Introductory document

AlgoPoker

March 10, 2022

Contents

Introduction

This document is a brief guide of the rules & structure of the competition, as well as some ideas for basic bots to get started.

1 Structure of the competition

The competition will run in two main stages:

1. The initial qualification round. This round will be used to determine who progresses to the final round.
2. The final round. This will be a subsequent round for those who do sufficiently well in the initial round, and the prizes will be awarded to those who do well in this round.

Each round will follow the same basic structure:

- The basic idea of the competition is to program a bot which can play heads-up no-limit Texas hold 'em poker.
- Each team will submit a bot and every pair of bots will play multiple hands (>1000) against one another.
- After all games have been played, the bots will be ranked by their total winnings. This will determine who proceeds to the next round (in the initial round) and who wins prizes (in the final round).

2 Rules

Here are the basic rules for the competition and these should be followed at all times by the contestants. The CU Algorithmic Games society reserves the right to **disqualify any team** found to be in breach of these

1. Plagiarism of large parts of preexisting code from sources including but not limited to the internet and books is prohibited (however, using code from the templates uploaded by the CU Algorithmic Games society is allowed). Research of poker strategies themselves are allowed and encouraged.

2. Bots are not allowed to access the internet.
3. No collaboration/collusion between bots. This includes any two bots working together in any way, such as purposefully losing against another bot to give away edge.
4. All bots must be submitted in Python.
5. Teams may have no more than 4 members.
6. No offensive team names.

3 Examples of strategies

Here are a few ideas and examples of strategies to get you started. They are not very good on their own, but hopefully they will inspire some better ones.

Strategy 1: Pair-hunting

On the flop, if the bot is dealt a pair (or any better hand), it goes all in. Otherwise the bot immediately folds. [Hopefully it is clear that this is a bad strategy and easily exploitable.]

Strategy 2: Monte-carlo simulation

On the flop, simulate games to figure out the relative strength of your hand. From here estimate the optimal bet size based on the limited information you have on the flop.

Strategy 3: Pair-hunting 2.0

Implement the exact same strategy as strategy 1, but this time if you get a pair, continually raise your bet by some fixed proportion $r > 2$ at each round of betting; i.e at each round, if the current bet size is a, you raise your bet to ar.

Strategy 4: A (not very good) game theoretic approach

Fold whenever your opponent raises and go all in when they check. [Note that this would actually work quite well against specific strategies, for example strategy 1 above.]

Strategy 5: A mixed strategy

Fix some $p_1, p_2, p_3, p_4 \in [0, 1]$ such that $\sum_{i=1}^{4} p_i = 1$ and at the flop, choose strategy i from the above with probability p_i.

4 Useful resources

Here are some resources which you may find useful

- eval7 python library - allows for various poker odds calculations through python.
- `https://www.pokerhandrange.com/`
- `https://github.com/algopoker` - templates and more useful material will be uploaded here.