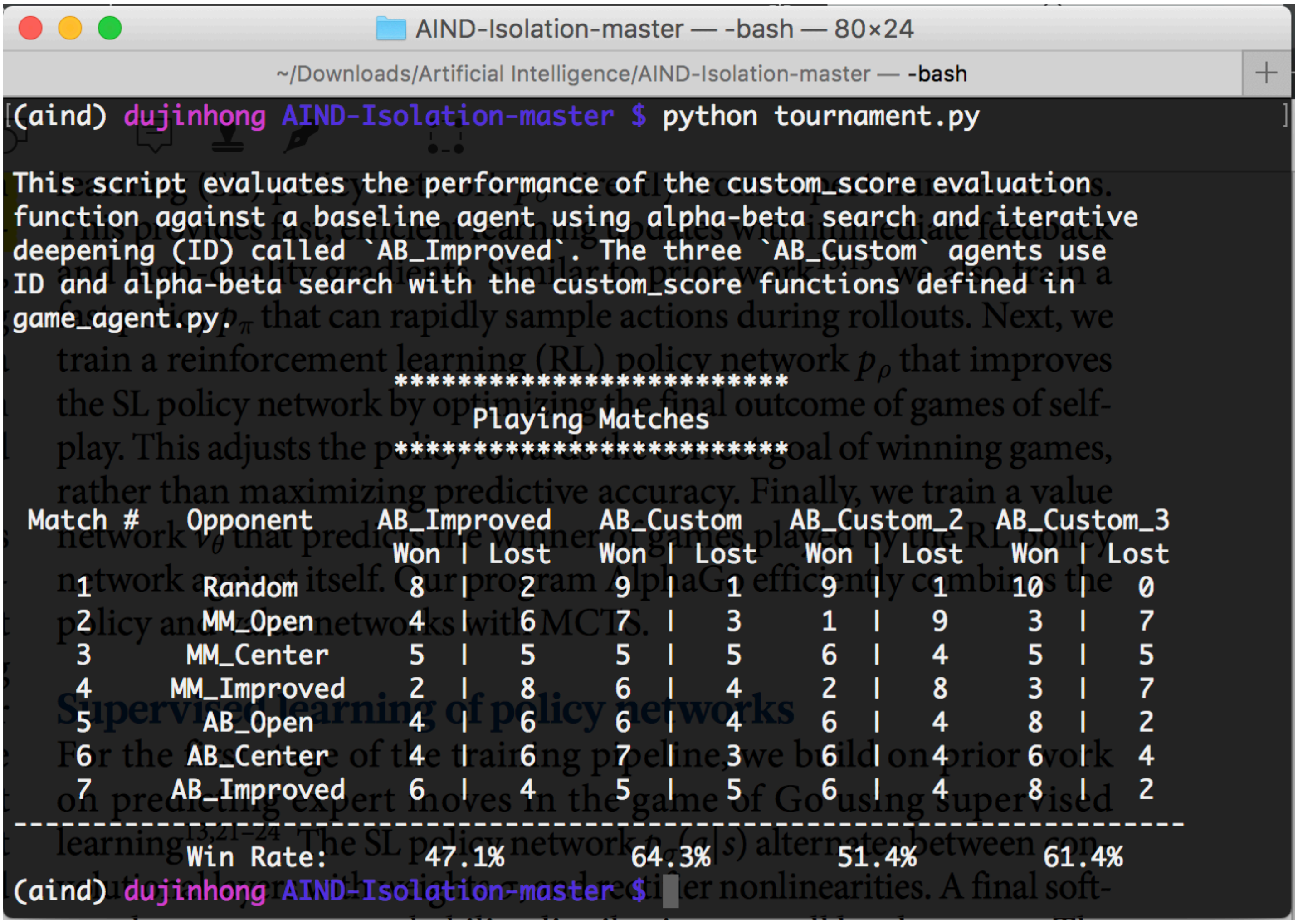
Heuristic Analysis

Rssult


AIND-Isolation-master — -bash — 80×24
~/Downloads/Artificial Intelligence/AIND-Isolation-master — -bash

```
(aind) dujinhong AIND-Isolation-master $ 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       8  |   2     9  |   1     9  |   1    10  |   0
    2       MM_Open      4  |   6     7  |   3     1  |   9     3  |   7
    3      MM_Center     5  |   5     5  |   5     6  |   4     5  |   5
    4     MM_Improved    2  |   8     6  |   4     2  |   8     3  |   7
    5       AB_Open      4  |   6     6  |   4     6  |   4     8  |   2
    6      AB_Center     4  |   6     7  |   3     6  |   4     6  |   4
    7     AB_Improved    6  |   4     5  |   5     6  |   4     8  |   2
--------------------------------------------------------------------------
           Win Rate:      47.1%        64.3%        51.4%        61.4%
(aind) dujinhong AIND-Isolation-master $
```

We can see the three heuristic function works better than the `AB_Improved` on almost time.

Analysis

- Custom 1

Here I use the piecewise function, using different stages in different situations.

1.When one of the player has won, just return as follows.

```
if game.is_loser(player):
    return float("-inf")

if game.is_winner(player):
    return float("inf")
```

2.When the game just has begun for a short time,

$$\frac{blank\_spaces}{total\_spaces} \leqslant \frac{1}{3}$$

I would like to force my player to conquer the center of the blank areas. It is because when there are many blank positions on the board, it will be more chioces for the players on such areas instead of the edges of the board. I use the following fomula to calculate the value of the heuristic function at this stage.

$$d = \|x - C\|_2^2$$

We can call it `improve_center_score`.

3.When the game proceeds for a while,

$$\frac{1}{3} < \frac{blank\_spaces}{total\_spaces} \leqslant \frac{2}{3}$$

I would like to combine the `improved_score` and the `improve_center_score`. Using

$$own\_moves - 2 \times opp\_moves + \frac{d}{total\_spaces}$$

to be the value of the heuristic function at this stage. Here the secend term plus by 2 because I want it to be more offensive, the third term divides by `total_spaces` because I want to reduce the influence of the `improve_center_score`, and I still keep it because when the former score give the same value on 2 different positions, the player can choose the better position.

We can call it `combined_improved_center_score`.

4.When the space of the board is almost run out,

$$\frac{2}{3} < \frac{blank\_spaces}{total\_spaces} \leqslant 1$$

the `improved_center_score` is no more work. Now I only want my players to survive and use the strategies of defend, so we only focus on the `open_move_score`

$$own\_moves$$

```
center = np.mean(game.get_blank_spaces())
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
if len(game.get_blank_spaces()) <= game.width*game.height/3:
    return np.sum((game.get_player_location(player) - center )**2)
elif len(game.get_blank_spaces()) <= game.width*game.height/3*2:
    return float(own_moves) - 2 * float(opp_moves) + np.sum((game.get_player_location(player) - center )**2)/game.width/game.height
else:
    return float(own_moves)
```

- Custom 2

1.When one of the player has won, just return.

2.When

$$\frac{blank\_spaces}{total\_spaces} \leqslant \frac{1}{2}$$

use the `combined_improved_center_score` to conquer best positions.

3.When

$$\frac{1}{2} < \frac{blank\_spaces}{total\_spaces} \leqslant 1$$

take a more conservative strategies an use

$$own\_moves - 0.8 \times opp\_moves$$

as the value of the heuristic function at this stage.

- Custom 3

1.When one of the player has won, just return.

2.When

$$\frac{blank\_spaces}{total\_spaces} \leqslant \frac{1}{2}$$

use the `combined_improved_center_score` to conquer best positions.

3.When

$$\frac{1}{2} < \frac{blank\_spaces}{total\_spaces} \leqslant 1$$

use

$$own\_moves + d$$

as the value of the heuristic function at this stage.