Udacity Artificial Intelligence Nanodegree Report:

Project 3 Building a Game Playing Agent

Author: JiaHsuan Lo

Date: 1/5/2018

# Introduction

The goal of this project is to build a game playing agent to play the isolation game. The project involves code implementation and analysis. The custom game playing agent was implemented using the Minimax and alpha-beta pruning algorithms with iterative deepening. Several heuristic evaluation functions were also implemented and their performance were analyzed. This document describes the main ideas of the heuristic evaluation functions and reports the analysis results.

# Heuristic Evaluation Functions

### Difference of Liberties (CustomPlayer.score\_moves)

This is the baseline heuristic that was introduced in the class. It calculates the difference in liberties between current player and opposing player, i.e. number of current player’s liberties – number of opposing player’s liberties.

### Close to Center (CustomPlayer.score\_center)

This heuristic is based on intuition that it is more likely to have space to operate when trying to stay closer to center of the board. In this function, the distances between players and the board center are calculated, then the difference in the distance-to-center between the current and opposing players are computed: score= distance-to-center of opposing player – distance-to-center of current player.

### Difference of Double Liberties (CustomPlayer.score\_moves2)

It is similar to the baseline heuristic. But now this evaluation function accounts for the liberties of the current location as well as the liberties of the liberties. As shown in Figure 1, the dark blue circle is the current location. The light blue circles are the liberties and the orange circles are the liberties of the light blue circles. If all the light blue and orange circles are defined as double liberties, the score is then defined as the number of current player’s double liberties – number of opposing player’s double liberties.

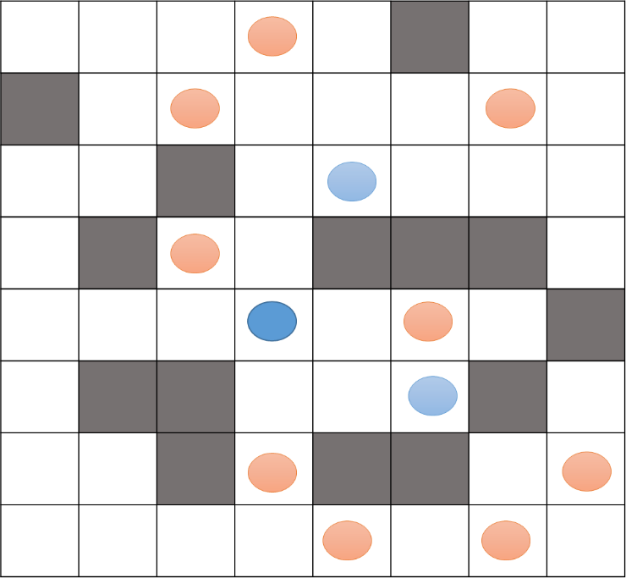


Figure 1: illustration of the double liberties. The deep blue circle is the current location. Light blue circles are the liberties of the current location. Orange circles are the liberties of the light blue circles. The dark squares are the closed locations.

### Progressive Heuristic (CustomPlayer.score\_progression)

This evaluation function combines the above three heuristics. The main idea is to keep the object close to the center initially (using score\_center), and then use moves and moves2 heuristics as the game progresses. The setup is:

If number of open cells >80 then score= score\_center()

Else-if number of open cells>60 then score= score\_move()

Else number of open cells then score= score\_moves2()

# Experiment Setup and Results:

Initially for the depth limit of one, each of the four heuristic variant of the custom player played 100 games against each of the Random, Greedy, and Minimax agents. Therefore a total of 4 x 3 x 100= 1200 games were played for the depth limit of one. Then the depth limit was increased by a increment of one, and same number (1200) of games were played for each depth limit setting, until the game playing agent was unable to finish the moves within time limit (150 ms). The fair flag was turned on during the runs.

The win percentage against the opposing game agents (Random, Greedy, Minimax) are summarized in the following table:

Table 1: Summary of experiment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Custom Agent Settings | | Winning Rate Against Opponent | | |
| Depth Limit | Eval Heuristic | Random | Greedy | MiniMax |
| 1 | **Moves (baseline)** | **84%** | **46%** | **25%** |
| Center | 82% | 43% | 23% |
| Moves2 | 88% | 69% | 44% |
| Progressive | 95% | 76% | 43% |
| 2 | **Moves (baseline)** | **86%** | **53%** | **31%** |
| Center | 74% | 53% | 26% |
| Moves2 | 94% | 70% | 51% |
| Progressive | 94% | 69% | 42% |
| 3 | **Moves (baseline)** | **95%** | **62%** | **50%** |
| Center | 90% | 66% | 35% |
| Moves2 | 96% | 80% | 68% |
| Progressive | 98% | 80% | 62% |
| 4 | **Moves (baseline)** | **96%** | **76%** | **51%** |
| Center | 91% | 70% | 42% |
| Moves2 | over time limit | over time limit | over time limit |
| Progressive | 98% | 88% | 66% |
| 5 | **Moves (baseline)** | **93%** | **83%** | **71%** |
| Center | 92% | 71% | 42% |
| Moves2 | over time limit | over time limit | over time limit |
| Progressive | 100% | 93% | 80% |

Some observations on the results table are listed below:

* Against Minimax agent, the “Center” heuristic has similar accuracy as the baseline heuristic when the iteration depth is low (Center: 23% vs Baseline: 25%) but is much less favorable as the iteration depth increases.
* Against Minimax agent, the “Moves2” heuristic has better accuracy than the baseline heuristic regardless of the iteration depth. However, the maximal depth it can achieve without exceeding the 150 ms time limit is three, while the baseline heuristic (Moves) is able to achieve a depth limit of five.
* Against Minimax agent, the “Progressive” heuristic also out-performs the baseline and it can achieve the same iteration depth as the baseline heuristic (depth= 5).

# Questions and Answers

#### What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

The Center heuristic uses current location and attempts to keep the location as close to the board center as possible. It is just based my experience in playing the game. By occupying the center of the board usually (not always) leads to more room to operate. At a low iteration depth, it performs similarly to the baseline (Moves) heuristic. Another advantage is that it is not computationally expensive and can achieve the same maximal depth as the baseline.

The Moves2 heuristic uses the liberties of the current location and the next level of liberties to predict the game result. It essentially provide one more step of prediction. Therefore, it works favorably than the baseline at any depth.

The Progressive heuristic combines the “Center”, “Moves”, and “Moves2” heuristic. The idea is when the game is at the beginning stage, the “Center” heuristic forces the player to occupy the center region of the board to get advantage. As the game goes on, the custom agent transitions to Moves and Moves2 strategies to increase prediction accuracy. This Progressive heuristic also improves the run time since the most computationally expensive Moves2 is only employed when the game is close to the end. So it can achieve to the same depth-limit (5) as the baseline heuristic.

#### Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

The “Center” and “Progressive” heuristic achieves a search depth of 5 (same as the baseline), while the “Moves2” heuristic can only achieve a depth of 3.

The “Moves2” heuristic is the most accurate since it can achieve higher winning percentage than others, as long as within the time limit. On the other hand, the “Progressive” heuristic provides faster run speed, and thus is able to achieve the same depth as baseline with superior winning percentage (Progressive: 80% vs Baseline: 71%, against Minimax agent). So it is a balanced strategy between speed and accuracy to achieve the better winning percentage.