The paper summarizes three heuristic functions used by the *AlphaBetaPlayer* function when performing iterative deepening search in the game isolation.

Rationale

The approach used was to start with the simple heuristics covered in the lectures and then test and implement more complex heuristics. Performance was measured as the sum of the win rates. Changes were incrementally applied and tested using tournament play to gain extensive experience understanding the impact of a variety of input factors on performance.

Three Heuristics

As hinted to in the lecture, the best heuristic was that that favored board position, as it allowed greater control in partitioning the move field. The score is calculated using a mixing function that adds value when the player is closer to the edge of the board when the game nears the end.

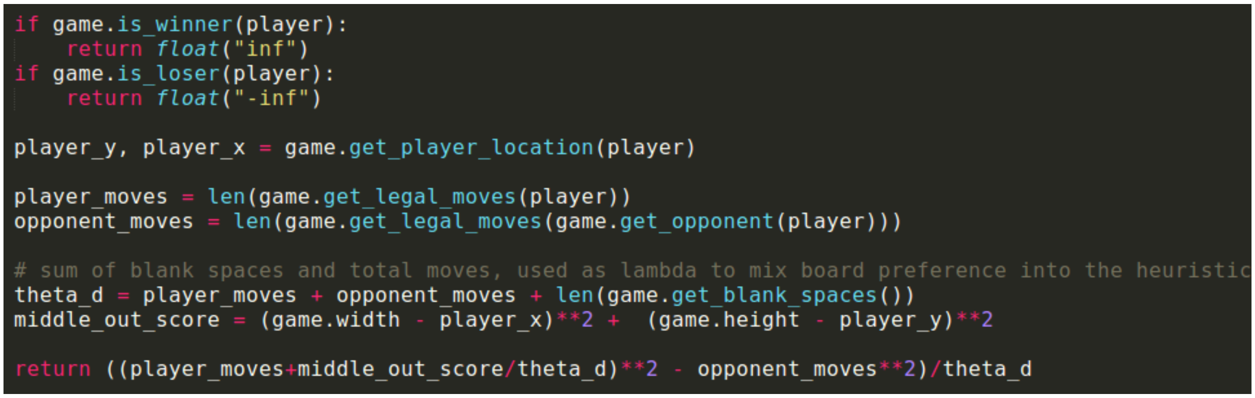
The second best performing heuristic was simple differencing of squared player moves minus the opponent moves squared.

A novelty heuristic was used see how adding a factor based on the game move count performed. In practice, it turns out that this factor did not significantly improve performance.

Recommendation

Of the three custom score functions, I recommend AB_Improved, called ‘middle-out’, for the following reasons:

1. ‘Middle-out’ is not computationally complex. Theta_d helps the clarity of the situation:



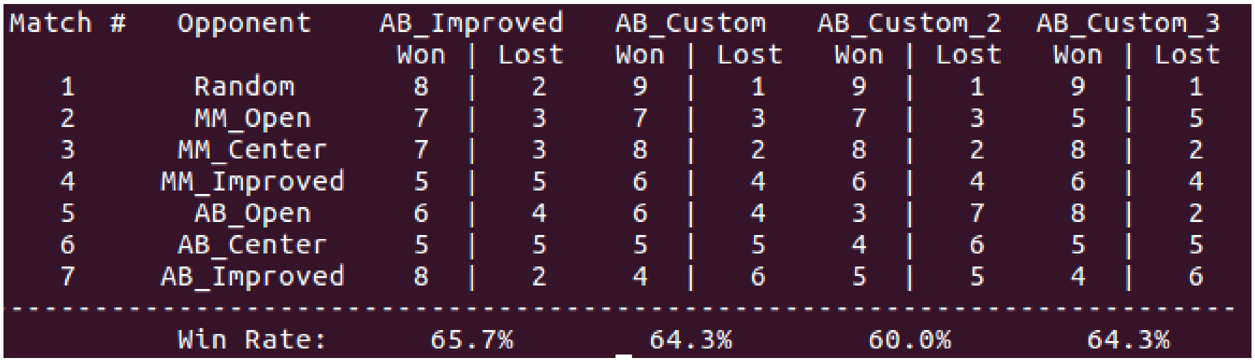
```
if game.is_winner(player):
    return float("inf")
if game.is_loser(player):
    return float("-inf")

player_y, player_x = game.get_player_location(player)

player_moves = len(game.get_legal_moves(player))
opponent_moves = len(game.get_legal_moves(game.get_opponent(player)))

# sum of blank spaces and total moves, used as lambda to mix board preference into the heuristic
theta_d = player_moves + opponent_moves + len(game.get_blank_spaces())
middle_out_score = (game.width - player_x)**2 +  (game.height - player_y)**2

return ((player_moves+middle_out_score/theta_d)**2 - opponent_moves**2)/theta_d
```

2. Middle-out’s ‘win rate’ is consistently high relative to the other methods. The performance summary below summarizes several rounds of performance under

tournament conditions. AB_Custom_1 shows higher average performance than both AB_Custom_2 and AB_Custom_3:

AB_Improved	AB_Custom_1	AB_Custom_2	AB_Custom_3
60.0%	60.0%	68.6%	58.6%
67.1%	64.3%	64.3%	58.6%
67.1%	62.9%	57.1%	68.6%
68.6%	65.7%	57.1%	60.0%

3. The AB_Custom_1 heuristic provides better predicts the final outcome of the game. The above tournament summary table showed considerable variance between tournament performance. In comparing the Win-Loss at the opponent level, we can see that AB_Custom_1 has solid relative outcome prediction.



```
Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random       8  |   2     9  |   1     9  |   1     9  |   1
    2       MM_Open      7  |   3     7  |   3     7  |   3     5  |   5
    3      MM_Center     7  |   3     8  |   2     8  |   2     8  |   2
    4     MM_Improved    5  |   5     6  |   4     6  |   4     6  |   4
    5       AB_Open      6  |   4     6  |   4     3  |   7     8  |   2
    6      AB_Center     5  |   5     5  |   5     4  |   6     5  |   5
    7     AB_Improved    8  |   2     4  |   6     5  |   5     4  |   6
--------------------------------------------------------------------------
          Win Rate:      65.7%        64.3%        60.0%        64.3%
```