HEURISTIC FUNCTIONS ANALISYS

Markus Buchholz

In this project the concept (defensive heuristic) presented during the lecture was applied. In this case the strategy was to measure the results of legal move between player 1: #my-moves and player 2 #opponent-moves. In order to boost presented concept(scores) several solutions were approached. The implemented heuristic functions (and test results) can be summarize as follows:

1. Custom Score

First function calculates difference between the number of legal moves for player 1 and player 2. Possible movement for player 2 are multiplied by 2. Following strategy supports playing defensive game by blocking.

```
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
move_score = own_moves - 2*opp_moves
return float(move_score)
```

2. Custom Score

Second function was developed by dividing move_score by number of blank spaces on the board, which normalized output.

```
blank_spaces = len(game.get_blank_spaces())
move_score = float(len(game.get_legal_moves(player)) * 4 /
                   - len(game.get_legal_moves(game.get_opponent(player))) * 2)
return move_score / blank_spaces
```

3. Custom Score

Third function was developed by additional measuring of the distance on the board between both players. The idea was to give an extra reward to the player who can be blocked in lowered probability if this distance increases.

```
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))

move_score = own_moves - (2 * opp_moves)

own_position = game.get_player_location(player)
opp_position = game.get_player_location(game.get_opponent(player))

distance = abs(own_position[0] - opp_position[0]) + abs(own_position[1] - opp_position[1])

return (float(move_score / float(distance)))
```

Conclusion and recommendations

During the research, considerable number of function were verified. Due to failure (low) test results it was decided to base all function on proven and show strategy. All applied in this project functions allow to achieve results over 65%. For the best applied function, which in addition computes the distance (giving the extra reward) the results is nearly 77%.

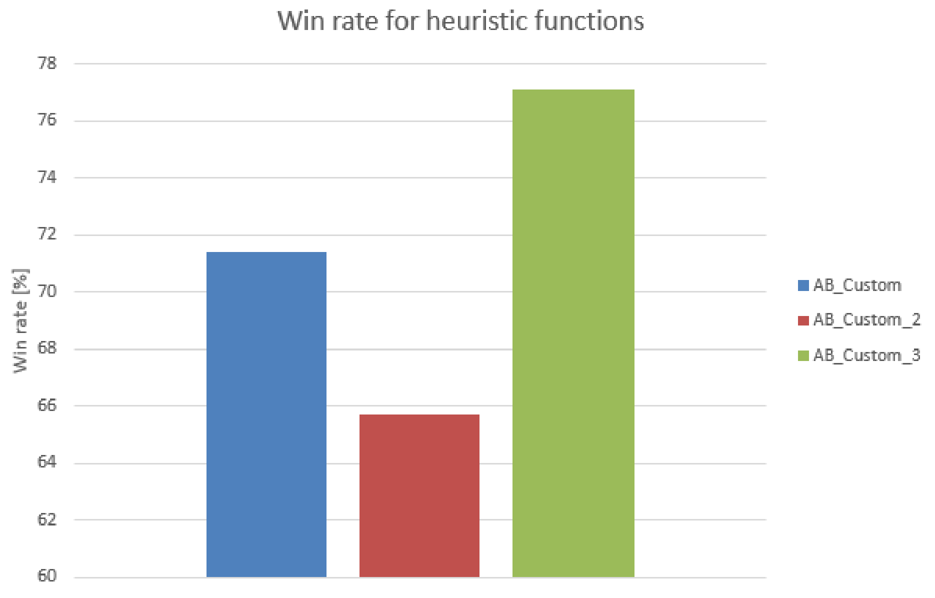
In this research is was proven that the best score could be achieved by use of third heuristic function. Beside the defensive strategy it give best results. It demands computation extra effort but results are high. It is highly **recommended** to use this function (where the difference of possible movements for player 1 and 2 are divided by the distance between them).

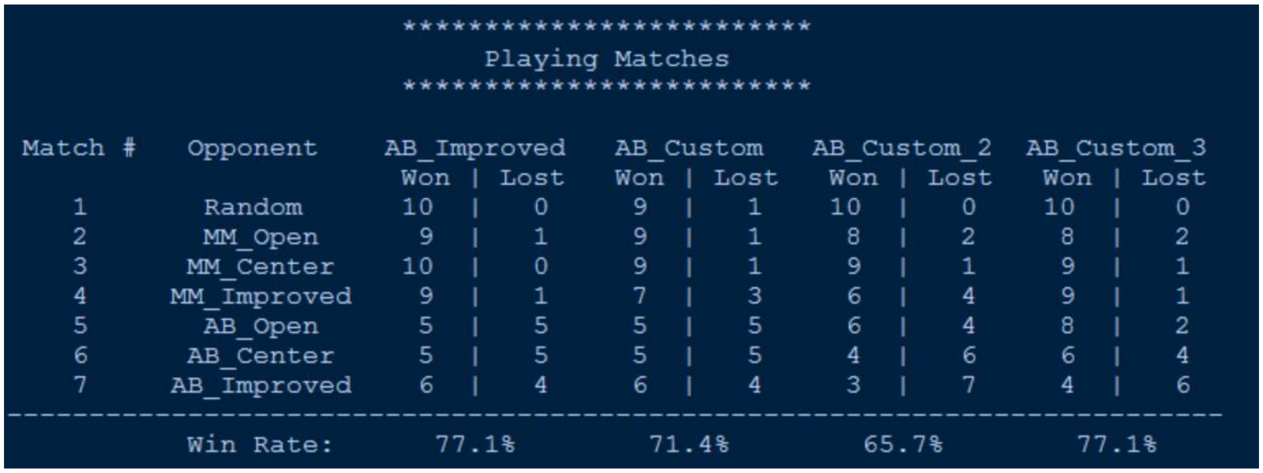
Reason 1. The function support (reward) player if the player has more moves that are possible then the opponent.

Reason 2. The chosen function penalize the player if the opponent has more possible movement on the board.

Reason 3. The function gives the extra reward to the player who can be blocked in lowered probability if this distance between player 1 and player 2 is bigger.

In addition, the function makes possible future adjustments (factors in function numerator can be adjusted if the game conditions will change).





```
                          *************************
                              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      10  |   0      10  |   0
    2         MM_Open      9  |   1      9  |   1       8  |   2       8  |   2
    3        MM_Center    10  |   0      9  |   1       9  |   1       9  |   1
    4       MM_Improved    9  |   1      7  |   3       6  |   4       9  |   1
    5         AB_Open      5  |   5      5  |   5       6  |   4       8  |   2
    6        AB_Center     5  |   5      5  |   5       4  |   6       6  |   4
    7       AB_Improved    6  |   4      6  |   4       3  |   7       4  |   6
--------------------------------------------------------------------------
           Win Rate:        77.1%          71.4%          65.7%          77.1%
```