Saurabh Vyas
Udacity Artificial Intelligence Nanodegree
05-Feb-18

Evaluating Game Playing Heuristics

Project 2: Build a Game-Playing Agent

I evaluated three basic heuristics that can be used to score in Isolation game.

1. Own moves (AB_Custom) : This heuristic calculates the number of moves available to the player. The basic idea is to maximize the number of moves available for the player so that there are more options to choose from.
2. Difference of own moves and opponent moves (AB_Custom_2): This heuristic focuses on maximizing difference between own moves and opponent moves. This makes more sense intuitively as the end goal to make sure that opponent runs out of moves.
3. Weighted difference of own moves and opponent moves (AB_Custom_3): This option evaluates if applying a different multipliers on own moves and opponent moves and then subtracting them leads to a better function. For this heuristic, the different multiplier were tried such as:
   - OwnM – 1.5*OppM
   - OwnM – 0.5*OppM (found to be best performing in several trials)
   - OwnM – (OppM)**0.5

The following are the results of tournament.py. Since the results of the game could vary based on opponent's first move, I tried running tournament.py several times to ensure results are reliable.

```
************************
     Playing Matches
************************
```

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	8	2	10	0	10	0
2	MM_Open	9	1	3	7	8	2	10	0
3	MM_Center	7	3	10	0	9	1	10	0
4	MM_Improved	4	6	4	6	8	2	9	1
5	AB_Open	5	5	5	5	5	5	7	3
6	AB_Center	6	4	3	7	5	5	4	6
7	AB_Improved	5	5	3	7	5	5	5	5
	Win Rate:	64.3%		51.4%		71.4%		78.6%	

```
***************************
       Playing Matches
***************************
```

Match #	Opponent	AB_Improved Won	Lost	AB_Custom Won	Lost	AB_Custom_2 Won	Lost	AB_Custom_3 Won	Lost
1	Random	9	1	8	2	10	0	10	0
2	MM_Open	6	4	7	3	6	4	7	3
3	MM_Center	8	2	8	2	8	2	10	0
4	MM_Improved	7	3	4	6	5	5	7	3
5	AB_Open	3	7	3	7	7	3	5	5
6	AB_Center	5	5	2	8	5	5	4	6
7	AB_Improved	3	7	2	8	7	3	6	4
Win Rate:		58.6%		48.6%		68.6%		70.0%	

```
***************************
       Playing Matches
***************************
```

Match #	Opponent	AB_Improved Won	Lost	AB_Custom Won	Lost	AB_Custom_2 Won	Lost	AB_Custom_3 Won	Lost
1	Random	8	2	10	0	10	0	9	1
2	MM_Open	7	3	5	5	8	2	10	0
3	MM_Center	10	0	7	3	8	2	8	2
4	MM_Improved	8	2	5	5	6	4	7	3
5	AB_Open	6	4	7	3	6	4	6	4
6	AB_Center	8	2	4	6	3	7	5	5
7	AB_Improved	4	6	3	7	4	6	6	4
Win Rate:		72.9%		58.6%		64.3%		72.9%	

Recommendation:

The above results show that the third heuristic performs significantly better than the other two. This leads to the conclusion that an objective function on the below lines could be a very good choice:

$F(x) = a*OwnMoves - b*OppMoves$

We can try to find the optimum values of 'a' and 'b' by simulation.

The reason why this function works well could be due to the following:

1. It tries to maximize the own moves
2. It also applies a penalty to those moves that give large number of options to the opponent
3. The penalty could be optimized further using simulations, which would lead to even better results.