heuristic_analysis.MD

In this project, the heuristic function is used to evaluate the game result when the searching tree reaches the maximum depth.

My heuristic is this:

```
score = a * own_moves^2 / (1 + opp_moves) + b * own_moves / (1 + opp_moves^2)
```

which own_moves stands for the legal moves of current player, opp_moves stands for that of opponent, "^" stands forpower and a,b are hyperparameters to tune.

To get the best choice of parameters a and b, I used an optimization algorithm called "twiddle" which I learned from course "artificial intelligence for robotics"(in "tournament.py", line 188-221). In each iter of twiddle, a new heuristic function(or a new sets of parameters of the function defined above) is evaluated and the result is compared with the previous result in order to update the parameters and the step sizes of parameter update.

Then after 10 iters of twiddle, the best parameters chosen are:

```
a = 1.001
b = 1.180
```

And with this set of parameters, my agent reached a win rate of 77.14%, which is higher than the "ID_Improved" heuristic with 66% win rate.