HEURISTIC ANALYSIS

By Vineet Jacob Kuruvilla

Introduction

As a part of the Isolation game playing agent project, I have implemented three heuristics- custom_score, custom_score_2 and custom_score_3. All the three heuristics uses one or more of the following factors, which I believe have an impact on the winnability of a move:

a. *Number of available moves of player*: Higher the number, better is the position of the player
b. *Number of available moves of the opponent*: If this number is much higher than the player then the player will be at disadvantage. Hence evaluating a move based only on the number of available moves of the player is not enough. A better move would be one where the number of moves available to the player is higher than its opponent
c. *Distance to the board centre of the player*: In the game of isolation, especially when the players have chess knight type L-shape movement, it is critical capture the central region of the board. This helps in maximising the *number of available moves* as well as the potential for higher number of future moves.

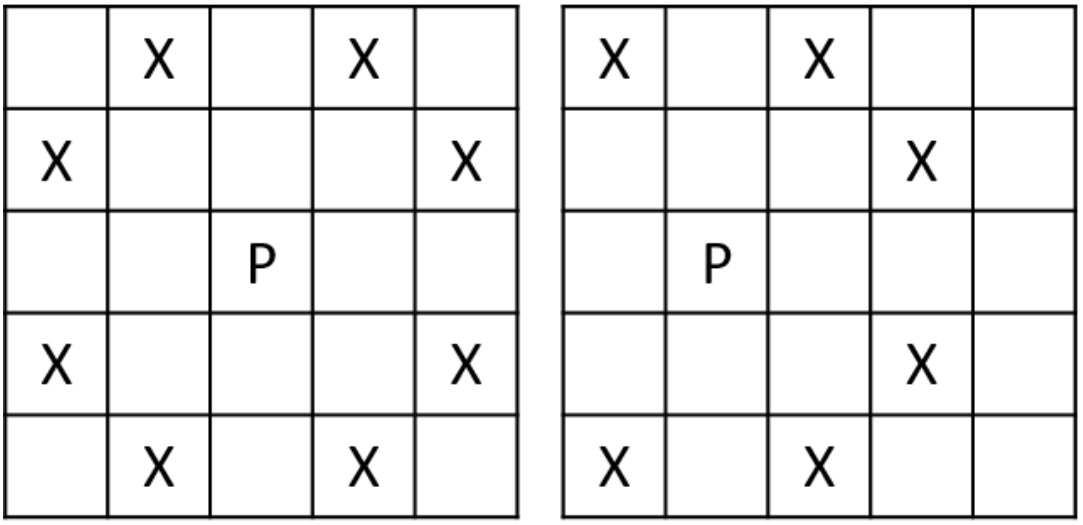
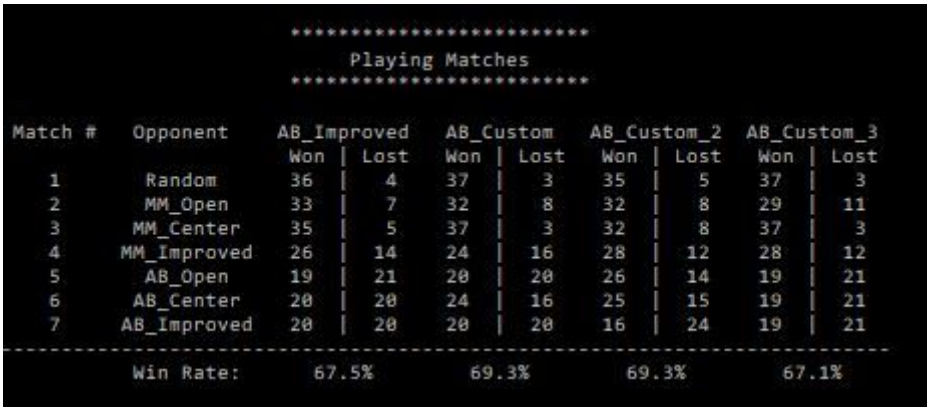


Fig 1 a) Player at the centre of the board with 8 possible moves, b) Player is off-centre has only 6 possible moves

d. *Distance to the board centre of the opponent*: Just like in the case of available moves, looking at the distance to the centre of the player alone may not be helpful. Since this is an adversarial game, it is important to know the status of the opponent as well. Lower this number, better is the player's move.

Analysis

The explanation of the three heuristics implemented using the factors listed in the Introduction section are described below. The results shown below are from tournament played with the test agents and cpu agents with 20 matches a side (i.e., 40 matches against each opponent).



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

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                      Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      36  |   4    37  |   3    35  |   5    37  |   3
   2       MM_Open     33  |   7    32  |   8    32  |   8    29  |  11
   3      MM_Center    35  |   5    37  |   3    32  |   8    37  |   3
   4     MM_Improved   26  |  14    24  |  16    28  |  12    28  |  12
   5       AB_Open     19  |  21    20  |  20    26  |  14    19  |  21
   6      AB_Center    20  |  20    24  |  16    25  |  15    19  |  21
   7     AB_Improved   20  |  20    20  |  20    16  |  24    19  |  21
--------------------------------------------------------------------------
          Win Rate:      67.5%        69.3%        69.3%        67.1%
```

Fig 2: Overall result from tournament (screenshot)

1. custom_score

This heuristic computes the difference between the distance to board centre of our agent and that of the opponent. The position of a player is better if it is closer to the centre of the board. It compares the distance of the player to its opponent for it's proximity to the board centre. If the player is closer to the centre than the opponent then the score will be higher.

score = distance_to_centre for opponent - distance_to_centre for our agent

Table 1: Result comparison- ID_improved vs. ID_custom_score

Match	Opponent	ID_improved (W-L)	ID_custom_score (W-L)
1	Random	36 - 4	37 - 3
2	MM_Open	33 - 7	32 - 8
3	MM_Center	35 - 5	37 - 3
4	MM_improved	26 - 14	24 - 16
5	AB_Open	19 - 21	20 - 20
6	AB_Center	20 - 20	24 - 16
7	AB_Improved	20 - 20	20 - 20
Result		**Win- 67.5%**	**Win- 69.3%**

custom_score heuristic achieves a marginally better win rate compared to improved_score heuristic.
improved_score heuristics looks at the difference between the number of available moves of the player and the opponent. This gives us information about the advantage of the player over the opponent for the current state of the game. *custom_score* indirectly gets this same information. But, by assessing its proximity to the board centre, custom_score also gives information about the potential for larger number of available moves in future.

2. custom_score_2

This heuristic looks at the number of moves available to the player and its opponent. The score is the ratio between the two counts.

Score = number of player moves/number of opponent moves

Table 2: Result comparison- ID_improved vs. ID_custom_score

Match	Opponent	ID_improved (W-L)	ID_custom_score (W-L)
1	Random	36 - 4	35 - 5
2	MM_Open	33 - 7	32 - 8
3	MM_Center	35 - 5	32 - 8
4	MM_improved	26 - 14	28 - 12
5	AB_Open	19 - 21	26 - 14
6	AB_Center	20 - 20	25 - 15
7	AB_Improved	20 - 20	16 - 24
Result		**Win- 67.5%**	**Win- 69.3%**

For *custom_score_2* too the win rate is slightly better compared to *improved_score*. But I am not able to give a full explanation for the result.
In AB_improved versus AB_custom_score_2, I believe AB_improved got better win rate due to the less computation time (and hence deeper search) of its heuristic. *custom_score_2* requires division which is computationally more expensive compared to subtraction.
However, AB_improved performed poorly (~50% win rate) against AB_Open and AB_Center while AB_custom_score_2 quite well against the same opponents. My expectation was both AB_improved and AB_custom_score_2 would fare well against the two opponents as they look at the opponents' status.

3. custom_score_3

The goal in the game of isolation is to maximise the available moves of the player at any given point of time. The chances of having more is moves after the next is also much higher if the current position is close to the centre of the board. This heuristic (custom_score_3) tries to evaluate the value of move by taking both these criteria.

The score of this heuristic is given as:
Score = w1 * number of available moves - w2 * (1/distance_to_centre)
(1/distance_ to_centre) will be greater as the position is closer to the centre.
If distance_ to_centre is equal to zero, score will be +inf.
w1 - weightage given to number of available moves
w2 - weightable given to distance to the board centre.
w1 + w2 = 1

Different values for w1 were tried and the best choice seems to be 0.5 (equal weightage for both parameters).

Table 3: Result comparison- ID_improved vs. ID_custom_score_3

Match	Opponent	ID_improved (W-L)	ID_custom_score_3 (W-L)
1	Random	36 - 4	37 - 3
2	MM_Open	33 - 7	29 - 11
3	MM_Center	35 - 5	37 - 3
4	MM_improved	26 - 14	28 - 12
5	AB_Open	19 - 21	19 - 21
6	AB_Center	20 - 20	19 - 21
7	AB_Improved	20 - 20	19 - 21
Result		**Win- 67.5%**	**Win- 67.1%**

The win rate of *custom_score_3* is almost same as AB_improved. No advantage can be found by having a heuristic with a combination of number of available moves and distance to the centre. In fact from the result, we can infer that the additional computation might led to *custom_score_3* to perform slightly worse than AB_improved. To make a clear conclusion about this, a tournament with larger number of matches need to be performed.

Recommendation

As a conclusion I would like to recommend the use of heuristic, custom_score to be used along with iterative deepening search method. These are the reasons for my recommendation:

1. Higher win rate (69.3%) compared to all the other heuristics. custom_score_2 has also has an overall win rate of 69.3%. However, it performs poorly against AB_improved (< 50% win rate) while AB_custom_score gets a 50% win rate

2. custom_score looks at the distance to centre of board of both the player as well its opponent's. Distance to centre is a good indicator of the higher number of moves at that state but also in future.
3. Its computation is simple. Hence less computation time and as a result more time to search deeper.