Isolation Heuristics Analysis

Aritifical Intelligence Nanodegree

Terence So February 16, 2017

Summary

Three custom players were evaluated for this project: **Aggressive**, **Balanced**, and **Monte Carlo**, each representing the primary strategy of the custom player in game of Isolation. Each player was evaluated using the provided tournament format. A baseline strategy **ID_Improved** was used with a winrate of **60.00%.** The following are the results in summary:

- The Aggressive strategy performed near ID_Improved with a winrate of **63.57%.**
- The Balanced Strategy performed markedly better at a **63.21%** winrate.
- Finally, the Monte Carlo strategy had the best performance with a winrate of **74.29%.**

Evaulation and Analysis

During the course of this project, it was observed that results varied from simulation to simulation. For example, ID_Improved has been observed to have winrates from 55% to about 63% with mixed match results with other players. This holds true with each evaluated strategy. In an effort to increase the accuracy of final score, the tournament format was slightly modified by bumping up the number of matches played from 20 to 40.

ID_Improved

Formula: `player moves - opponent moves`

This is the provided baseline player. The results of one tournament run is as follows:

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 37 to 3
  Match 2: ID_Improved vs   MM_Null     Result: 27 to 13
  Match 3: ID_Improved vs   MM_Open     Result: 20 to 20
  Match 4: ID_Improved vs MM_Improved   Result: 18 to 22
  Match 5: ID_Improved vs   AB_Null     Result: 23 to 17
  Match 6: ID_Improved vs   AB_Open     Result: 22 to 18
  Match 7: ID_Improved vs AB_Improved   Result: 21 to 19

  Results:
  ----------
  ID_Improved         60.00%
```

Aggressive

Formula: `blank spaces - opponent moves`

This strategy was designed as direct counter to the Open strategy. Whereas the Open player plays defensively by favoring board states with more open moves, the Aggressive strategy actively limits the opponent's moves.

The rationale for this strategy is that the horizon effect is particularly pronounced in this version of Isolation since moves are L-shaped 'knight' moves. Unlike regular isolation, each new board position has a completely different set of next possible positions from the previous. So while it may seem like moving to a position with a lot of open moves is a good idea, it may also be a trap where all the next possible moves will be the player's last. Thus, it may be more effective to go on the offensive.

Below is the result of its tournament run:

```
*************************
Evaluating: Student Aggressive
*************************

Playing Matches:
----------
  Match 1: Student Aggressive vs   Random       Result: 36 to 4
  Match 2: Student Aggressive vs   MM_Null      Result: 28 to 12
  Match 3: Student Aggressive vs   MM_Open      Result: 24 to 16
  Match 4: Student Aggressive vs MM_Improved    Result: 16 to 24
  Match 5: Student Aggressive vs   AB_Null      Result: 26 to 14
  Match 6: Student Aggressive vs   AB_Open      Result: 26 to 14
  Match 7: Student Aggressive vs AB_Improved    Result: 22 to 18


Results:
----------
Student Aggressive     63.57%
```

Overall, the results show that the strategy is slightly better than our baseline, at least in this tournament run. Due to the variance in results, it is difficult to conclusively say it performs better than our baseline.

Balanced

Formula: `player moves * (blank spaces - opponent moves)`

This scorer was conceived as a possible improvement over the Aggressive strategy by factoring in open moves similar to ID_Improved. Instead of addition, we use multiplication. The rationale is that the score is really two terms with different units and the player's open move may not be equal in value to the opponent's.

Below is the result of its tournament run:

```
*************************
Evaluating: Student Balanced
*************************

Playing Matches:
----------
  Match 1: Student Balanced vs   Random         Result: 34 to 6
  Match 2: Student Balanced vs   MM_Null        Result: 29 to 11
  Match 3: Student Balanced vs   MM_Open        Result: 17 to 23
  Match 4: Student Balanced vs MM_Improved      Result: 20 to 20
  Match 5: Student Balanced vs   AB_Null        Result: 28 to 12
  Match 6: Student Balanced vs   AB_Open        Result: 24 to 16
  Match 7: Student Balanced vs AB_Improved      Result: 25 to 15


Results:
----------
Student Balanced     63.21%
```

Unfortunately, it did not seem to do much better than the previous strategy, although in some simulations it can reach a winrate of about 67%. Again, we cannot conclusively say it does better than our baseline.

Monte Carlo

The last strategy uses Monte Carlo simulations (MCS) to estimate the viability of a move or board state. Starting from a board state we simulate a game with random moves to the end state and recording if its a win or loss. By running several simulations, we can obtain the ratio of wins over the number of simulations or in essence, a 'probability of winning' for that move. This is good because we can dampen the horizon effect somewhat and give our player a 'blurry' vision of the future.

There were some hurdles getting this to work with the project setup, particularly getting it to work within the constraints of minimax and time limits. Moreover, it didn't seem particularly efficient to use Monte Carlo with Iterative Deepening. After several trials, the optimal strategy seemed to be applying a time limit of **2ms** as well as setting a maximum number of simulations to **50** for evaluating each board state with MCS. The upper bound for simulations doesn't matter as much as long as we set it to a sufficiently high value since the function runs out of time more often than not.

Below is the result of its tournament run:

```
*************************
 Evaluating: Student MCS
*************************

Playing Matches:
----------
  Match 1: Student MCS vs   Random      Result: 36 to 4
  Match 2: Student MCS vs   MM_Null     Result: 34 to 6
  Match 3: Student MCS vs   MM_Open     Result: 25 to 15
  Match 4: Student MCS vs MM_Improved   Result: 27 to 13
  Match 5: Student MCS vs   AB_Null     Result: 34 to 6
  Match 6: Student MCS vs   AB_Open     Result: 27 to 13
  Match 7: Student MCS vs AB_Improved   Result: 25 to 15


Results:
----------
Student MCS         74.29%
```

The result is very encouraging. In other tournament runs, winrates were consistently above 70%. Moreover, it beats every other player by a significant margin in each matchup.

Note to the Reviewer

I should mention that the heuristic interface test will fail. This is because the 'player' parameter that is passed into the function is a string instead of a CustomPlayer and I needed the time_left() function for MCS.