Kok Leong Fong April 5th, 2018

Heuristic Analysis Report

In this report, three types of heuristic functions or so called evaluation function for playing a simplified chess game is discussed. The functions are conceptually similar because they are quantitative of the states of the chess game. The chess game has two players and the players take turns to make a 2 by 1 move across the board of fixed width and height. Each occupied square on the board must not be revisited. Therefore, it makes sense to consider, at any time of the game, the number of open moves of the board, the difference between current player and opponent player open moves and also the distance of active player to the center of the board.

First function name is AB_Custom which computes the linear combination in the following form

```
AB_Custom(x,y) = x*#my_moves - y*#opponent_moves
```

where `x` and `y` are parameters. The variables `#my_moves` and `#opponent_moves` both mean the number of active player open move and the number of opponent open moves, respectively. In this test, `x=0.2` and `y=0.8`. Note that AB_Improved have both parameters set to `1`. Reducing the impact of the evaluation does not seem to improve the score. Similarly, increasing the value of the parameters also show no significant winning improvement.

The second function name is AB_Custom_2 which computes the linear combination in the following form

```
AB_Custom_2 = #my_moves + AB_Custom(1,1) + distance_to_center
            = 2*#my_moves - #opponent_moves +distance_to_center
```

which is essentially sum of all three types of heuristic functions.

Finally, the third function called AB_Custom_3 has the following form

```
AB_Custom_3(x,y) = x*AB_Custom(1,1) + y*#my_moves
                 = (x+y)*#my_moves - x*#opponent_moves
   subject to (x+y) >= 1 and x < 1
```

is another different way to evaluate move value using a linear combination of two functions described earlier. In this test, `x=0.7` and `y=0.3`. Figure 1 shows the results of the three heuristic functions executed twice. AB_Custom_3 has been able to show head to head performance with default heuristic function AB_Improved. It also managed to maintained at least 5 wins against any type of CPU players consistently throughout many runs.

The performance consistency of AB_Improved varies especially when played against alpha-beta CPU players which uses same pruning mechanism. This is obvious against AB_Open and AB_Center CPU players. This algorithm is naive because it always penalizes moves that do not isolate the opponent effectively. Hence, we considered a more aggressive function in AB_Custom_3 where it penalizes moves that do not isolate the opponent effectively at a lesser magnitude. This is a good heuristic as it focuses on avoiding being isolated but also capturing the impact it has on isolating the opponent. AB_Custom_3 is a special case of AB_Custom_1 where for every `x` value it penalizes `#opponent_moves`, it awards `x+y` value of `#my_moves` with the condition that `x+y >= 1` and `x < 1` to ensure a consistent influence between the two move counts. The other function AB_Custom_2 show no improvements over the rest even though its a linear combination of all three heuristics with linear coefficients one. This shows that a board game has a big state space to be considered and each state may require different strategy. For board games like this, it requires tuning and studying a finite set of CPU strategies. When played against human with much more strategies or bigger state space, algorithm like neural network used in Google AlphaGo may be required together with a good amount of training data.

Figure 1 consists of two subfigures ***a*** and ***b***. The figures are results of two runs playing against 7 types of CPU player using the same heuristic functions to show consistency.

```
                    *************************
                         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     10  |   0     9  |   1
    2       AB_Open      6  |   4      7  |   3      3  |   7     5  |   5
    3      AB_Center     5  |   5      3  |   7      3  |   7     5  |   5
    4     AB_Improved    5  |   5      4  |   6      2  |   8     6  |   4
    5       MM_Open      9  |   1      7  |   3      7  |   3     9  |   1
    6      MM_Center     9  |   1      8  |   2     10  |   0     9  |   1
    7     MM_Improved    7  |   3      8  |   2      6  |   4     9  |   1
----------------------------------------------------------------------------
          Win Rate:      72.9%        67.1%        58.6%        74.3%
```

1a

```
                    *************************
                         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      9  |   1      9  |   1     9  |   1
    2       AB_Open      6  |   4      3  |   7      5  |   5     7  |   3
    3      AB_Center     9  |   1      5  |   5      3  |   7     5  |   5
    4     AB_Improved    3  |   7      4  |   6      3  |   7     7  |   3
    5       MM_Open      7  |   3      5  |   5      7  |   3     9  |   1
    6      MM_Center    10  |   0      7  |   3     10  |   0     9  |   1
    7     MM_Improved    8  |   2      9  |   1      7  |   3     7  |   3
----------------------------------------------------------------------------
          Win Rate:      75.7%        60.0%        62.9%        75.7%
```

1b