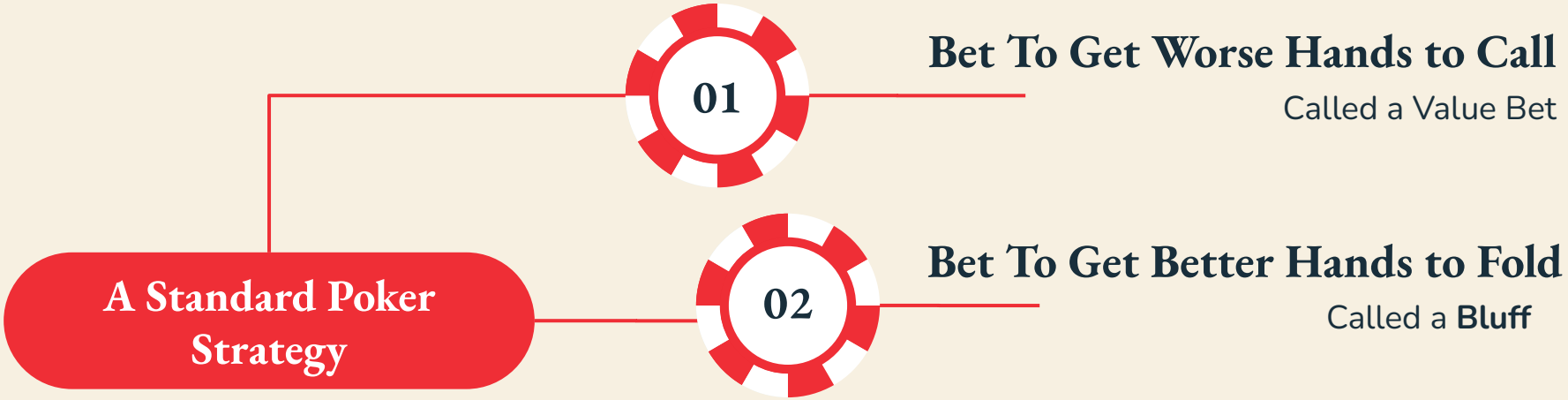
Understanding Push-Fold Pre-Flop Strategy for Heads Up NLH Poker

Robbie Selwyn
CS238 Final Project, Winter 2023

What is Poker?

- Poker is a card game where players are dealt cards, and bet across multiple rounds based on the strength of their hand with the goal to win the most amount of money.
- Poker has many variants, including No Limit Texas Hold' Em (one of the most popular), Pot Limit Omaha (PLO), Limit Texas Hold' Em, as well as many others.
- There have been many attempts to solve poker. In 2007, the Counterfactual Regret Minimization (CFR) algorithm was published [1], which has proven to be the standard algorithm for solving games like Poker. More recently, researchers at CMU built Libratus [2], which applied the CFR algorithm (along with other AI techniques) to try to build the best heads-up poker bot in existence.
- Traditional algorithms like MCTS perform poorly for creating optimal strategy.

Basic Poker Strategy



Why does this matter?

Understanding Mathematically Optimal Poker

Nash Equilibrium

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \text{ for all } s_i \in S_i$$

Problem

If we always bet when we have a good hand and fold when we have a bad hand, it will be too easy for our opponent to realize and outplay us!

Solution

Value hands (made hands) must be equally balanced out with bluff hands (hands where we bet to try to get opponents to fold)

A Simple Probabilistic Strategy: Push/Fold against SB Open

- Before the flop, we are either going to fold, or go all in with 10BB against our opponent's 2BB open.
- The other player can either call this all in, or fold.
- Because both players have to put in "blinds," there is a reward if the opponent folds.
    - Likewise, if we fold, we lose our blind.
- No post flop play. This massively simplifies the game, and the number of decisions we have to make.

Reward Matrix

Actions	Push	Fold
Win (Opponent Call)	+10	-1
Win (Opponent Fold)	+3	
Lose	-10	

Approach: CFR-Based Analysis

Counterfactual Regret Minimization

CFR Algorithm Key Components:

- Starts with a uniform strategy
- At each point in the game tree, the algorithm tracks the current regret — how much better we are performing by playing the current strategy rather than the probabilistic strategy
- Computes counterfactual heuristic as the product of reaching a game state (with the current player and opponent in mind) and the expected value of all of the possible outcomes of that state

Approach 1: CFR-Based Analysis

Counterfactual Regret Minimization

These three equations govern how the CFR algorithm improves the strategy over time.

CFR, The Math [1].

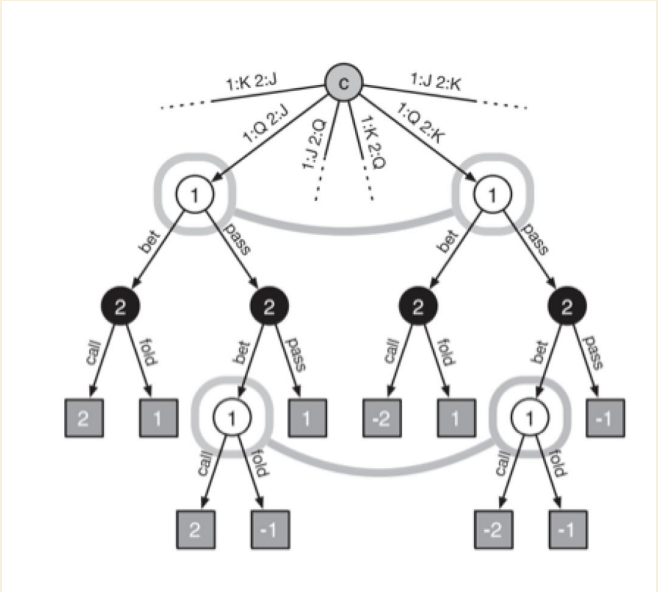
$$u_i(\sigma, I) = \frac{\sum_{h \in I, h' \in Z} \pi^{\sigma}_{-i}(h) \pi^{\sigma}(h, h') u_i(h')}{\pi^{\sigma}_{-i}(I)}$$

$$R_i^T(I, a) = \frac{1}{T} \sum_{t=1}^{T} \pi^{\sigma^t}_{-i}(I) \left(u_i(\sigma^t|_{I \to a}, I) - u_i(\sigma^t, I) \right)$$

$$\sigma_i^{T+1}(I)(a) = \begin{cases} \frac{R_i^{T,+}(I,a)}{\sum_{a \in A(I)} R_i^{T,+}(I,a)} & \text{if } \sum_{a \in A(I)} R_i^{T,+}(I,a) > 0 \\ \frac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

Why CFR?

- We have learned many algorithms for solving problems involving uncertainty in class — so why CFR?
- MCTS and Minimax work very poorly because of the uncertainty involved in the game. A worse hand preflop can still win a fair amount of the time at the end.
- Poker is typically not considered a game where a Markov model is applicable because it is very history dependent.

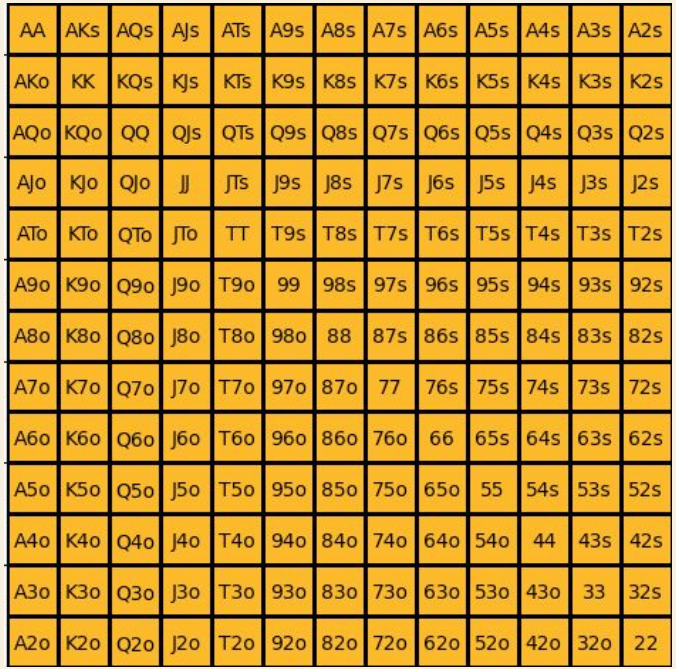


A possible poker game tree [4]

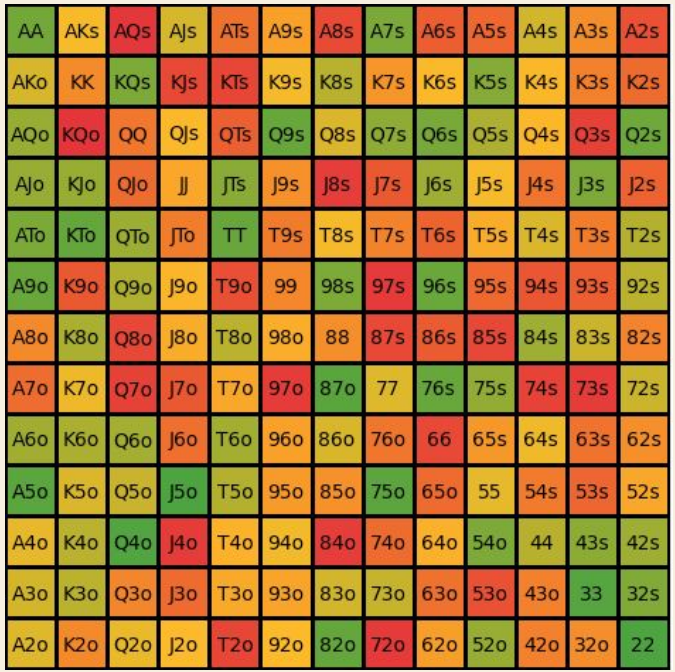
A Brief Discussion about Stack Depth

- The Small Blind (SB) and Big Blinds (BB) are the amount of money one has to put into the hand without seeing their cards
- We have said that the Small Blind is going to open to 3BB with any hand, and are asking the question of which hands are we shoving with.
- We expect that as the **stack sizes get larger**, we are shoving a **smaller** proportion of our hands because we have more money relative to the money in the pot.

Results: Baselines

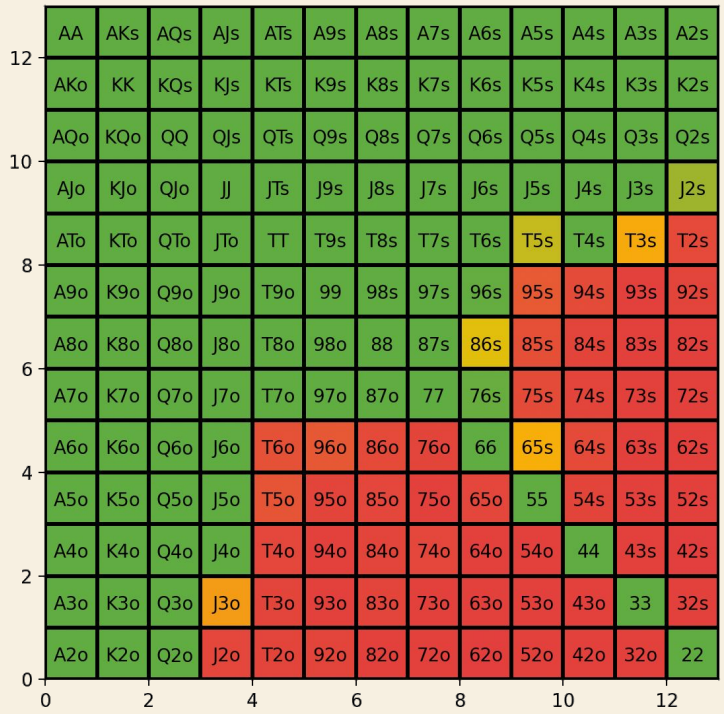


Uniform Strategy

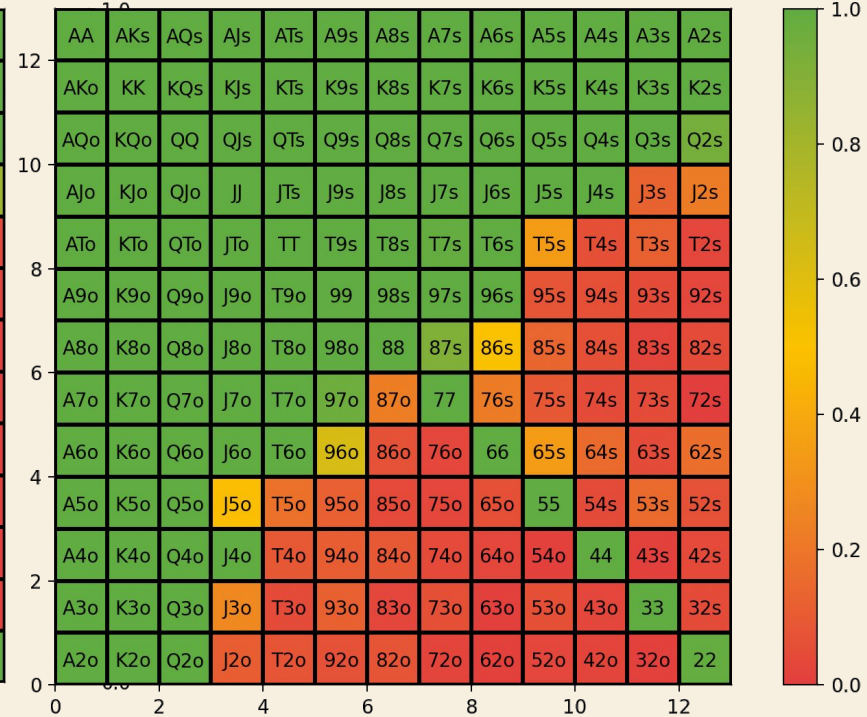


Random Strategy

Results: 10BB Strategy

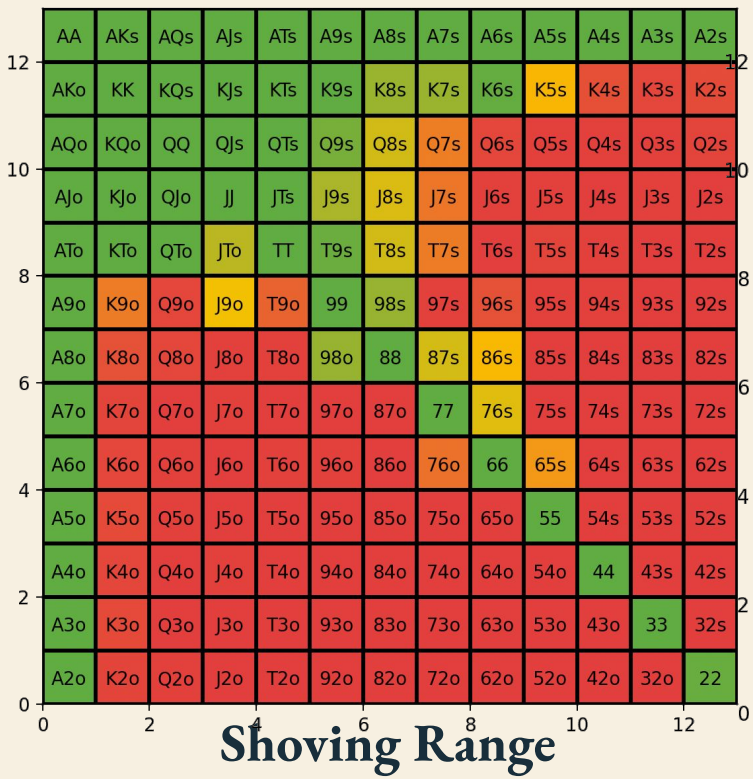


Shoving Range



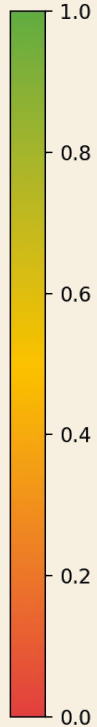
Defense Range

Results: 50BB Strategy

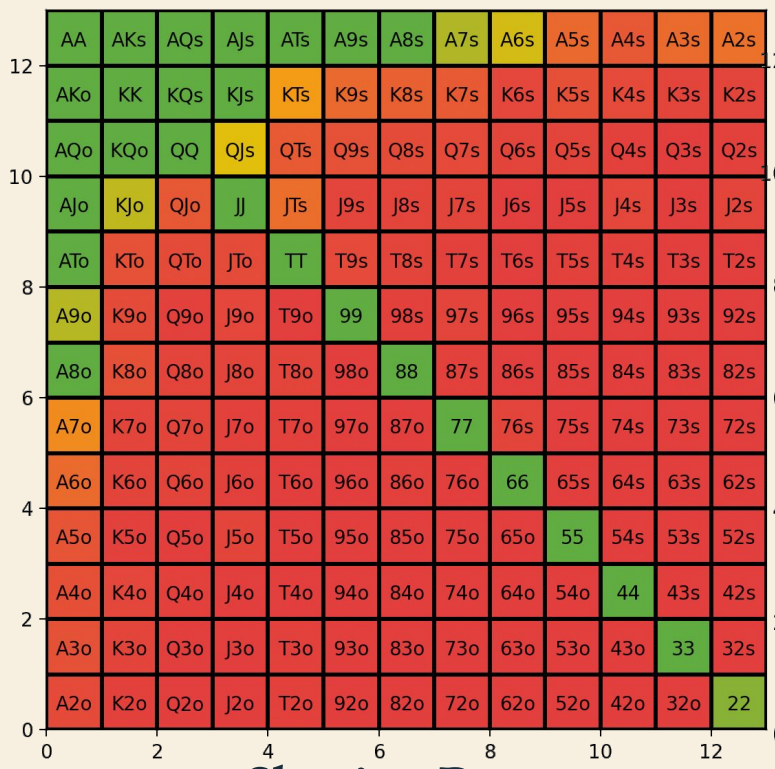


Shoving Range

Defense Range

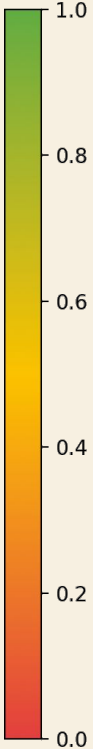


Results: 100BB Strategy

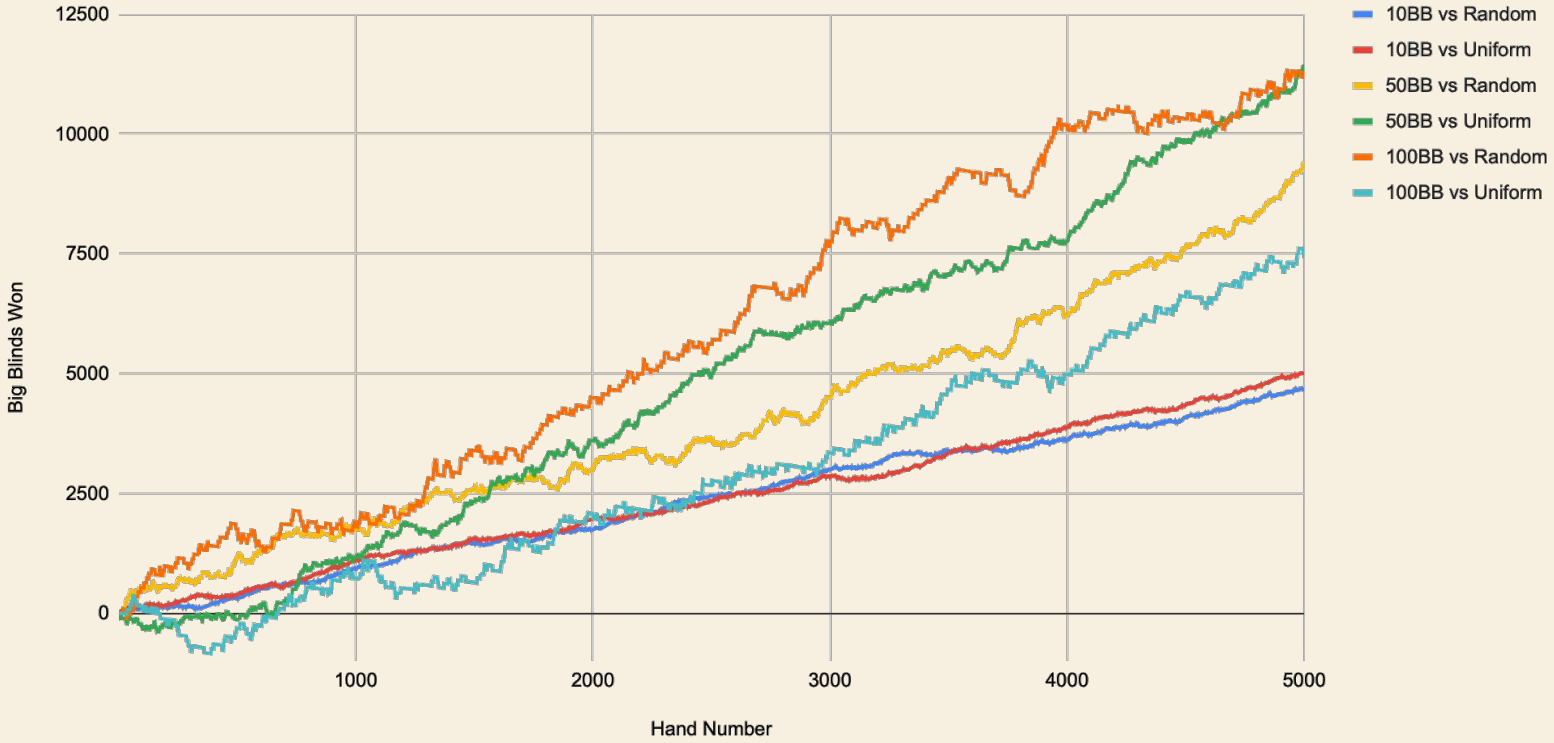


Shoving Range

Defense Range



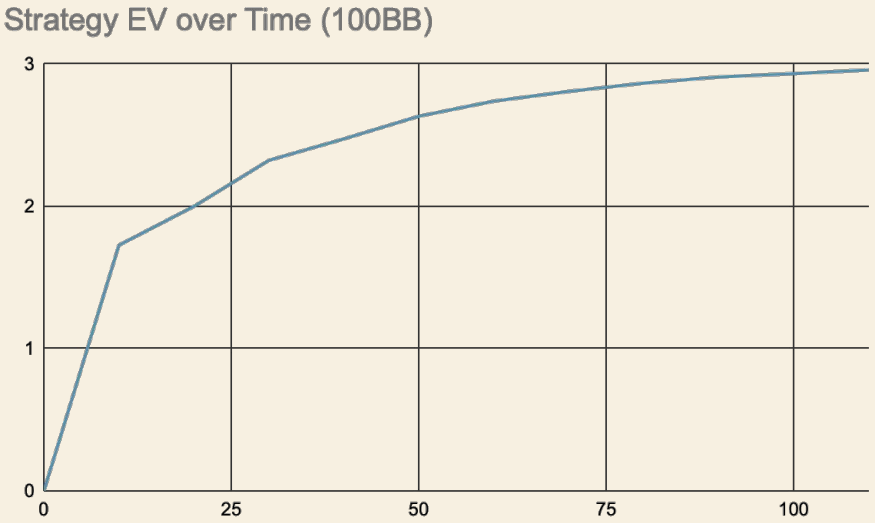
BB Won over 5000 Hands with Different Strategy Matchups



BB Won / 100 Hands

	Random Opponent	Uniform Opponent
10BB	93.3 bb/100	99.9 bb/100
50BB	188.1 bb/100	228.3 bb/100
100BB	225.6 bb/100	148.1 bb/100

Data from Training



Results Discussion

- Training poker solvers is **really** difficult. The game tree can become huge extremely quickly, and simulating poker hands can get **very** slow.
- Results ended up looking somewhat expected, particularly for the 50BB and 100BB strategy. The 10BB solver strategy is willing to shove/defend which a much larger proportion of hands than I expected.
- These solver strategies are **crushing** random and uniform strategies

References

[1] Regret Minimization in Games with Incomplete Information

[2] Libratus: The Superhuman AI for No-Limit Poker

[3] Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization

[4] AIPT Section 4.1: CFR – The CFR Algorithm