

Heuristic 1:

This is an adaptation of the heuristic which checks for the difference between the legal moves of the two players. Further fine tuning is done to accommodate the following:

1. Game Count at any Point – game.move_count
2. Legal moves available at the given time for player and opponent (a, b)
3. Weight based on Game Count:

Formula:

Game Count (max) – game.move_count

Available Legal moves for Player 1 = a

Available Legal moves for Player 2 = b

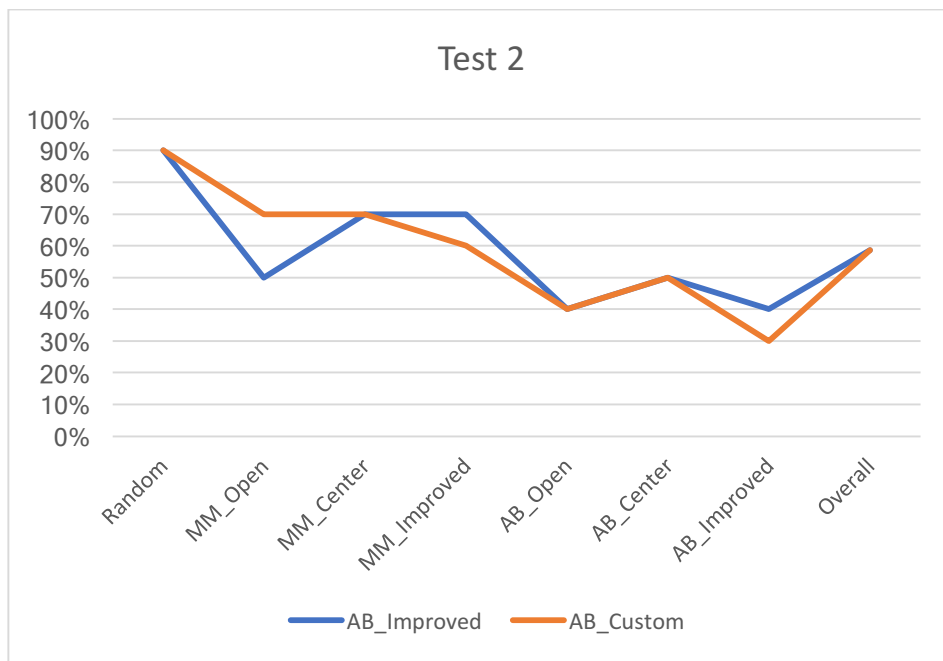
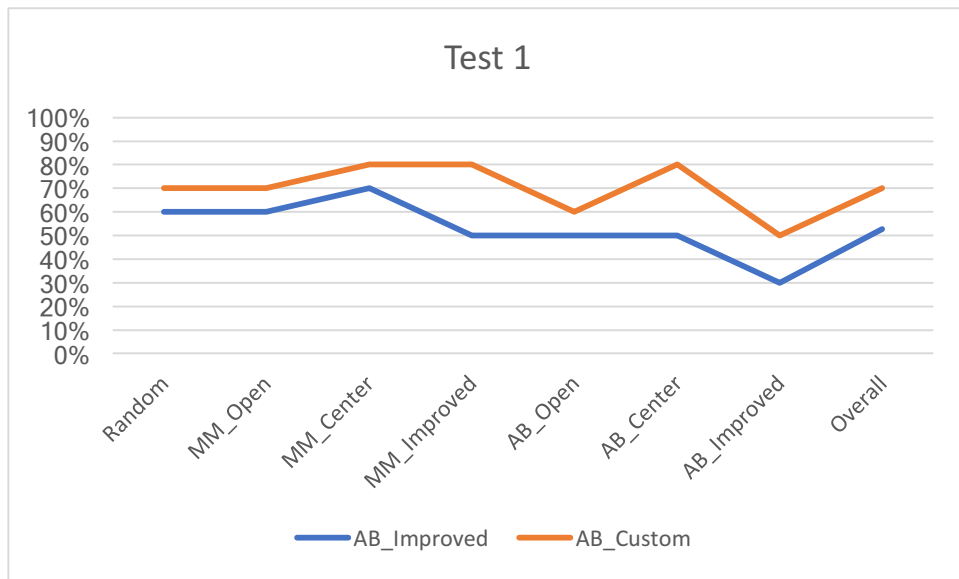
Weight based on move_count: $w = 10 / (\max + 1)$

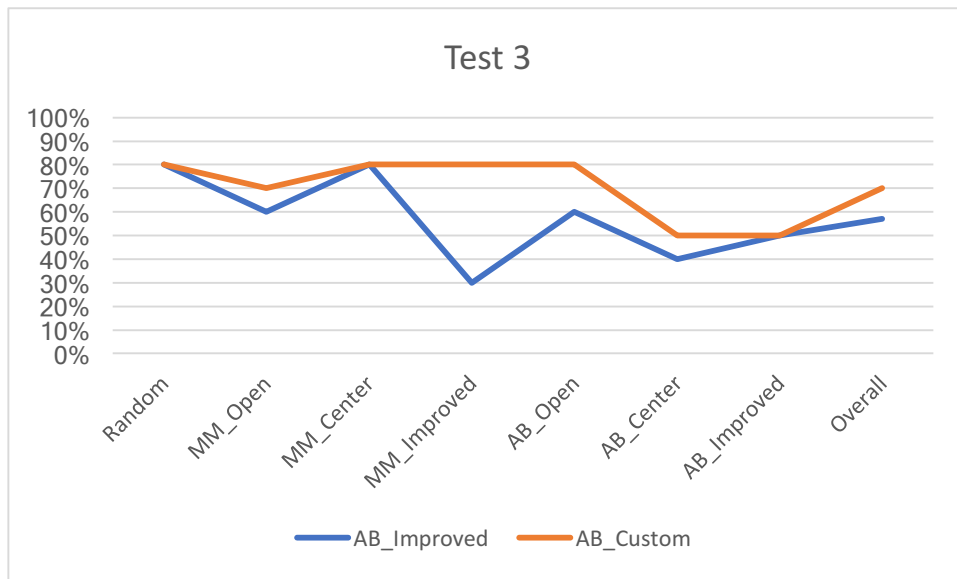
Difference = $a - w * b$

The idea is penalize the player at the earlier stages when max – is high so that the player is defensive and selects the right moves.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	6	4	7	3	9	1	6	4
2	MM_Open	6	4	7	3	7	3	8	2
3	MM_Center	7	3	8	2	5	5	8	2
4	MM_Improved	5	5	8	2	6	4	6	4
5	AB_Open	5	5	6	4	8	2	4	6
6	AB_Center	5	5	8	2	6	4	4	6
7	AB_Improved	3	7	5	5	6	4	4	6
Win Rate:		52.9%		70.0%		67.1%		57.1%	

This heuristic performs the same or better than AB_Improved.





Heuristic 2:

This is a heuristic which checks if the player is near the wall of the isolation game and is given a score as per the same. The player is penalized if near the wall and given a higher score if not near the wall.

1. Total Spaces available in the start of the game
2. Blank Spaces available at any point in the game
3. Co-efficient or a weight that is computed on the difference between the total space and available blank space. Higher co-efficient at the start and lower co-efficient towards the walls of the game.
4. Score calculated to penalize the player if near the wall

Formula:

Maximum Space at Game Start (max_space) = $\text{game.width} * \text{game.height}$

Blank Space at any point (blank_space) = $\text{len}(\text{game.get_blank_spaces}())$

Weight based on above (coefficient) = $\text{float}(\text{max_space} - \text{blank_spaces}) / \text{float}(\text{max_space})$

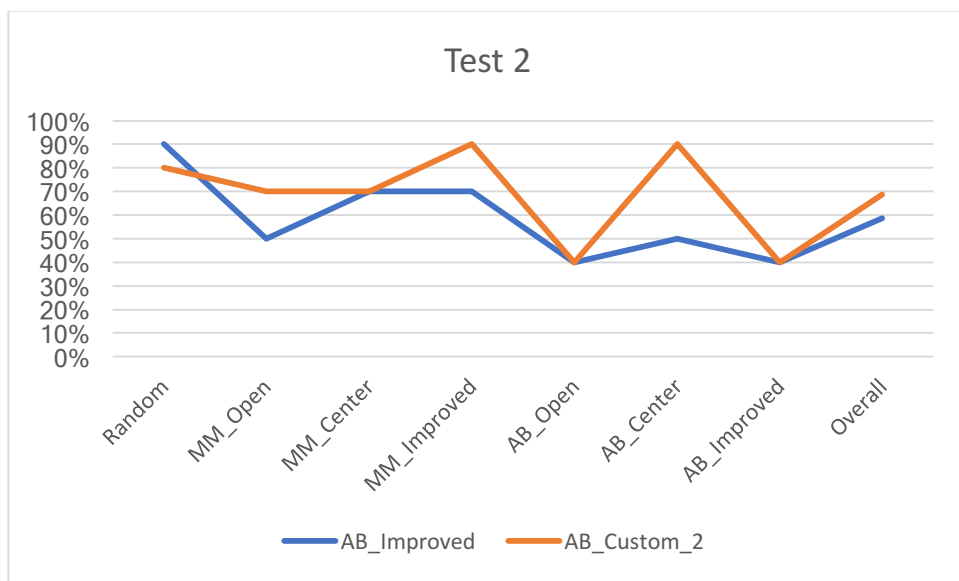
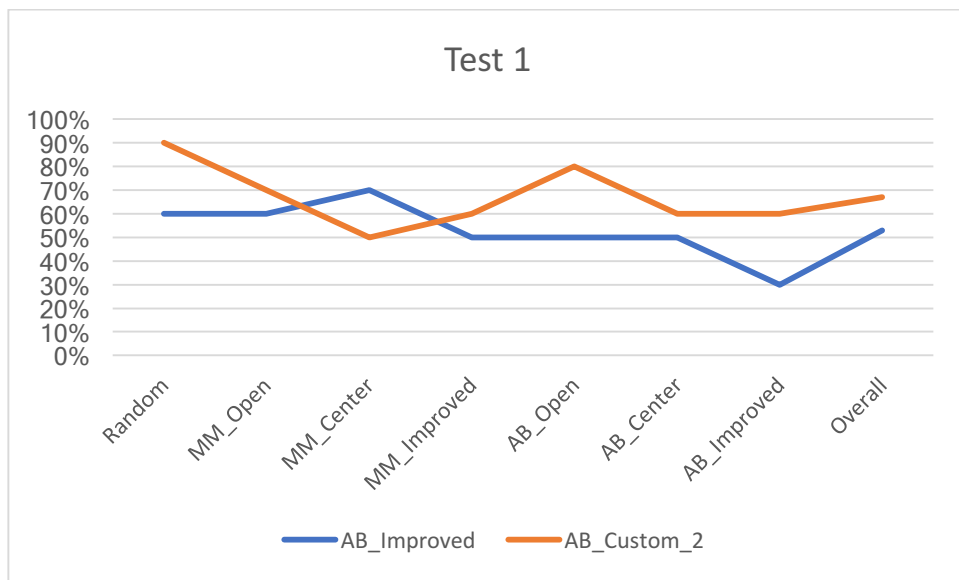
Available Legal moves for Player 1 = a

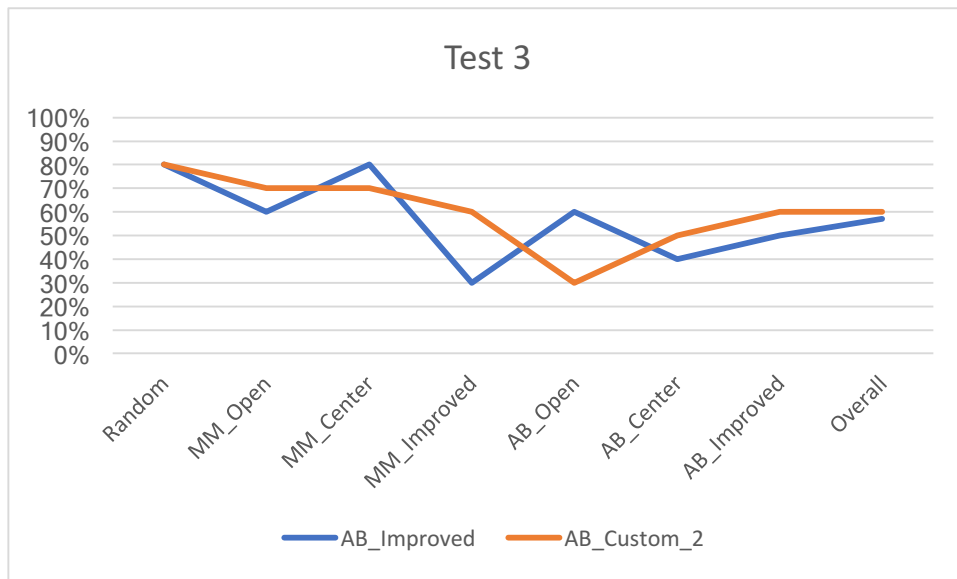
Available Legal moves for Player 2 = b

Score calculated on basis of co-efficient and if player is near walls of the game (score) = $1 - \text{co-efficient} * 1$ (if player is near wall) or $1 - \text{co-efficient} * 0$ if player is not near walls of the game.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	8	2	10	0
2	MM_Open	5	5	7	3	7	3	4	6
3	MM_Center	7	3	7	3	7	3	8	2
4	MM_Improved	7	3	6	4	9	1	6	4
5	AB_Open	4	6	4	6	4	6	5	5
6	AB_Center	5	5	5	5	9	1	4	6
7	AB_Improved	4	6	3	7	4	6	7	3
Win Rate:		58.6%		58.6%		68.6%		62.9%	

This heuristic performs most times better than AB_Improved.





This heuristic performs most times better than AB_Improved.

Heuristic 3:

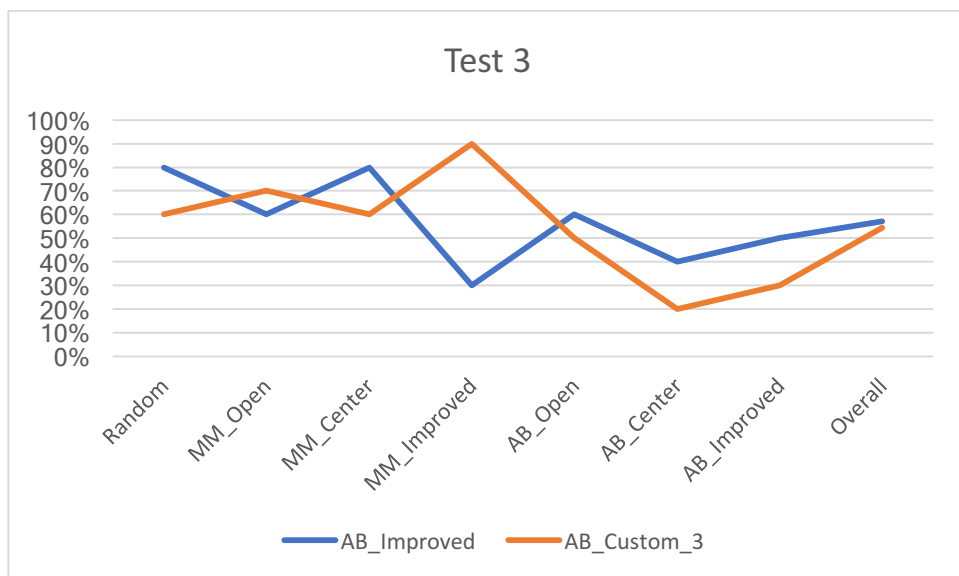
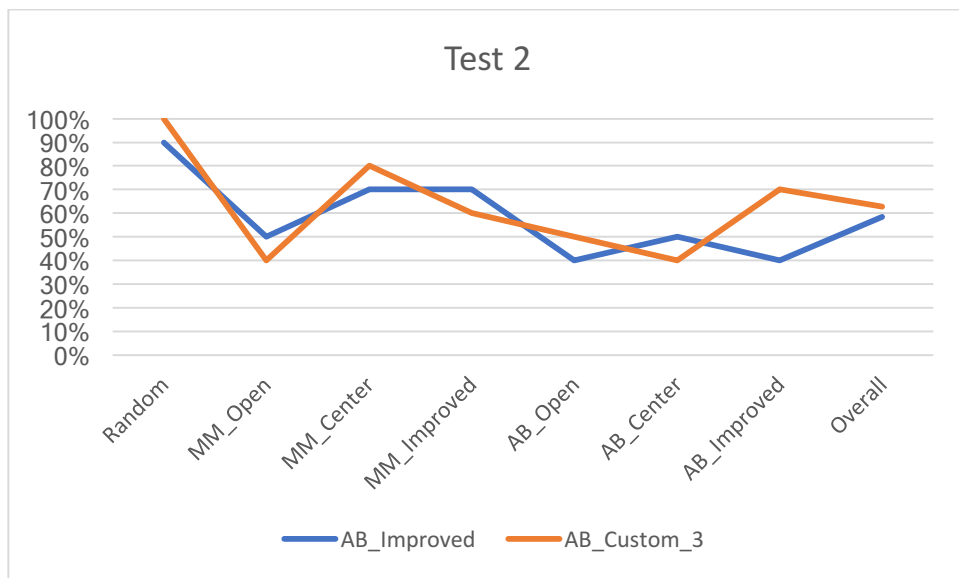
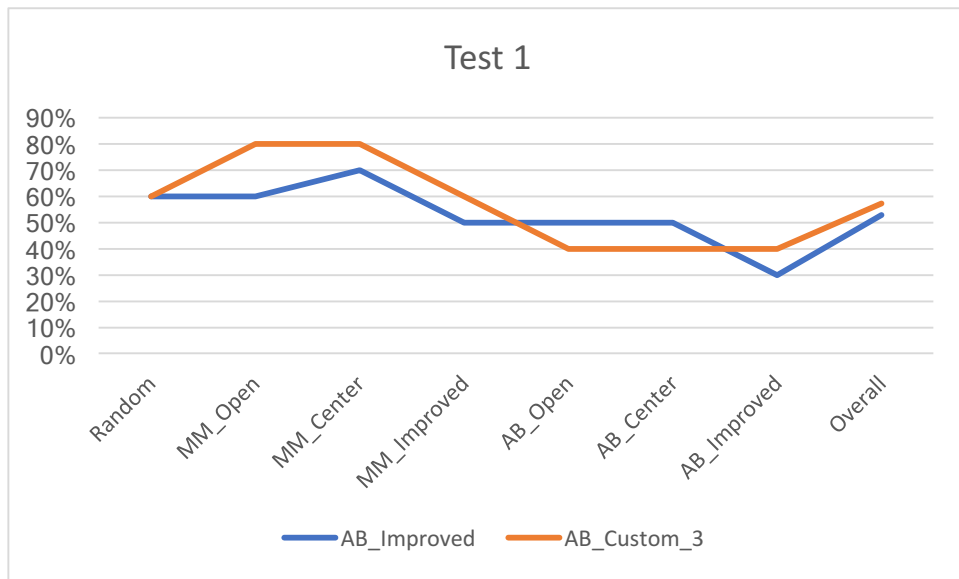
This is an adaptation of the null score heuristic in sample_players.py. Here we also assume no knowledge of terminal states and return a random number in the range (-10, 10).

Formula:

Return a random number in the range(-10,10)

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	6	4
2	MM_Open	6	4	7	3	7	3	7	3
3	MM_Center	8	2	8	2	7	3	6	4
4	MM_Improved	3	7	8	2	6	4	9	1
5	AB_Open	6	4	8	2	3	7	5	5
6	AB_Center	4	6	5	5	5	5	2	8
7	AB_Improved	5	5	5	5	6	4	3	7
Win Rate:		57.1%		70.0%		60.0%		54.3%	

This heuristic performs only some times better than AB_Improved.



Recommendation:

Based on the performance, Heuristic 2 (custom_score_2) is the recommended heuristic. This heuristic is recommended because of the following

1. Win rate – Win rate is the highest among all the heuristics
2. This penalizes the player for being confined to a side or to the wall. In some ways it helps the player to avoid being partitioned out.
3. This heuristic is quite simple to understand and is implemented quite easily. It is also something that does not take a lot of time to process and is quite fast.
4. It rewards players for being aggressive and penalizes them for being defensive.