

## River Monster Bot

Paul Calvetti and Cole Kingston

### Introduction:

The main idea behind our bot was to have a base strategy that could be varied in game to adjust to specific opponents. We achieved this utilizing a wide range of features that will be outlined below. We added features to our bot throughout the competition, some of which improved the bot and some of which didn't improve things much. Here we will highlight the main features and changes for each of our weekly tournament bot submissions.

### Week 1 Bot:

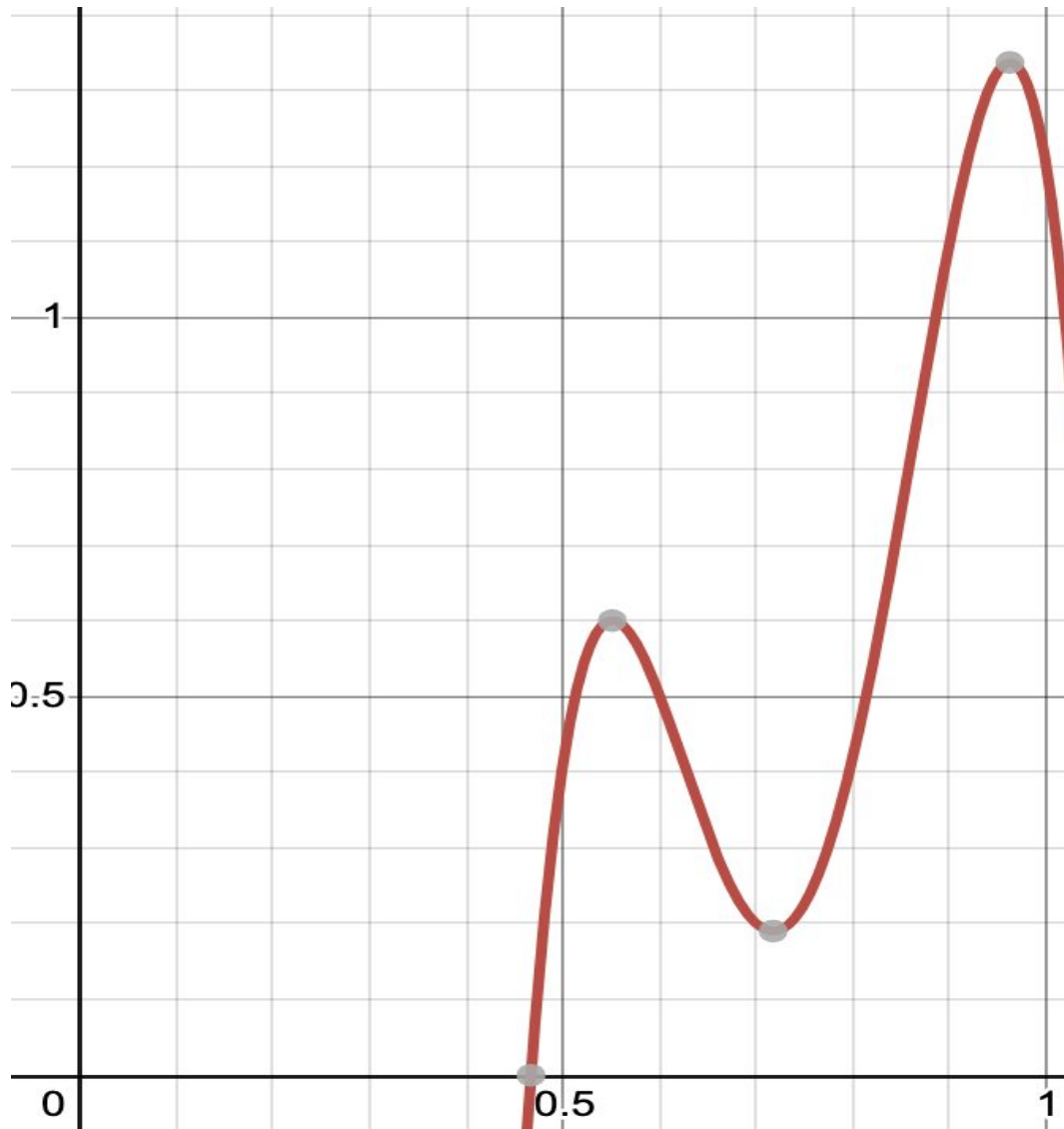
For this bot we made our own card allocation strategy. The idea was that we wanted to maximize the strength we had on board 3, then board 2, and lastly board 1. We thought it would be more valuable to have a strong hand on board 3 because that board was inflated by the most number of chips. We allocated by increasing the strength on board 3 by a factor of 3, board 2 by a factor of 2 and board 1 by a factor of 1. The logic behind this is that due to the larger inflation of the pot on higher numbered boards, the pots would be larger and thus we would want stronger hands on those boards. Due to this logic we chose to allocate out cards by maximizing the following expression:

$$\max[3 * \text{board\_3\_strength} + 2 * \text{board\_2\_strength} + 1 * \text{board\_1\_strength}]$$

Besides this, we did not make many changes to the bot, we kept the staff logic for when to call bets and used the staff strategy for betting.

### Week 2 Bot:

This bot we began to implement changes to the staff betting strategy. We implement a non-linear bet sizing function. The function is shown below with strength on the x-axis and bet size (as a percentage of the pot) on the y-axis.



### Week 3 Bot:

For this bot we began to fine tune our strategy for when to call bets. In order to do this we used a variable intimidation factor. Without going into too much detail, our

intimidation factor increased with the number of bets an opponent had made and with the size of the most recent bet. Additionally we would be more intimidated by bets from an opponent who raised less often, and less intimidated by a bet from an opponent who bet often. We thought this strategy would allow us to exploit the tendencies of both passive and aggressive opponents to our advantage.

#### Final Bot:

For this bot we made changes to our preflop betting strategy. We added a strategy that would raise more often from the small blind because we would have an ideal post - flop position. Additionally, we added features that would raise more often when opponents showed weakness. Lastly, we tuned many variables that determined bet amounts and fold sensitivities.

#### **Card Allocation:**

For our card allocation, we first found our hole card strengths for all possible pairs and allocated them to have the best three hands possible. For each possible allocation, we summed the hand strengths for each of the hole cards and put them into a list. We then sorted this list to get the highest possible sum with the hole cards ranked in order of strength. At first, we put our strongest holes on the board with highest inflation, but after trying other allocations, we found that putting our strongest on board 2, middle on board 1, and worst on board 3. This allowed us to have a big advantage against our old strategy which we figured many of the teams were using. We also thought this would work because our weakness would only be against a 3, 1, 2

allocation which does very poorly against the perceived normal allocation(1, 2, 3). Also, we found that the pot inflation has little effect on the outcomes because the betting on each board usually dwarfs the pot inflation. In the end, we realized that if we used the same allocation, a team may pick up on it and change their allocation to consistently beat us. To combat this, we start with the (2, 3, 1) allocation for the first 20 hands, then try to detect if they are using a specific allocation on each hand or if it is random. If its random, then we also go random because we don't want them to use our allocation against us. Otherwise, we adjust our allocation strategy to beat two of their hands constantly. Overall, if the opponent gave away their allocations, then we adjust to beat it, otherwise we go random with our allocations.

```
        best_allocation = allocation_strength_list
    if self.opp_board_avg_strength_ranking is not None:
        count = 0
        for board in self.opp_board_avg_strength_ranking:
            self.board_allocations[board] = best_allocation[(count + 1) % 3][0]
            self.hole_strengths[0][board] = best_allocation[(count + 1) % 3][1]
            count += 1
    else:
        self.board_allocations = [best_allocation[1][0], best_allocation[2][0], best_allocation[0][0]]
        self.hole_strengths[0] = [best_allocation[1][1], best_allocation[2][1], best_allocation[0][1]]
```