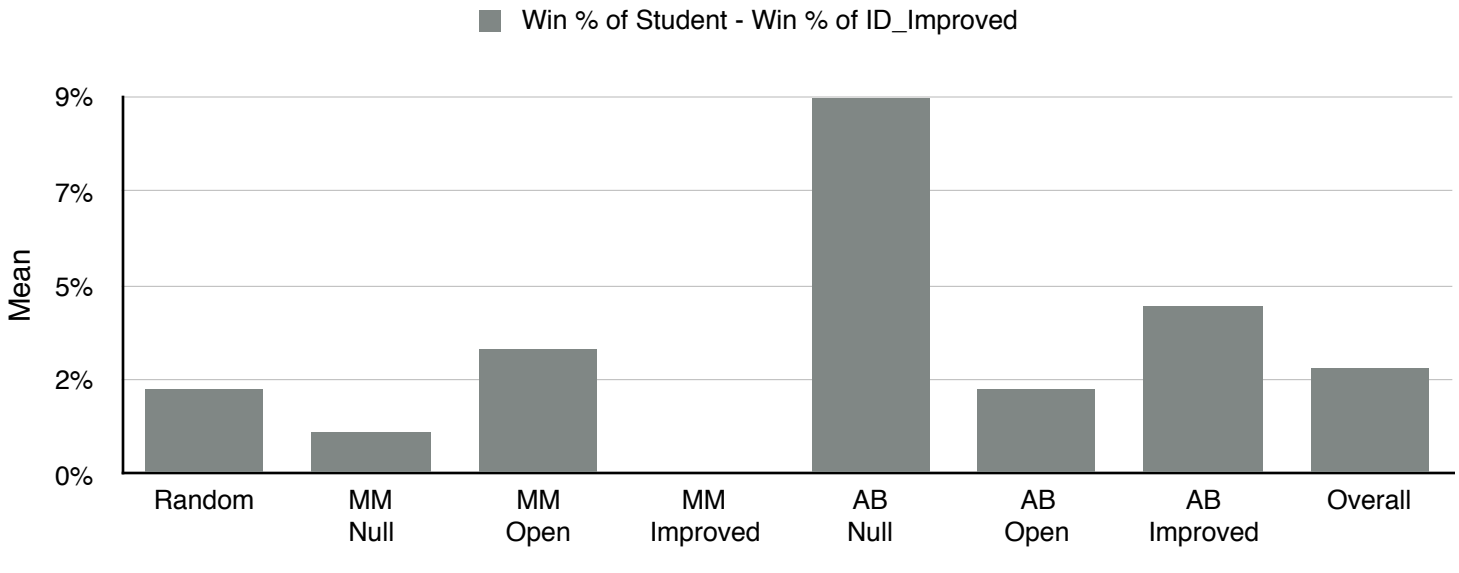
Heuristic Analysis

Explanation of analysis for each heuristic:
For each heuristic I ran the tournament 5 times, as I found that results vary with each run and wanted to get an average value instead of just running it once. For each tournament run I calculated the win rate % of "Student" minus win rate % of the "ID_Improved," as essentially we are interested in performance with respect to base level. These values are listed in the tables that follow. The average value of this metric, win rate % of "Student" minus win rate % of the "ID_Improved" over 5 rans for each match and the overall value is plotted in the bar chats that follow.

Heuristic 1:
This heuristic is a simple addition to the base line heuristic. Instead of considering legal moves of player and opponent in 1:1 ratio when taking the differences we can specify the ratio. This is based on the intuition that at least in the beginning of the game you want to move to places where you have lot more legal moves than your opponent, in other words you are aggressively chasing the opponent. We can control how aggressive we want to be by adjusting the ratio number variable, one I used was 1.5. Ratio number between 1.5 to 2.5 work nicely, others I have found to be too aggressive or just not worth it as most of the moves will relatively be the same. As you see from results below on average the heuristic improves the base line result by 3% points.

Win % of Student - Win % of ID_Improved

	Random	MM Null	MM Open	MM Improved	AB Null	AB Open	AB Improved	Overall
Run1	0%	-15%	-5%	0%	15%	15%	5%	2%
Run2	5%	20%	-5%	-5%	10%	0%	5%	4%
Run3	0%	-5%	0%	-5%	-5%	5%	10%	0%
Run4	10%	-5%	10%	0%	15%	-20%	10%	3%
Run5	-5%	10%	15%	10%	-5%	10%	-10%	4%
Mean	**2%**	**1%**	**3%**	**0%**	**6%**	**2%**	**4%**	**3%**

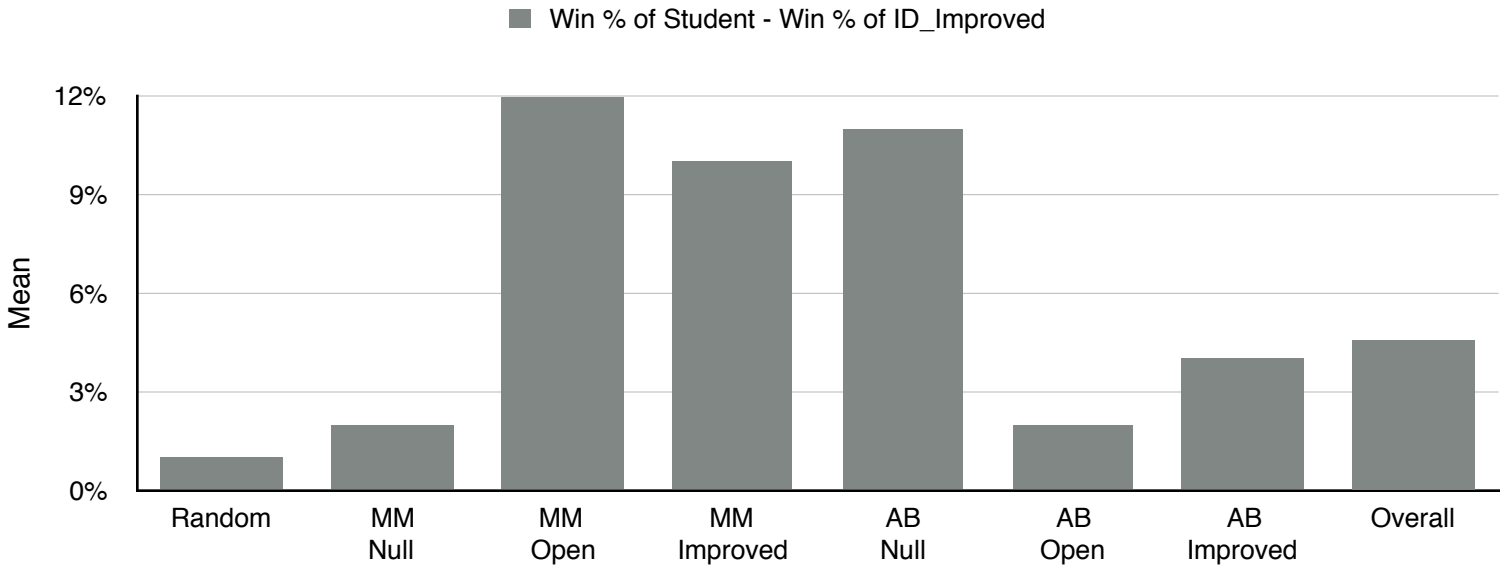


Heuristic 2:

Generally speaking it may not be useful to move into corners or walls when you have inside blocks left, as corners and walls restrict number of potential next moves. So I wanted to weight legal moves to corners and wall lower than other places while counting the number of legal moves. Obviously I weighted corners lower than walls, as corners are more restrictive. This will slightly take more time to compute as we are looping through each legal move and checking if the move is to a wall or a corner. Therefore I did not want to do this in the initial phase of the game where there are many possible moves. Once we get to certain occupancy in the game board (i.e middler of the game). I started this heuristic, in initial phase it is using heuristic 1. I tested few values for selecting at which stage we want to start the scheme and selected 60% of occupancy. Below are results, this heuristic performs better than previous one. On average it is 6% points better than the baseline.

Win % of Student - Win % of ID_Improved-1

	Random	MM Null	MM Open	MM Improved	AB Null	AB Open	AB Improved	Overall
Run1	-10%	15%	0%	5%	-20%	5%	5%	0%
Run2	5%	-20%	15%	25%	20%	-15%	20%	7%
Run3	-5%	10%	-5%	-15%	20%	-5%	5%	0%
Run4	15%	-5%	30%	15%	15%	10%	0%	12%
Run5	0%	10%	20%	20%	20%	15%	-10%	11%
Mean	**1%**	**2%**	**12%**	**10%**	**11%**	**2%**	**4%**	**6%**

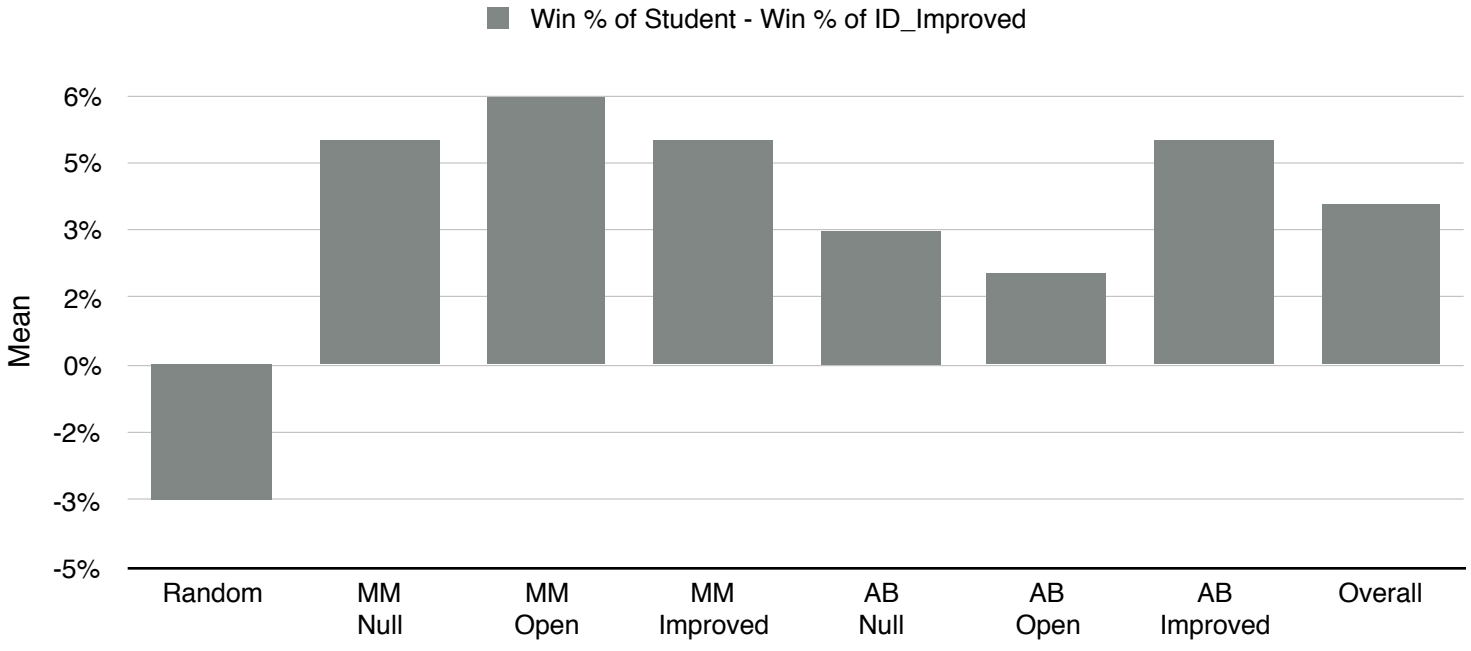


Heuristic 3:

After looking at heuristic 2 performance I wanted to test a more generic version of heuristic 2. So instead of just considering corners and walls, I wanted to weight each spot with a weight proportional to number possible hops from that spot given the state of the board / blank spots in the board. I divided number of possible potential hops by 8 as 8 is the maximum possible number. I used this metric to weight the legal move before counting. Again since I knew this is going to take time I wanted to start after the initial phase or certain board occupancy. Heuristic 3 is comparable to heuristic 2 but more consistent across different matches. But on overall basis heuristic 2 is doing better, which might be due to the fact it is less time consuming than 3 and explores more nodes on average. Another interesting thing to note is that this heuristic does not beat the random one by a lot. Note that in the random case the win rate is very high and usually these runs only vary slightly.

Win % of Student - Win % of ID_Improved-1-1

	Random	MM Null	MM Open	MM Improved	AB Null	AB Open	AB Improved	Overall
Run1	-5%	10%	20%	-5%	0%	-10%	5%	2%
Run2	-10%	15%	15%	0%	-5%	15%	-5%	4%
Run3	5%	15%	-10%	0%	-5%	0%	10%	4%
Run4	5%	-10%	5%	5%	30%	5%	0%	6%
Run5	-10%	-5%	0%	25%	-5%	0%	15%	3%
Mean	**-3%**	**5%**	**6%**	**5%**	**3%**	**2%**	**5%**	**4%**



Reasons & Recommendation

- ✦ Average over 5 runs, considering overall of all matches, heuristic 2 performs the best and hence I selected heuristic 2 as my final evaluation function.
- ✦ Heuristic 2 is a combination of heuristic 1 in the initial phase and corners and walls weighting after that. It is interesting to note that heuristic 1 does not perform relatively well in MM Improved compared to base player, once you add the walls and corners condition to it, it improves a lot, which is another reason I like heuristic 2.
- ✦ Although heuristic 3 is comparable to 2 and more consistent across matches, I was unhappy that it does not beat on average base line performance with the random player.
- ✦ The reason for heuristic 3 not able to beat heuristic 2 might be the fact that it explores lower depth as the execution time is slightly more.
- ✦ With regards to weighting scheme in heuristic 2 I tried a bunch of weights and one take away from it is that aggressively lowering the weight of corners or walls will not yield great results as sometimes they are indeed part of the best possible path.
- ✦ Also another take away is, it is useful to divide the game into different phases and use different heuristics at different phases (i.e more complex ones in the later phases as the number of nodes to explore decreases).
- ✦ Similarly it is beneficial to have a less aggressive heuristic (i.e relatively a good approximation rather than going for one with only of variance - some times performs well and some times terribly) in the starting phases.