Heuristic Analysis

- author: Xin Chen
- email: Bismarrck@me.com

1. Overview

In this project three different parameterized heuristics are implemented. The parameters for these score functions are learnt with grid search:

```
python3 grid_search.py {fn1|fn2|fn3} --num_matches=10
```

2. Score Functions

This section introduces the details of the three custom score functions and the searched parameters.

2.1 Custom Score 3

The heuristic `custom_score_3` is an improved version of the score function `improved_score`:

```
float(num_own_moves * a - num_opp_moves * b)
```

- The best parameter set is `a = 6, b = 3`.

2.2 Custom Score 2

The heuristic `custom_score_2` not only considers the current available legal moves (`custom_score_3`) but also tries to include the possible moves for next turn.

```
player_score = num_next_own * b + num_own_moves * a
opp_score = num_next_opp * c
return float(player_score - opp_score)
```

- `num_opp_moves` is ignored because it is included in `num_next_opp`.
- The best parameter set is `a = 7, b = 1, c = 2`

2.3 Custom Score 1

The heuristic `custom_score` is further improved from `custom_score_2` as only unique future moves are included:

```
num_own_controlled = len(set(own_controlled))
num_opp_controlled = len(set(opp_controlled))
```

The final score has four learnable parameters:

```
own_score = num_own_moves * a + num_own_controlled * c
opp_score = num_opp_moves * b + num_opp_controlled * d
return float(own_score - opp_score)
```

- The best parameter set is `a = 5, b = 3, c = 1, d = 1`

3. Results

The tournament settings:

- NUM_MATCHES: 20
- TIME_LIMITS: 150 ms
- CPU: **E5-2670v3**

3.1 Default matches

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

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      40  |   0    38  |   2    37  |   3    36  |   4
    2       MM_Open     32  |   8    30  |  10    35  |   5    32  |   8
    3      MM_Center    34  |   6    37  |   3    31  |   9    34  |   6
    4     MM_Improved   29  |  11    32  |   8    33  |   7    29  |  11
    5       AB_Open     17  |  23    20  |  20    22  |  18    18  |  22
    6      AB_Center    26  |  14    23  |  17    21  |  19    19  |  21
    7     AB_Improved   16  |  24    23  |  17    21  |  19    19  |  21
--------------------------------------------------------------------------
           Win Rate:      69.3%        72.5%        71.4%        66.8%
```

3.2 Custom matches

```
 Match #   Opponent    AB_Custom_1   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost
    1     AB_Custom_1                18  |  22    16  |  24
    2     AB_Custom_2   22  |  18                 19  |  21
    3     AB_Custom_3   24  |  16    21  |  19
---------------------------------------------------------------
          Win Rate:        57.5%        48.8%        43.8%
```

4. Conclusion

`Custom_function_1` should be chosen because:

1. **`Custom_function_1`** has the best performance against default opponents.
2. **`Custom_function_1`** does not loose any single match.
3. **`Custom_function_1`** can beat **`Custom_function_2`** and **`Custom_function_3`**.
4. **`Custom_function_2`** has similar performance with **`Custom_function_1`** and is faster. However, in the internal tests it loses to **`Custom_function_1`**.