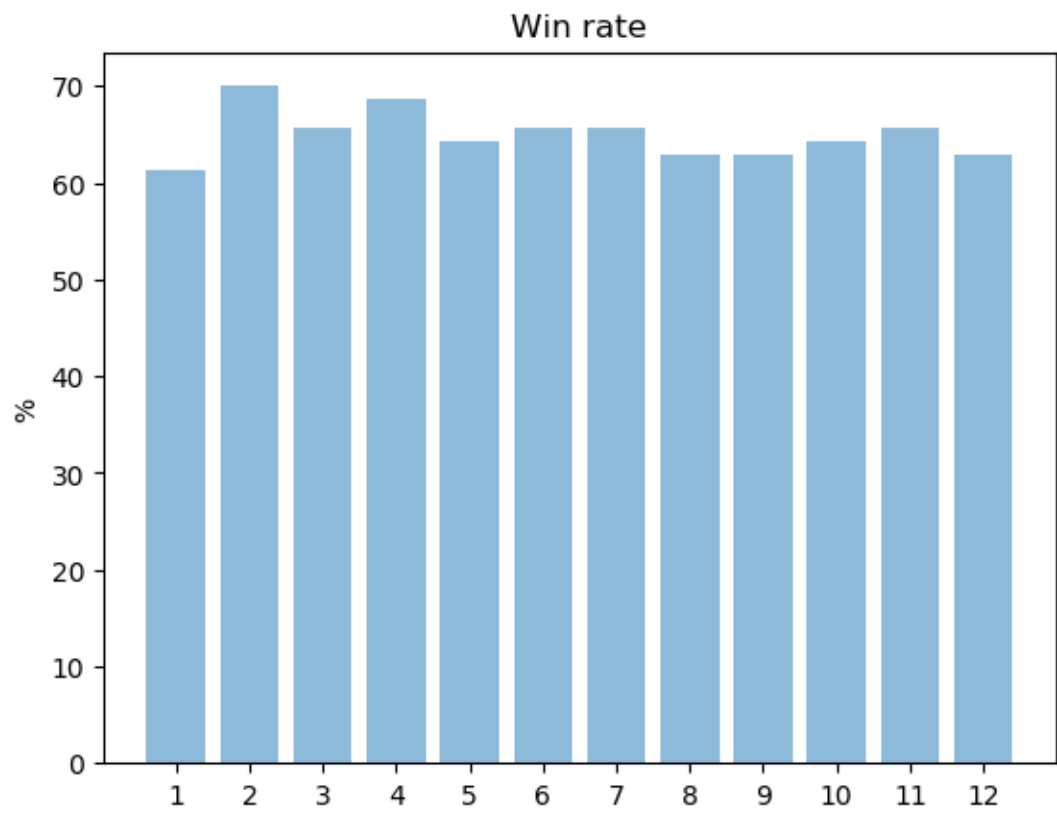


	Evaluation Function	Win Rate	Won Improved Win Rate
1	$\#My_moves - 2 * \#Opponent_moves$	61.4%	62.9% (NO)
2	$\#My_moves - \#Opponent_moves$	70.0%	62.9% (YES)
3	$\#My_moves$	65.7%	62.9% (YES)
4	$\#My_moves - 3 * \#Opponent_moves$	68.6%	64.3%(YES)
5	$2 * \#My_moves - \#Opponent_moves$	64.3%	64.3%(YES)
6	$\#My_moves^2 - 4 * \#Opponent_moves$	65.7%	64.3%(YES)
7	$4 * \#My_moves - \#Opponent_moves^2$	65.7%	64.3%(YES)
8	$\#My_moves^2 - \#Opponent_moves^2$	62.9%	64.3%(NO)
9	$\#My_moves^2 - \#Opponent_moves^3$	62.9%	64.3%(NO)
10	$\#My_moves - 10 * \#Opponent_moves$	64.3%	51.4%(YES)
11	$4 * \#My_moves^4 - \#Opponent_moves^4$	65.7%	51.4%(YES)
12	$20 * \#My_moves - 10 * \#Opponent_moves$	62.9%	51.4%(YES)

- Evaluation function { $\#My_moves - 10 * \#Opponent_mov$ } was the best evaluation function comparing it with ID_improved along with all The tournament opponents list (MM_Open, MM_Center, ..).
- Evaluation function { $20 * \#My_moves - 10 * \#Opponent_mov$ } was the only evaluation function that won the ID_improved for all of The tournament opponents list (MM_Open, MM_Center, ..).
- Evaluation function { $\#My_moves - \#Opponent_moves$ } did make the highest win rate.



```

9 def custom_score(game, player):
10     if game.is_loser(player):
11         return float("-inf")
12     if game.is_winner(player):
13         return float("inf")
14     weight = 2
15     my_moves = float(len(game.get_legal_moves(player)))
16     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
17     return my_moves - weight*opp_moves
18
19
20 def custom_score_2(game, player):
21     if game.is_loser(player):
22         return float("-inf")
23     if game.is_winner(player):
24         return float("inf")
25     my_moves = float(len(game.get_legal_moves(player)))
26     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
27     return my_moves - opp_moves
28
29
30 def custom_score_3(game, player):
31     if game.is_loser(player):
32         return float("-inf")
33     if game.is_winner(player):
34         return float("inf")
35     my_moves = float(len(game.get_legal_moves(player)))
36     return my_moves
37

```

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	8	2	9	1	6	4
2	MM_Open	6	4	6	4	8	2	9	1
3	MM_Center	5	5	7	3	8	2	8	2
4	MM_Improved	8	2	6	4	7	3	8	2
5	AB_Open	6	4	5	5	4	6	2	8
6	AB_Center	7	3	5	5	7	3	7	3
7	AB_Improved	3	7	6	4	6	4	6	4
<hr/>									
Win Rate:		62.9%		61.4%		70.0%		65.7%	

```

def custom_score(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    weight = 3
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return my_moves - weight*opp_moves

def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    weight = 2
    return weight*my_moves - opp_moves

def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    exp = 2
    return my_moves**exp - 4*opp_moves

```

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	9	1	8	2
2	MM_Open	8	2	7	3	5	5	9	1
3	MM_Center	7	3	9	1	10	0	8	2
4	MM_Improved	6	4	6	4	5	5	6	4
5	AB_Open	5	5	6	4	7	3	4	6
6	AB_Center	5	5	4	6	5	5	7	3
7	AB_Improved	5	5	6	4	4	6	4	6
<hr/>									
Win Rate:		64.3%		68.6%		64.3%		65.7%	

```

8
9 def custom_score(game, player):
10     if game.is_loser(player):
11         return float("-inf")
12     if game.is_winner(player):
13         return float("inf")
14     exp = 2
15     my_moves = float(len(game.get_legal_moves(player)))
16     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
17     return 4*my_moves - opp_moves**exp
18
19
20 def custom_score_2(game, player):
21     if game.is_loser(player):
22         return float("-inf")
23     if game.is_winner(player):
24         return float("inf")
25     my_moves = float(len(game.get_legal_moves(player)))
26     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
27     exp = 2
28     return my_moves**exp - opp_moves**exp
29
30
31 def custom_score_3(game, player):
32     if game.is_loser(player):
33         return float("-inf")
34     if game.is_winner(player):
35         return float("inf")
36     my_moves = float(len(game.get_legal_moves(player)))
37     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
38     exp1 = 2
39     exp2 = 3
40     return my_moves**exp1 - opp_moves**exp2

```

 Playing Matches

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

```

3
4
5 def custom_score(game, player):
6     if game.is_loser(player):
7         return float("-inf")
8     if game.is_winner(player):
9         return float("inf")
10
11     my_moves = float(len(game.get_legal_moves(player)))
12     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
13     return my_moves - 10*opp_moves
14
15
16 def custom_score_2(game, player):
17     if game.is_loser(player):
18         return float("-inf")
19     if game.is_winner(player):
20         return float("inf")
21
22     my_moves = float(len(game.get_legal_moves(player)))
23     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
24     exp = 4
25     return 4*my_moves**exp - opp_moves**exp
26
27
28 def custom_score_3(game, player):
29     if game.is_loser(player):
30         return float("-inf")
31     if game.is_winner(player):
32         return float("inf")
33
34     my_moves = float(len(game.get_legal_moves(player)))
35     opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
36     return 20*my_moves - 10*opp_moves
37
38

```

 Playing Matches

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	9	1	9	1	10	0
2	MM_Open	5	5	6	4	4	6	5	5
3	MM_Center	5	5	8	2	7	3	5	5
4	MM_Improved	5	5	6	4	7	3	5	5
5	AB_Open	5	5	7	3	8	2	8	2
6	AB_Center	6	4	5	5	7	3	6	4
7	AB_Improved	2	8	4	6	4	6	5	5
<hr/>									
Win Rate:		51.4%		64.3%		65.7%		62.9%	

