center than the opponent, if the player is far then the b heuristic comes into play.

2. custom_score_2:

Center Heuristic: it states that the player near to the center has better chances to win the game.

```
# Center Heuristic
own_position = game.get_player_location(player)
opp_position = game.get_player_location(game.get_opponent(player))

own_distance_from_center = math.sqrt((own_position[0] - game.width/2)**2 + (own_position[1] - game.height/2)**2)
opp_distance_from_center = math.sqrt((opp_position[0] - game.width/2)**2 + (opp_position[1] - game.height/2)**2)

return float(game.width - own_distance_from_center)
```

custom_score_3:

Legal moves left heuristic: this one is intuitive, one with the more moves available has more chances to win.

```
# Number of moves heuristic
if game.is_loser(player):
    return float("-inf")

if game.is_winner(player):
    return float("inf")

own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))

return float(own_moves - 2 * opp_moves)
```