Build a Game-Playing Agent: Heuristic Analysis
Omar Ghaleb

1. Introduction

This paper will present three evaluation functions or heuristics for winning the isolation game. The evaluation is based on the tournament performance.

2. Heuristics

2.1. Heuristic 1: Agent moves vs. opponent moves

The *ID_Improved* heuristic uses the function:

$$score = my\_moves - opponent\_moves$$

This evaluation function can be improved by assigning weights to the opponents moves or my moves, so we assigned weights to the opponents moves to give more weight to the moves available that gives the opponents less moves. The new function:

$$score = my\_moves - 3 * opponent\_moves$$

The weight value can be different and we can change how early the cutoff of the tree is, but in this we will just use one value which is 3.

2.2. Heuristic 2: Agent position vs. opponent position

This evaluation function uses the position of the agent and the opponent to evaluate the moves available for the agent. The moves that distance the agent from the opponent gets a higher score better than the one that keeps him closer. The evaluation function is as follow:

$$score = my\_position - opponent\_position$$

2.3. Heuristic 3: Using both the position and number of moves

This heuristic uses both the first and the second heuristic to improve the score function. This allows the agent to take into consideration the number of moves available for it and at the same time, evaluating them according to the distance from the opponent. The function is as follows:

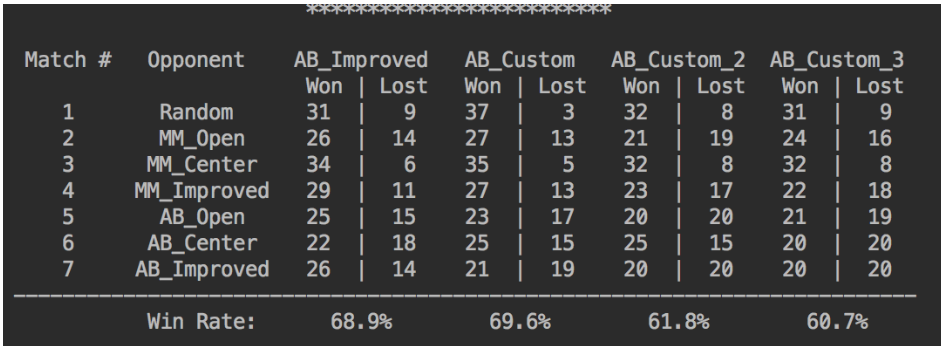
$$distance = my\_position - opponent\_position$$

$$score = (my\_moves + distance) - 3 * opponent\_moves$$

2.4. Results

To evaluate the performance of the heuristic, the tournament was used to compare the win rate. In the results, *AB_custom* is H3 which uses the combination of H1 and H2. *AB_custom_2* is H2. *AB_custom_3* is H1.

The performance rates obtained are shown in this graph:



```
                         **************************

 Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random      31  |   9    37  |   3    32  |   8    31  |   9
    2       MM_Open     26  |  14    27  |  13    21  |  19    24  |  16
    3      MM_Center    34  |   6    35  |   5    32  |   8    32  |   8
    4     MM_Improved   29  |  11    27  |  13    23  |  17    22  |  18
    5       AB_Open     25  |  15    23  |  17    20  |  20    21  |  19
    6      AB_Center    22  |  18    25  |  15    25  |  15    20  |  20
    7     AB_Improved   26  |  14    21  |  19    20  |  20    20  |  20
--------------------------------------------------------------------------
           Win Rate:      68.9%        69.6%        61.8%        60.7%
```

In the graph, we can see that the mixed heuristic, which is *AB_custom*, outperforms the other two heuristics by almost 9 % increase in the win rate. In *AB_custom_2*, which is using only the number of moves, the win rate is 61.8 %. And in *AB_custom_3*, which uses the position of the agent and the opponent, the win rate is even lower with 60.7 %. After combining both to create the first heuristic *AB_custom*, we can see a huge increase in the win rate to obtain a 69.6 % win rate.

3. Conclusion

In this report, three heuristics were introduced:

- **My Moves vs Opponent Moves:** that shows better performance than ID_Improved sometimes.
- **My Position vs Opponent Position:** This uses the distance between the two players to evaluate a move.
- **Mixed:** This uses both the first and second heuristics.

From the results, we can see that the mixed heuristic is the recommended one according to the following reasons:

- It is built on the bases of the number of moves which is considered already a good heuristic.
- Using a weight will make the agent make aggressive to cut the search trees as early as possible.
- The results obtained in experimentation shows a better performance against *ID_Improved*
- Mixing both the moves and the positions heuristics, make use of the advantages of both.

Due to randomness, sometimes the results fluctuate being better or worse than *AB_improved*.