Isolation Heuristic Analysis

To explore good isolation game heuristics, I start with information given in the lectures and empirical results from running the tournament.py script. Running the tournament.py script unchanged showed that the AB Improved player does quite well against all comers with a winning percentage north of 75%. Also, in the "Solving 5x5 Isolation" lecture, we were told that the best first move is always the center square for both agents, at least for the 5x5 version. Intuitively, biasing moves toward the center of the board seems to provide the most options for later moves. From this line of thought, the first evaluation function that I evaluated is a combination of the center scoring and improved scoring evaluation functions (custom_score_3()):

```
score = improved_score() - x * center_score()
```

Note that the center_score is the distance from the center, so biasing moves toward center means subtracting it from the improved score so that moves further away from center are less favored. The range of center_score varies from 0.5 to 24.5, and the range of improved_score varies from -7 to 7. To limit the impact of center_score I set x to 0.2 to give the center_score component a range from 0.1 through 4.9. Since center_score is larger for player positions farther from the center, the above evaluation penalizes moves further away from the center of the board.

Note also that I increased the match count to 20 for more resolution. Here is the performance of this strategy in 20 matches against each non-random opponent:

score = improved_score() - 0.2 * center_score() performance

```
 Match #   Opponent    AB_Improved  AB_Custom_3
                        Won | Lost   Won | Lost
    1       MM_Open     13  |   7    13  |   7
    2      MM_Center    18  |   2    15  |   5
    3     MM_Improved   16  |   4    15  |   5
    4       AB_Open      9  |  11    11  |   9
    5      AB_Center    15  |   5    11  |   9
    6     AB_Improved   10  |  10     9  |  11
--------------------------------------------------------------------------
           Win Rate:      67.5%        61.7%
```

Biasing moves toward the center actually degraded the improved score evaluation function. This would seem to indicate that while an opening move at the board center is good, over the course of a 7x7 game the variety of possible board positions makes always biasing moves toward the center suboptimal.

I also evaluated the case where the center score is biased away from the board's center. Since this is a minor change to the evaluation function I still consider this option AB_Custom3.

score = improved_score() + 0.2 * center_score() performance

```
 Match #   Opponent    AB_Improved  AB_Custom_3
                        Won | Lost   Won | Lost
    1       MM_Open     18  |   2    12  |   8
    2      MM_Center    19  |   1    17  |   3
    3     MM_Improved   13  |   7    14  |   6
    4       AB_Open     14  |   6    12  |   8
    5      AB_Center    15  |   5    11  |   9
    6     AB_Improved   11  |   9     9  |  11
----------------------------------------------------------------------
           Win Rate:      75.0%        62.5%
```

This approach also underperformed AB_Improved alone. It seems trying to bias moves away from the board center in combination with AB_Improved also degrades performance in general.

Note that in general all AB strategies did substantially better than all MM strategies because their better pruning of the search tree allows for deeper searches with iterative deepening in the 150 msec time limit per move.

For *custom_score_2()*, I looked at the relative importance of maximizing our available moves versus minimizing the opponent's moves by weighting these values differently in the evaluation function. I added a multiplicative factor to our available moves before subtracting the opponents available moves, yielding the following evaluation function:

```
score = float(factor * own_moves - opp_moves)
```

I tested the cases of factor = 2 (weight own moves more) and factor = 0.5 (weights opponent moves more)

score = float(factor * own_moves - opp_moves) performance

```
                                 factor=0.5   factor=2.0
 Match #   Opponent    AB_Improved  AB_Custom_2  AB_Custom_2
                        Won | Lost   Won | Lost   Won | Lost
    1       MM_Open     17  |   3    17  |   3    15  |   5
    2      MM_Center    20  |   0    19  |   1    19  |   1
    3     MM_Improved   12  |   8    16  |   4    16  |   4
    4       AB_Open     11  |   9     8  |  12    13  |   7
    5      AB_Center    14  |   6    11  |   9    14  |   6
    6     AB_Improved   10  |  10     9  |  11    10  |  10
----------------------------------------------------------------------
           Win Rate:      70.0%        66.7%        72.5%
```

While there are differences in performance, the extent is not quite as drastic as with adding center_score() to improved_score in *custom_score_3()*. The above result does show a trend where more emphasis on the number of own legal moves produces better result, though given the wide variety of starting positions and

their potential impacts a case can be made that these results are within the error margin.

From the lecture it seems that making a move at the board center whenever possible along with identifying partitions will for sure help improve winning chances. Partitions are essentially shortcuts to identifying winning or losing moves without performing a full search of the end-games. They are characterized by no intersection between our own legal moves and the opponent's legal moves. These conditions don't always happens but when they do we can immediately identify a winning or losing move and respond accordingly.

These changes are complementary to a good evaluation function so in *custom_score_2()* and *custom_score_3()*, I wanted to see if I can improved on AB_Improved. I decided to use the weighted improved_score() method with factor=2. I will use this evaluation function along with the board center opening moves and partition identification in *custom_score()*.

For this round I will only compare performance against the AB agents. The proven inferiority of the MM agents the variety of random start positions causes too much variance in win/loss results. Limiting comparisons to AB agents provides a tighter comparison.

improved_score() with center moves and partition identification performance

```
 Match #   Opponent    AB_Improved   AB_Custom
                        Won | Lost   Won | Lost
    1       AB_Open      4  |  16    13  |   7
    2      AB_Center    11  |   9    10  |  10
    3     AB_Improved   10  |  10    10  |  10
----------------------------------------------------------------------
           Win Rate:      41.7%        55.0%
```

Looks like *custom_score()* could outperform AB_Improved, though the AB_Open results might be skewed with overly bad starting positions for AB_Improved.

Among custom_score(), custom_score_2(), and custom_score_3(), I would favor custom_score() since it performs at least as well or better than AB_Improved and in certain cases where partitions are created it can immediately take the winning move or avoid a sure losing move.