

# AIND Isolation game playing agent report

Andrzej Wytyczak-Partyka

December 5, 2017

The task was to implement an agent playing the Isolation game, using Minimax with alpha-beta pruning and iterative deepening. The agent would use one of 3 heuristics for assessing current board state during termination of each search. Termination happens either when end-game is reached or when the search reaches specified depth.

The heuristic implemented were:

- AB\_Custom - For first 5 moves - use the distance from center heuristic, Later in the game - use the number of my moves - number of opponent moves
- AB\_Custom\_2 - For first 40 moves - use distance between opponent's pieces, later in the game - use the square of number of available moves
- AB\_Custom\_3 - For first 3 moves - assign High score for center move, later assign higher points to locations near the center, finally use square of number of available moves

I examined the results of 20 runs of the tournament.py script (shown in Figure 3) and ran simple statistical analysis, as shown in Figure 1.

The result of the analysis (as seen in Figure 2) confirms that AB\_Custom\_3 is far worse than AB\_Improved, as well as that AB\_Custom and AB\_Custom\_2 are not statistically different from AB\_Improved.

```

print("h0 is that AB_Improved has the same performance as AB_Custom")

[stat,pvalue] = stats.ttest_ind(df['AB_Improved'], df['AB_Custom'], 0)
if(pvalue < 0.05):
    print('h0 rejected - AB_Improved does not have the same performance as AB_Custom')
else:
    print('h0 accepted - AB_Improved does have the same performance as AB_Custom')

print("h0 is that AB_Improved has the same performance as AB_Custom_2")

[stat,pvalue] = stats.ttest_ind(df['AB_Improved'], df['AB_Custom_2'], 0)
if(pvalue < 0.05):
    print('h0 rejected - AB_Improved does not have the same performance as AB_Custom_2')
else:
    print('h0 accepted - AB_Improved does have the same performance as AB_Custom_2')

print("h0 is that AB_Improved has the same performance as AB_Custom_3")

[stat,pvalue] = stats.ttest_ind(df['AB_Improved'], df['AB_Custom_3'], 0)
if(pvalue < 0.05):
    print('h0 rejected - AB_Improved does not have the same performance as AB_Custom_3')
else:
    print('h0 accepted - AB_Improved does have the same performance as AB_Custom_3')

```

Figure 1: Python code to determine if heuristics differ from AB\_Improved

```

h0 is that AB_Improved has the same performance as AB_Custom
h0 accepted - AB_Improved does have the same performance as AB_Custom
h0 is that AB_Improved has the same performance as AB_Custom_2
h0 accepted - AB_Improved does have the same performance as AB_Custom_2
h0 is that AB_Improved has the same performance as AB_Custom_3
h0 rejected - AB_Improved does not have the same performance as AB_Custom_3

```

Figure 2: Results of t-test verification

| #  | AB_Improved | AB_Custom | AB_Custom_2 | AB_Custom_3 |
|----|-------------|-----------|-------------|-------------|
| 0  | 52.9        | 60.0      | 61.4        | 60.0        |
| 1  | 61.4        | 57.1      | 67.1        | 55.7        |
| 2  | 67.1        | 62.9      | 70.0        | 55.7        |
| 3  | 61.4        | 65.7      | 64.3        | 50.0        |
| 4  | 61.4        | 54.3      | 60.0        | 57.1        |
| 5  | 68.6        | 64.3      | 64.3        | 64.3        |
| 6  | 70.0        | 65.7      | 74.3        | 62.9        |
| 7  | 65.7        | 62.9      | 61.4        | 64.3        |
| 8  | 67.1        | 60.0      | 64.3        | 55.7        |
| 9  | 67.1        | 57.1      | 54.3        | 58.6        |
| 10 | 67.1        | 57.1      | 62.9        | 58.6        |
| 11 | 62.9        | 62.9      | 55.7        | 55.7        |
| 12 | 64.3        | 67.1      | 58.6        | 52.9        |
| 13 | 71.4        | 61.4      | 58.6        | 60.0        |
| 14 | 67.1        | 61.4      | 67.1        | 64.3        |
| 15 | 60.0        | 61.4      | 58.6        | 61.4        |
| 16 | 54.3        | 58.6      | 60.0        | 60.0        |
| 17 | 61.4        | 65.7      | 64.3        | 55.7        |
| 18 | 64.3        | 68.6      | 70.0        | 65.7        |
| 19 | 61.4        | 60.0      | 70.0        | 64.3        |

Figure 3: Percent of games won during 20 runs of ‘python tournament.py’

| #    | Custom vs Improved | Custom_2 vs Improved | Custom_3 vs Improved |
|------|--------------------|----------------------|----------------------|
| 0    | -7.1               | -8.5                 | -7.1                 |
| 1    | 4.3                | -5.7                 | 5.7                  |
| 2    | 4.2                | -2.9                 | 11.4                 |
| 3    | -4.3               | -2.9                 | 11.4                 |
| 4    | 7.1                | 1.4                  | 4.3                  |
| 5    | 4.3                | 4.3                  | 4.3                  |
| 6    | 4.3                | -4.3                 | 7.1                  |
| 7    | 2.8                | 4.3                  | 1.4                  |
| 8    | 7.1                | 2.8                  | 11.4                 |
| 9    | 10.0               | 12.8                 | 8.5                  |
| 10   | 10.0               | 4.2                  | 8.5                  |
| 11   | 0.0                | 7.2                  | 7.2                  |
| 12   | -2.8               | 5.7                  | 11.4                 |
| 13   | 10.0               | 12.8                 | 11.4                 |
| 14   | 5.7                | 0.0                  | 2.8                  |
| 15   | -1.4               | 1.4                  | -1.4                 |
| 16   | -4.3               | -5.7                 | -5.7                 |
| 17   | -4.3               | -2.9                 | 5.7                  |
| 18   | -4.3               | -5.7                 | -1.4                 |
| 19   | 1.4                | -8.6                 | -2.9                 |
| sum: | 42.7               | 9.7                  | 94.0                 |

Figure 4: Difference between Custom heuristics and AB\_Improved heuristic