I developed 3 scoring functions which I codenamed cutthroat, aggressive, and centralness. All of them were essentially modified versions of the “improved” score function (my moves - their moves), but each with a twist.

Cutthroat looks at if there’s an overlap in the available moves of the two players, and when there is an overlap of 1 in the legal moves gives a bonus if it’s the player’s turn, or a penalty if it’s the opponent’s turn. The idea is that a cutthroat player might prefer to steal the available spaces that the opponent can move to, thereby reducing their number of moves.

Aggressive is inspired by the lectures, where as the game progresses it puts a larger and larger multiplier on the “their moves” side of the “improved” score function. The later on in the game, the more this agent focuses on reducing the opponent’s moves relative to keeping its own moves.

Centralness values board position by giving a bonus when the player’s piece is within the center of the board, and a further bonus when the opponent’s piece is outside of the center of the board.

After developing my scoring functions by running tournament.py a few times, I ran two large batches by modifying tournament.py to increase the NUM_MATCHES to 100 and remove all of the opponents aside from AB_Improved. I did this because I felt the sample size was far too small in the default settings to draw conclusions from.

AB_Custom = cutthroat
AB_Custom_2 = aggressive
AB_Custom_3 = centralness

```
 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1     AB_Improved   93  |  107   98  |  102   109 |  91    106 |  94
--------------------------------------------------------------------------
           Win Rate:      46.5%        49.0%        54.5%        53.0%
```

I had expected cutthroat to perform the best, however it actually performed the worst out of them! I figured that the knight movement version of isolation didn’t have the board partitioning aspect of the queen movement version, so positioning in a way to steal the opponent’s moves might be a good strategy. Aggressive and centralness both seemed to outperform AB_Improved marginally, however given that AB_Improved wasn’t 50/50 with itself indicates a bit of a margin of error to me that they both fall within so I can’t confidently say any of my scoring functions reliably outplay AB_Improved. I would guess some intelligent combination of the 3 of them might be able to though, since from a high level the idea of stealing opponent’s moves, positioning centrally, and being aggressive later in the game could work well together.

I think the most promising idea is the one behind cutthroat. Although it would need more tuning before it is empirically outperforming AB_Improved, I think it could be the best strategy of the 3 implemented because:

1) Looking at overlapping moves almost lets it look a ply deeper, since it is considering where it and the opponent will be able to move to in evaluated game state is sort of like looking ahead.
2) Strategically, preferring positions where it will be able to take a space that the opponent could've moved to, or prevent the opponent from doing that to it, should help it win games.
3) It doesn't really cost more than AB_Improved to run since you already need the sets of moves for player and opponent to compute that, so looking at the intersection of the move sets shouldn't add much time to the computation and thus should not have much effect on the search depth, yet it does gain an important insight at this minimal cost. So it does everything that AB_Improved does, but with an extra little insight at almost no cost using the same available data.