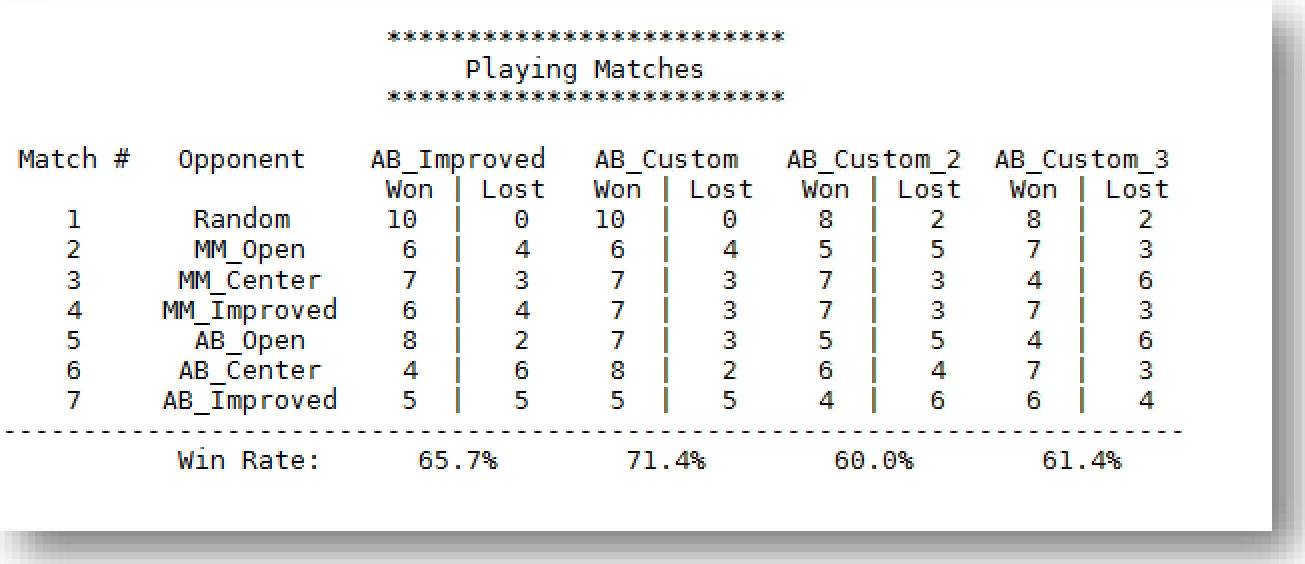
Kelvin Kong
Udacity Artificial Intelligence Nanodegree
Aug 14, 2017

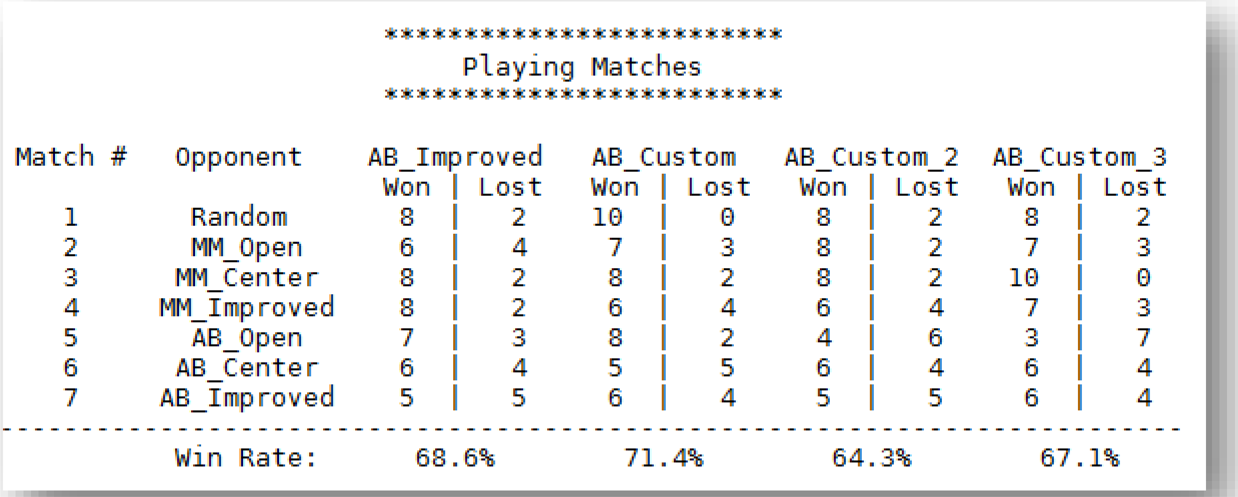
Isolation Heuristics Analysis



```
                         **************************
                              Playing Matches
                         **************************

 Match #     Opponent     AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                           Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1         Random       10  |   0    10  |   0     8  |   2     8  |   2
    2         MM_Open       6  |   4     6  |   4     5  |   5     7  |   3
    3        MM_Center      7  |   3     7  |   3     7  |   3     4  |   6
    4       MM_Improved     6  |   4     7  |   3     7  |   3     7  |   3
    5         AB_Open       8  |   2     7  |   3     5  |   5     4  |   6
    6        AB_Center      4  |   6     8  |   2     6  |   4     7  |   3
    7       AB_Improved     5  |   5     5  |   5     4  |   6     6  |   4
--------------------------------------------------------------------------
           Win Rate:      65.7%        71.4%        60.0%        61.4%
```

Figure 1 Tournament Results Round 1



```
                         **************************
                              Playing Matches
                         **************************

 Match #     Opponent     AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                           Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1         Random        8  |   2    10  |   0     8  |   2     8  |   2
    2         MM_Open       6  |   4     7  |   3     8  |   2     7  |   3
    3        MM_Center      8  |   2     8  |   2     8  |   2    10  |   0
    4       MM_Improved     8  |   2     6  |   4     6  |   4     7  |   3
    5         AB_Open       7  |   3     8  |   2     4  |   6     3  |   7
    6        AB_Center      6  |   4     5  |   5     6  |   4     6  |   4
    7       AB_Improved     5  |   5     6  |   4     5  |   5     6  |   4
--------------------------------------------------------------------------
           Win Rate:      68.6%        71.4%        64.3%        67.1%
```

Figure 2 Tournament Results Round 2

Opponent Chase Score – Implemented as Custom_Score

1. Increase player aggressiveness by favoring moves which overlaps with the opponent
    a. Outputs a score which factors in the number of move overlaps between the player and the opponent
    b. An aggressive player will tend to choose moves that are closer to the opponent. Overlapping moves means that the player is within close range of the opponent. Our assumption is that

the higher the count of overlaps, the closer the distance between the player and the opponent.

c. This score will also maximize winning by reinforcing priority on moves that will result in the opponent having less possible moves in the next state, by penalizing the score with the opponent's number of legal moves by a factor of 2. In other words, the less number of legal moves the opponent have will result in a higher score.

d. We will design our heuristics to favor such moves by turning this observation into a score multiplier.

e. Increase the score by multiplying the number of overlaps with the number of legal moves the player currently has.

f. This heuristic is implemented with the formula:

$$Number\ of\ overlapping\ moves\ with\ Opponent + Number\ of\ Player\ Legal\ Moves - (\ 2 * Number\ of\ Opponent\ Legal\ Moves)$$

$$where\ Number\ of\ overlapping\ moves\ with\ Opponent:$$

$$Number\ of\ player\ legal\ moves\ which\ also\ exist\ in\ opponent's\ legal\ move\ list$$

Analysis of the tournament results shows that the opponent chase score (implemented in AB_Custom) achieved a 71.4% win percentage overall. Against the strongest opponent AB_Improved, it managed a tie (5 vs 5) on the first round and had a close win on the 2nd round with 6 vs 4. Although results do vary on different attempts, on average it managed to outperform other score heuristics, thus we choose this as our optimum score.

Opponent Edge score - Implemented as Custom_Score_2

1. This heuristic works by using the idea that if the opponent is closer to the edge, the opponent is more likely to have less possible moves.
2. In other words, if the opponent have more moves that are closer to the center, it is not in our favor since moves in the center generally opens up more legal moves for the opponent
3. We implement this heuristic by increasing the score of the forecasted board if the opponent's average squared distance of his legal moves are further away from the center of the board.
4. This score is implemented with the following formula:

$$Num\ of\ player\ legal\ moves + Average\ Opponent\ Squared\ Distance\ from\ Center - 2 * Number\ of\ Opponent\ Legal\ Moves$$

$$where\ Average\ Opponent\ Squared\ Distance\ from\ Center:$$

$$\frac{\sum(Distance\ of\ Opponent\ move\ to\ Center\ of\ board)^2}{Num\ of\ Opponent\ Moves}$$

5. The Opponent Edge score managed a 60.0% & 64.3% win against other tournament agents, winning most of the Minimax-based agents while managing a tie with Alpha-Beta based agents. This is expected since this heuristic is based on Alpha Beta search as well.

This heuristic appears to be the least competitive among the rest, although it still a formidable opponent since it incorporates the algorithm that will minimize the opponent's legal moves while maximizing its own.

Opponent Chase & Push Opponent to Edge Score – Implemented as Custom_Score_3

1. This heuristic is the combination of the two ideas above, in an attempt to create an aggressive player that will:
    a. Chase / favor moves that will bring the computer player closer to the opponent.
    b. Maximize winning by favoring moves that will result in the opponent have less possible moves
    c. Pushing opponents to the edge by favoring moves that will result in opponents being further away from the center of the board.
2. This heuristic is implemented by:

$$\begin{aligned} &Number\ of\ overlapping\ moves\ with\ Opponent + Number\ of\ Player\ Legal\ Moves \\ &\quad + Average\ Opponent\ Squared\ Distance\ from\ Center \\ &\quad - (\ 2 * Number\ of\ Opponent\ Legal\ Moves) \end{aligned}$$

3. On a glance, it is expected that this score heuristic should perform the best since it incorporates both opponent chasing and opponent edge score heuristics. However, through multiple iterations of testing, it does not seem to be the case. This combination did not manage to outperform the Opponent Chase score (AB_Custom), and is also less competitive than AB_Improved. It did manage to beat the Opponent Edge score heuristic (AB_Custom_2), showing that the number of overlapping moves plays a larger role in determining the winner than a score that evaluates whether the opponent is closer to the edge.

Conclusion

The Opponent Chase Score heuristic did manage to outperform AB_Improved in most cases, showing that factoring the number of overlapping moves into the score calculation may play a role in increasing the chances of winning. More variations of the equation can be experimented to determine the best winning formula.

This experiment had also shown that combinations or complex heuristics may not always be the best – It could be bogged down by consuming too much processing time and power and ended up not finding the best move that is hidden way below the search tree, or the combination simply did not represent the state of the player well enough. Many times, it is the simple heuristics that performs the best.

Therefore, for this game of isolation, the Opponent Chase Score heuristic is the best evaluation function to be used. Reason is that based on the various tournament results, this score function performed the best by giving an average win rate of 71.4% against other opponents, which is the highest of all. It consistently beat AB_Improved, which is our baseline opponent that managed a score 65.7% and 68.6% in both tournaments. In one of the rounds, it managed to outperform AB_Improved with 6 wins vs 4 losses. The function is simple in terms of complexity, as the number of overlapping moves is easily calculated from the board state and does not heavily impact the processing time. It also outperformed the more complicated Opponent Chase & Push Opponent to Edge score which managed to obtain 67.1% and 61.4% win rates, proving that complexity is not always better.