Three-Person No-limit Texas Hold’em Bot
Preliminary Report

Peter Chapman  Yan Huang
Brielin Brown  Sahnghyun Cha  Matthew Yu

Current Exploration

We have adapted some of our previous West Virginia bots to work with the three-person no-limit Texas Hold’em rules. These bots are simple hand-strength-based probabilistic programs. That is, actions are derived from a probabilistic triple generated from the hand-strengths calculated at different stages of betting. They all beat random raise/call bots by a large margin.

During our experiments we made an interesting observation: the included pseudo-random number generator does not seem to create random card deals. When we face 3 identical bots together in a game to play, for example, 10000 matches, we found specific seeds favor one table position over another. It would be expected that as the number of games increases indefinitely, each bot would approach breaking even, but this does not seem to be the case. Hence, to facilitate evaluating a bot, we developed a shell script that cycles through all 6 possible arrangements and tallies the game results. After this step, program improvements become evident from consistent winnings and losings reported by the script. We are happy to share our test environment and suggest all other teams use a similar evaluation technique.

We also noticed that the third opponent is very influential on the final results of a game. This is observed by selecting different combinations bots to play the game. The lose-win results are far from linear. Obviously, there is big opportunity to be exploited from a good opponent modeling algorithm.

Preliminary Bot  Bot_e is an example bot implemented to test the hand strength function from Darse Billing’s paper: Opponent Modeling in Poker. This function was designed to compute the probability of holding the best hand among Post-Flop active players. For the Pre-Flop, we decided to use “Apu’s Pre-Flop Folding Strategy”\(^1\) to determine appropriate situations to fold. The pre-flop strategy was based on the statistics of 4 million simulated games. It seems to be a very efficient way to making decisions in the Pre-Flop stage, as indicated by our ability to beat a random player with this strategy in the Pre-Flop and always calling until the Showdown. After the Pre-Flop, we use the hand strength function to decide the next move. Bot_e looks up the result of the function in every stage, and we fold if the probability is lower than the current pot odds. Moreover, we raise if the probability is greater than 75%. Otherwise, we make a call. For bluffing, 10% of folding decisions are randomly selected and changed to a raise. Figure 1 shows how our newest bot Bot_e performs against adapted West Virginia Hold’em bots 0.1 and 0.3.

<table>
<thead>
<tr>
<th>Bot 0.1</th>
<th>Bot_e</th>
<th>Bot 0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>-71057.5</td>
<td>119253.5</td>
<td>-48194</td>
</tr>
</tbody>
</table>

Figure 1: Winnings over 25,000 hands.

\(^1\)https://www.cs.indiana.edu/~kapadia/nofoldem/strategy_total.html
**Future Directions**

We are considering extending our pod odds tree strategy which plays well in the 2-player West Virginia Hold’em to work with 3-player Texas Hold’em. To this end, we need more sophisticated code to automate the tree generation and the process of decision making. We will introduce many more parameters as build the 3-player tree, especially for pre-river betting rounds (e.g. each opponents probability of calling, raising, and hand-potential). We envision a Monte Carlo simulation-based exploration could strengthen our knowledge to calculate an accurate tree.

**Working Plan**

We will collaborate closely to discuss ideas for good poker strategies and help each other solving implementation problems. However, we would also like to subdivide our group into smaller teams of 1-2 people to work on a clear idea and realize it into working bots. Then, we can have internal bot matches to identify the best bots or integrating interesting ideas into a new bot.