Artem Arakcheev
October 6, 2017

Heuristic Analysis

The document describes analysis of three custom heuristic functions. I used the `tournament.py` script for evaluation of the effectiveness of my custom heuristic functions. I measured relative performance of my agent in a round-robin tournament against several other pre-defined agents. The agent uses time-limited Iterative Deepening along with my custom heuristics.

Performance evaluation

I have presented results of execution of the `tournament.py script` below:

```
$ python tournament.py

This script evaluates the performance of the custom_score evaluation
function against a baseline agent using alpha-beta search and iterative
deepening (ID) called `AB_Improved`. The three `AB_Custom` agents use
ID and alpha-beta search with the custom_score functions defined in
game_agent.py.

                        *************************
                             Playing Matches
                        *************************

 Match #    Opponent     AB_Improved    AB_Custom    AB_Custom_2  AB_Custom_3
                         Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1        Random       9  |   1      9  |   1      9  |   1      8  |   2
    2        MM_Open      6  |   4      7  |   3      8  |   2      7  |   3
```

```
    3      MM_Center     8  |   2     9  |   1     6  |   4     7  |   3
    4     MM_Improved    6  |   4     5  |   5     9  |   1     8  |   2
    5       AB_Open      5  |   5     5  |   5     7  |   3     5  |   5
    6      AB_Center     5  |   5     8  |   2     4  |   6     6  |   4
    7     AB_Improved    4  |   6     5  |   5     4  |   6     5  |   5
  --------------------------------------------------------------------------
-
               Win Rate:      61.4%       68.6%       67.1%       65.7%

    Your ID search forfeited 164.0 games while there were still legal moves
available to play.
```

My Custom Heuristic Functions

I used the following ideas of heuristic functions.

AB_Custom

AB_Custom heuristic function calculates value of difference between player legal moves and his opponent doubled legal moves. The difference is multiplied on count of already applied moves. This is the best heuristic function in the analysis. The heuristics has win rate 68.6%.

AB_Custom_2

AB_Custom_2 heuristic is one of the most simple heuristic functions. And it gives nice results. The idea of the heuristics is just get difference between count of player possible moves and count of his opponent possible moves. The heuristics win rate is 67.1%.

AB_Custom_3

AB_Custom_3 heuristics returns value of difference of counts of player and doubled opponent moves multiplied on applied moves count and divided on square of opponent moves plus one. The heuristics win rate is 65.7%

Results of heuristics analysis

The main goal was to develop a heuristic such that outperforms ID_Improved. As we see in results of executing `tournament.py`, the goal is reached. My AB_Custom* heuristics gave out better results (68.6%, 67.1%, 65.7%) than AB_Improved heuristic (61.4%).