Heuristic performance review

Custom Heuristic 1

The first custom heuristic formula is as following:

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

w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
y2, x2 = game.get_player_location(game.get_opponent(player))

distance_to_center = float((h - y) ** 2 + (w - x) ** 2)
opp_distance_to_center = float((h - y2) ** 2 + (w - x2) ** 2)
return float((own_moves - opp_moves) + (opp_distance_to_center - distance_to_center))
```

The goal for of this formula is to combine the two heuristic “improved_score” and “center_score” together. But after run the tournament for a while, the result is not meet my expectation.

Match #	Opponent	AB_Improved		AB_Custom	
		Won	Lost	Won	Lost
1	Random	10	0	10	0
2	MM_Open	10	0	9	1
3	MM_Center	10	0	8	2
4	MM_Improved	7	3	7	3
5	AB_Open	5	5	6	4
6	AB_Center	7	3	7	3
7	AB_Improved	7	3	4	6
	Win Rate:	80.0%		72.9%	

We can see the first of my custom heuristic win rate is 72.9% that is a little bit less than AB_improved. But it still beat the most of the other heuristic except AB_improved.

Custom Heuristic 2

The second custom heuristic formula is as following:

```
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float((own_moves - opp_moves)**2)
```

The goal of this formula is to subtract from number of my moves to opp moves with square increasing. The performance is the worst one.

Match #	Opponent	AB_Improved		AB_Custom_2	
		Won	Lost	Won	Lost
1	Random	10	0	10	0
2	MM_Open	10	0	5	5
3	MM_Center	10	0	10	0
4	MM_Improved	7	3	7	3
5	AB_Open	5	5	4	6
6	AB_Center	7	3	6	4
7	AB_Improved	7	3	5	5
	Win Rate:	80.0%		67.1%	

The second custom heuristic win rate is 67.1%, which is the worst one. It lose two match with AB_Open and AB_Improved.

Custom Heuristic 3

The third custom heuristic formula is as following:

```
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float(1.7 * own_moves - opp_moves)
```

The goal for this formula is multiple certain weight to the number of my move.

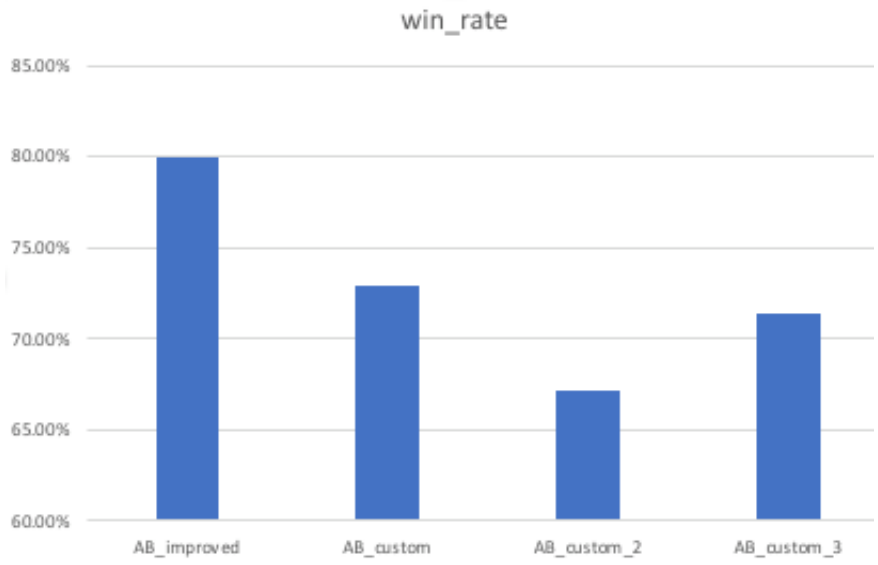
Match #	Opponent	AB_Improved		AB_Custom_3	
		Won	Lost	Won	Lost
1	Random	10	0	9	1
2	MM_Open	10	0	9	1
3	MM_Center	10	0	10	0
4	MM_Improved	7	3	5	5
5	AB_Open	5	5	6	4
6	AB_Center	7	3	4	6
7	AB_Improved	7	3	7	3
	Win Rate:	80.0%		71.4%	

The third custom heuristic win rate is 71.4%. It is a little be worse than the first one. But still much better than second one. I surprised that this heuristic win the AB_Improved but lose the AB_Center.

Conclusion

According to the performance report for each of the heuristic function. We recommend to use the AB_improved function. The reason is as following:

1. The overall performance is the highest one and it reach 80% win rate.



2. 100 % win rate for min max function. Completely surpass the old method so much.
3. The worst match win rate is equal comparing with each of AlphaBeta pruning.

Finally, according the all performance report, the simplest one is the best one.