Heuristic Analysis

The basic score functions I come up is:
1. The number of legal move of the player, which reflect how well my situation is
2. The negative value of the number of legal move of the opponent player, which will be higher when the opponent's situation is bad
3. The number of legal move of the player minus the number of legal move of the opponent player, which is the combination of above two

And I also come up some variants of these basic functions:
1. I want to take the information of remaining open spaces on the board into account, so I use this number as the scaling factor, divide the basic functions by this number.
2. I also notice that if I remove the *if* statements of setting the score of win-game as *inf* and lose-game as *-inf*, sometimes the result will be better. So, I want to see whether removing the if statements will give us better result on all the three.

Test result:
Without scaling and without if statement

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	7	3	10	0	10	0
2	MM_Open	7	3	6	4	5	5	7	3
3	MM_Center	7	3	9	1	6	4	10	0
4	MM_Improved	6	4	5	5	4	6	8	2
5	AB_Open	7	3	5	5	4	6	5	5
6	AB_Center	6	4	3	7	5	5	2	8
7	AB_Improved	5	5	5	5	4	6	4	6
	Win Rate:	64.3%		57.1%		54.3%		65.7%	

Custom 3 can beat the sample AB_Imporved w.r.t the total win rate. It performs very good against random player and minimax player, but not as good as the AB_Improved against alpha-beta pruning player.

With scaling and without if statement:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	8	2	8	2	10	0
2	MM_Open	6	4	5	5	4	6	7	3
3	MM_Center	7	3	6	4	7	3	10	0
4	MM_Improved	8	2	6	4	7	3	6	4
5	AB_Open	5	5	2	8	3	7	5	5
6	AB_Center	6	4	5	5	4	6	8	2
7	AB_Improved	4	6	5	5	3	7	5	5
	Win Rate:	62.9%		52.9%		51.4%		72.9%	

Again custom 3 can beat the sample AB_Imporved w.r.t the total win rate. It performs very likely as the previous one, good at playing with random and minimax player and not so good as alpha-beta player, but with a much higher win rate.

With scaling and with if statement:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	7	3	9	1	10	0	8	2
2	MM_Open	8	2	6	4	6	4	4	6
3	MM_Center	9	1	10	0	6	4	8	2
4	MM_Improved	4	6	6	4	6	4	6	4
5	AB_Open	5	5	6	4	4	6	6	4
6	AB_Center	5	5	4	6	6	4	5	5
7	AB_Improved	5	5	4	6	3	7	3	7
	Win Rate:	61.4%		64.3%		58.6%		57.1%	

Custom 1 beat the sample AB_Imporved, with quite same advantage over random, minimax and alpha-beta opponents.

Thus, we choose our finalized three custom function as:
Best: The number of legal move of the player minus the number of legal move of the opponent player, with scaling and without if statement
Reason:

1. The tournament performance is the best among all
2. It only require two get_legal_moves() operations and one get_blank_spaces() operation, for the first two each has a list comprehension of 8 iterations, and the third has a list comprehension of 7*7=49 iterations, which are not computational expensive
3. The score function take the information of both self-player and opponent player into account, is a more comprehensive expression of the board situation

Two alternatives:
Custom 2: The negative value of the number of legal move of the opponent player, with scaling and without if statement
Custom 3: The number of legal move of the player, with scaling and with if statement

Final result:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	10	0	9	1	7	3
2	MM_Open	7	3	8	2	6	4	6	4
3	MM_Center	8	2	9	1	8	2	7	3
4	MM_Improved	6	4	7	3	6	4	7	3
5	AB_Open	4	6	5	5	4	6	6	4
6	AB_Center	8	2	5	5	5	5	7	3
7	AB_Improved	6	4	4	6	4	6	6	4
	Win Rate:	67.1%		68.6%		60.0%		65.7%	