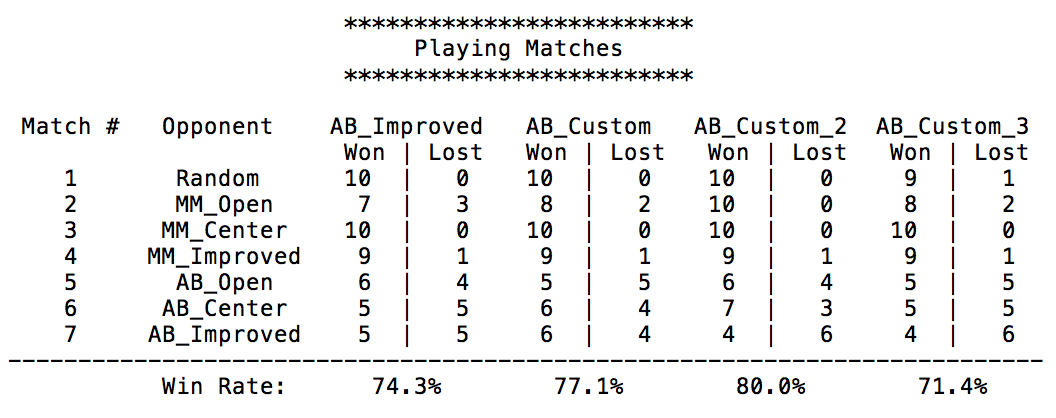
Heuristic Analysis

The following, are the results obtained for the whole process including all the different cases being tested:



Further detail regarding each case will be indicated below.

**Case 1:** *Custom score 1*

What I intended with this heuristic was to force the opponent to the limits, since that's where most of the losses take place. To do this, I built it over the AB\_Improved heuristic. This case maximizes the distance from the center for the opponent and minimizes the distance from the center for the player. However, it is true that after a certain number of moves, it becomes rarer to carry out moves towards the center of the board and these moves may as well not be the optimum ones since they may limit more than at the beginning the movement of the player. For this reason, Hence, I am reducing the weight of this distance parameter as more moves are played. 0.7, which I found gave better results when playing against AB\_Improved.

Results analysis:

This case performs adequately. Although the winning rate isn’t as good as the winning rate for AB\_Improved, it gets quite close when playing 40 games. It is quick to compute and involves additional information about the state of the board. The varying weight defines the importance given to the distance to the center during the game, being the constant used obtained after testing several cases iteratively against the AB\_improved model. Given that the tested cases may not cover all combinations, this number could still be improved by further testing in order to obtain a higher accuracy.

**Case 2:** *Custom score 2*

The intuition behind this heuristic was to stay in the center and use Manhattan distance to compute distance to the center instead of the Euclidean distance. During the initial 10% of the game, the agent tries to aggressively capture the center positions. For the rest of the game, the agent maximizes its own moves.

Results analysis:

This heuristic performs adequately, but is clearly much worse than the simple AB\_Improved. I think the Manhattan distance formula to compute the distance to the center might be performing better than the Euclidean distance formula since it is more relevant to the game player and is also faster to compute. The idea of switching strategies based on a stage in the game is also a good one, but the time to switch the strategy and the optimal strategy to use during a game stage is difficult to find. Towards the end of the game, we’re not looking at the distance to the center because most moves will be away from the center and towards the walls anyway.

**Case 3:** *Custom score 3*

The intuition behind this heuristic was to penalize moves that are on the walls of the board and assess the quality of a move by the distance to the center for each of the future moves. Lesser the distance, better the quality of the move. Towards, the end, the agent aggressively tries to minimize the moves of the opponent.

Results analysis:

Clearly this heuristic performs significantly worse than AB\_Improved. Even though this heuristic takes multiple inputs to assess the quality of the board state, just a summation is clearly not enough. Adding weights to the inputs might help in improving the win rate. This heuristic is also more expensive to compute since we must go through future moves and compute the distance to the center for each. Again, switching strategies based on the stage in the game might be a good idea, but it is difficult to predict when the switch strategies.

**Overall comparison and results:**

****

As you can see from the above visualization, the custom heuristic #1 and AB\_Improved perform comparably. It is possible that after tuning the constant portion of the weight, custom heuristic #1 can achieve better results than AB\_Improved. Custom heuristic #1 is also computationally least expensive. By using the distance to the center for the opponent and the player, it captures more information about the board state when compared to AB\_Improved. Hence, I recommend this heuristic over the others. The remaining two evaluation functions have good ideas like switching strategies and penalization. But these ideas require further research before they can achieve better results than AB\_Improved.