Multiplayer Guts Poker with Staggered Payouts

Hyunwoo Park[*]
Carnegie Mellon University
hp2@andrew.cmu.edu

Shiva Oswal[*]
Carnegie Mellon University
soswal@andrew.cmu.edu

Abstract

Multiplayer poker tournaments with staggered payouts present complex strategic challenges distinct from two-player zero-sum games. Traditional models like the Independent Chip Model offer heuristics for chip-to-monetary value conversion but fall short in capturing the nuanced incentives in multi-player settings. We introduce a simplified variant of Guts poker to analytically and computationally explore equilibrium strategies in tournaments with more than two players. For the two-player case, we derive a unique Nash equilibrium with closed-form threshold strategies, highlighting positional advantages. Extending to multiplayer scenarios, we develop a Monte Carlo-based fictitious play algorithm to approximate equilibria, uncovering phenomena such as middle stack pressure. Our findings advance the understanding of tournament poker dynamics.

1 Introduction

Poker tournaments are quintessential examples of multiplayer games with staggered payout structures, where players earn higher rewards by outlasting opponents. Unlike two-player zero-sum games, the strategic landscape in tournaments is shaped by factors such as chip accumulation, positional advantages, and risk management. A prominent heuristic in this domain is the Independent Chip Model(ICM), which approximates players' monetary equity based on their chip stacks. However, ICM does not fully encapsulate the strategic depth introduced by multiple competitors and tiered prize distributions [1].

Guts poker, a simplified variant involving single-round decision-making and uniformly distributed hand strengths, serves as an ideal framework to dissect these complexities. By reducing the game to binary fold or call decisions, we eliminate multi-street betting and community cards, focusing solely on the interplay of threshold strategies and positional advantages. This simplification allows for rigorous analytical exploration of Nash equilibria in scale.

In two-player Guts poker, we establish the existence of a unique Nash equilibrium characterized by threshold strategies, where each player's decision to call is based on surpassing a specific hand value cutoff. This equilibrium reveals a positional advantage for the second player, aligning with empirical observations in traditional poker variants. Extending our analysis to multiplayer tournaments, we develop a Monte Carlo-based fictitious play algorithm to approximate equilibrium strategies. This approach enables the identification of strategic phenomena such as middle stack pressure, where players with intermediate chip stacks adjust their aggression levels to optimize their tournament standing.

Our contributions are:

- We derive a closed-form Nash equilibrium for two-player Guts poker, providing insights into threshold-based decision-making and positional advantages.
- We develop a Monte Carlo-based fictitious play algorithm to approximate equilibria in multi-player Guts poker, addressing the computational challenges inherent in multi-agent continuous-action games.

[*]Equal contribution.

- We empirically demonstrate strategic patterns in multiplayer settings, such as middle stack pressure, thereby validating the relevance of our simplified model to real-world tournament dynamics.

2 Background

2.1 Independent Chip Model

The relationship between chip stacks and monetary payouts in poker tournaments becomes increasingly complex with more than two players. Unlike two-player tournaments, where chips and money are directly proportional, multi-player tournaments introduce non-linear payout structures that decouple chip equity from monetary equity. For instance, consider a scenario with three players holding chip stacks of 5000, 4900, and 100, and a blind structure of 50/100. If the player with 100 chips folds, and the player with 5000 chips goes all-in, the player with 4900 chips faces a strategic dilemma. In a cash game, this player would likely be indifferent to calling or folding if they anticipate a 50-50 outcome, as both actions would result in an expected monetary payoff of zero (neglecting blinds).

However, the dynamics change in a tournament setting with staggered payouts, such as 50% for first place, 30% for second, and 20% for third. If the player with 4900 chips folds, they are well-positioned to outlast the short-stacked player and have a 50% chance of finishing first or second, leading to an expected payout of \$40. Conversely, if they call, they risk elimination and are guaranteed third place if they lose, yielding an expected payout of \$35. Thus, the strategic value of folding increases, demonstrating how payout structures influence decision-making beyond chip counts.

To address this disparity between chip stacks and monetary equity, the Independent Chip Model has been widely adopted in tournament analysis. ICM estimates a player's expected monetary equity based on their current stack size relative to the total chips in play. This model assumes that finishing positions are determined solely by relative stack sizes, independent of future strategic adjustments or external factors.

ICM calculates the expected monetary value (U_i) for a player i as follows:

$$U_i = P_1 \cdot \frac{s_i}{S} + P_2 \cdot \sum_{j \neq i} \left(\frac{s_j}{S} \cdot \frac{s_i}{S - s_j} \right) + P_3 \cdot \sum_{j \neq i} \sum_{k \neq i, k \neq j} \left(\frac{s_j}{S} \cdot \frac{s_k}{S - s_j} \cdot \frac{s_i}{S - s_j - s_k} \right), \quad (1)$$

where:

- U_i represents the expected monetary equity of player i,
- s_i is the chip stack of player i,
- $S = \sum_j s_j$ is the total number of chips in play,
- P_1, P_2, P_3 are the payouts for first, second, and third places, respectively.

ICM simplifies the complex dynamics of tournament decision-making by providing a heuristic estimate of a player's monetary equity. It assumes that outcomes depend only on the probabilistic distribution of chip stacks across players, ignoring dynamic factors such as collusion, varying skill levels, or strategic adjustments.

This model has gained widespread acceptance in the poker community and serves as the foundation for many popular tournament tools, such as those referenced in sng [2]. While ICM is not without its limitations, it provides a practical framework for analyzing strategic decisions in tournaments.

2.2 Guts poker

Guts poker, in the simplified form discussed here, is starkly different from the complexity of no-limit Texas hold'em poker described earlier. Instead of dealing multiple cards and executing a sequence of betting rounds, guts poker involves a single, static assessment of hand strength and a straightforward decision process for each player.

While numerous variations of gut poker exist, we define the specific variant referenced in this work as follows. All players at the table begin by posting a fixed ante (for example, each player puts 1 chip into the pot). After all players have anted, each player is dealt a single hand value, drawn from a continuous uniform distribution on the interval $[0, 1]$. There are no community cards and no further cards are dealt at any point. The strength of each player's hand is determined immediately and does not change over the course of the hand.

Once all players have received their hand values, a single round of decision-making takes place in a predetermined, sequential order. Each player, when it is his turn to act, must choose one of two options: (1) fold: relinquish any claim to the pot and withdraw from the hand completely, investing no further chips; or (2) call: commit exactly one additional chip to the pot to remain in contention for the win. There is no concept of raising, and the notion of jamming or pushing all-in is not applicable here. Each player who calls effectively "qualifies" for a potential showdown, whereas those who fold forfeit their chance at winning.

After every player has either folded or called, any player who folded is removed from contention, and the remaining players reveal their hand values simultaneously. A "showdown" occurs among all callers: the player with the highest hand value among them wins the entire pot. In the event that two or more callers share an identical (and thus jointly highest) hand value — ties are resolved uniformly at random.

Guts poker provides an analytically tractable framework through its simple structure: players post antes, receive a single hand value from a uniform distribution, and make one binary decision to fold or call. Unlike traditional poker variants with multiple betting rounds and community cards, this simplified model isolates key strategic elements while maintaining mathematical rigor. The game's streamlined mechanics - no raising, no multi-street betting, and uniform hand distributions - allow us to analyze games with more players purely on threshold-based decision making and positional advantages along with stack sizes.

3 2-Player Guts Poker: ChipEV Analysis

Consider a simplified two-player version of the guts poker variant described previously. In this head-to-head setting, we focus on the strategic behavior of each player by examining their calling frequencies and how these frequencies influence chip expected value.

We model each player's decision by introducing a parameter that determines their calling threshold. Each player's hand, x_i, is drawn independently from a uniform $[0, 1]$ distribution. A player's "strategy" in this simplified scenario can be fully described by the probability with which they choose to call, which corresponds to a lower cutoff on their hand strength. By adjusting these cutoffs, players effectively tune their aggressiveness and risk tolerance in anticipation of their opponent's strategy.

3.1 Player 1's Strategy (α)

Player 1 sets a calling frequency $\alpha \in [0, 1]$. Since hands are uniform in $[0, 1]$, the most natural threshold strategy is to call on any hand that exceeds a certain cutoff and fold otherwise. If Player 1 wishes to call with probability α, then a simple and optimal cutoff policy is as follows:

$$\text{Player 1 calls if } x_1 \geq 1 - \alpha, \quad \text{and folds if } x_1 < 1 - \alpha.$$

This rule ensures that Player 1 calls with exactly a fraction α of all possible hands. High-value hands (close to 1) are always worth calling, while lower-value hands are folded to avoid risking additional chips with weak holdings.

3.2 Player 2's Strategy (β)

Player 2 adapts their calling strategy $\beta \in [0, 1]$ given the information about Player 1's action. Player 2's situation is contingent on what Player 1 does:

- **If Player 1 Folds:** When Player 1 chooses not to call, Player 2 faces no competition. Since Player 1 has effectively abandoned the hand, Player 2 will win the pot by calling regardless of the value of their hand. Thus, if Player 1 folds, it is trivially optimal for Player 2 to call with every possible hand. This corresponds to $\beta = 1$ in that scenario, since Player 2 automatically secures the pot uncontested and there is no downside.
- **If Player 1 Calls:** Now Player 2 must decide how to respond given that Player 1's hand strength is at least $1 - \alpha$. Player 2's best response is again defined by a threshold rule. If Player 2 wants

to call with frequency β, the cutoff is:

$$\text{Player 2 calls if } x_2 \geq 1-\beta, \quad \text{and folds if } x_2 < 1-\beta.$$

By implementing this threshold, Player 2 only calls with sufficiently strong hands, acknowledging that Player 1's call indicates a relatively high-value holding. Calling with weaker hands would result in a negative expectation, while folding such hands saves chips in the long run.

4 2-Player Guts Poker Strategy

We formalize the game as a zero-sum normal-form game and establish the existence of a unique Nash equilibrium. Moreover, we prove that the equilibrium strategies exhibit a *threshold structure*, a key property that simplifies both analysis and computation.

4.1 Game Definition and Structure

Consider an extensive-form game $\Gamma = (n, \mathcal{H}, \mathcal{I}, \mathcal{A}, u)$ where:

1. The set of players is $[2] = \{1, 2\}$.

2. The set of histories $\mathcal{H}$ consists of: - Root node $\varnothing$ - Signal nodes $(x_1, x_2) \in [0,1]^2$ where $x_i \sim \text{Uniform}[0,1]$ independently - Action nodes $\mathcal{H}_i$ for each player $i \in n$ - Terminal nodes $\mathcal{Z} \subset \mathcal{H}$

3. The information sets $\mathcal{I}_i$ partition $\mathcal{H}_i$ such that: - Player 1's information sets contain nodes with identical x_1 values - Player 2's information sets contain nodes with identical x_2 values and identical Player 1 actions

4. The action space $\mathcal{A}(\mathcal{I}) = \{\text{fold}, \text{call}\}$ for all $\mathcal{I} \in \mathcal{I}_i$, $i \in n$

5. The expected reward function $\mathbb{E}[u_i] : \text{co } \mathcal{X}_1 \times \text{co } \mathcal{X}_2 \to \mathbb{R}$ is:

$$\mathbb{E}[u_1(\sigma_1, \sigma_2)] = \mathbb{E}_{x_1, x_2}\left[\sum_{z \in \mathcal{Z}} \prod_{h \prec z} \sigma_{\iota(h)}(h) u_1(z)\right]$$

where $\iota(h)$ denotes the player who acts at history h, and:

$$u_1(z) = \begin{cases} -1 & \text{if player 1 folds} \\ 1 & \text{if player 1 calls, player 2 folds} \\ 2\mathbf{1}\{x_1 > x_2\} - 2\mathbf{1}\{x_1 < x_2\} & \text{if both call} \end{cases}$$

The game is zero-sum, so $\mathbb{E}[u_2(\sigma_1, \sigma_2)] = -\mathbb{E}[u_1(\sigma_1, \sigma_2)]$.

4.2 Threshold Structure of Equilibrium Strategies

We now show that any Nash equilibrium of this game must have *threshold strategies*. A threshold strategy is defined by a cutoff $\theta \in [0, 1]$ such that the player calls if and only if their private card $x \geq \theta$. Formally:

$$a_i(x_i) = \begin{cases} \text{call}, & x_i \geq \theta_i \\ \text{fold}, & x_i < \theta_i \end{cases}$$

To see why threshold strategies arise naturally, consider Player 1's optimization problem. Player 1 must balance the probability of winning uncontested (when Player 2 folds), the probability of winning at showdown (when both call), and the risk of losing chips when calling with a weak hand. Because the distribution is uniform and continuous, any non-threshold deviation would not yield a higher expected payoff. Intuitively, if Player 1 found it profitable to call on some interval of hands below a certain cutoff, continuity and the monotonic relationship between card values and winning probability would suggest that calling slightly better hands just above that interval would be at least as profitable. This argument extends to Player 2, who reacts to Player 1's strategy by adjusting their own threshold.

4.3 Threshold Structure of Equilibrium Strategies

Let $\Gamma = (n, \mathcal{H}, \mathcal{I}, \mathcal{A}, u)$ be our two-player extensive-form game. We establish that Nash equilibrium strategies must have a threshold structure.

Definition 4.1 (Threshold Strategy)**.** *A strategy $\sigma_i \in \Sigma_i$ is a threshold strategy if there exists $\theta_i \in [0, 1]$ such that:*

$$\sigma_i(x_i) = \begin{cases} \textit{call} & \textit{if } x_i \geq \theta_i, \\ \textit{fold} & \textit{if } x_i < \theta_i \end{cases}$$

Lemma 4.2 (Monotonicity)**.** *For any fixed strategy profile σ_{-i}, the expected utility $\mathbb{E}[u_i(x_i, \sigma_{-i})]$ is strictly increasing in x_i.*

Proof. For any $x_i < x_i'$ and fixed σ_{-i}:

$$\mathbb{P}(x_i > x_{-i}) < \mathbb{P}(x_i' > x_{-i}).$$

Therefore:

$$\mathbb{E}[u_i(x_i, \sigma_{-i})] < \mathbb{E}[u_i(x_i', \sigma_{-i})].$$

The inequality follows from the fact that higher values of x_i strictly increase the probability of winning at showdown while never reducing the probability of winning uncontested. □

Theorem 4.3 (Threshold Structure)**.** *In any Nash equilibrium of Γ, both players employ threshold strategies.*

Proof. We proceed by contradiction for each player:

Player 1's Strategy. Suppose Player 1's equilibrium strategy σ_1^* is not a threshold strategy. Then there exist $y < z$ in $[0, 1]$ such that:

$$\sigma_1^*(y) = \text{call} \quad \text{and} \quad \sigma_1^*(z) = \text{fold}.$$

By monotonicity (Lemma 1):

$$\mathbb{E}[u_1(z, \sigma_2^*)] > \mathbb{E}[u_1(y, \sigma_2^*)].$$

This contradicts σ_1^* being a best response, as Player 1 could improve their payoff by calling at z instead of y.

Player 2's Strategy. For any threshold strategy σ_1^* of Player 1 with threshold θ_1, Player 2's expected utility conditional on Player 1's call is:

$$\mathbb{E}[u_2(x_2, \sigma_1^* | \text{call})] = \begin{cases} -1 & \text{if fold,} \\ 2\mathbb{P}(x_2 > x_1 | x_1 \geq \theta_1) - 2\mathbb{P}(x_2 < x_1 | x_1 \geq \theta_1) & \text{if call.} \end{cases}$$

The same monotonicity argument applies: higher values of x_2 strictly increase the probability of winning at showdown, making the optimal strategy a threshold.

Therefore, any Nash equilibrium must consist of threshold strategies for both players. □

4.4 Unique Nash Equilibrium and Payoffs

With threshold strategies established, let Player 1's equilibrium threshold be $1 - \alpha$ and Player 2's equilibrium threshold be $1 - \beta$, where $\alpha, \beta \in [0, 1]$. Solving the resulting system of best-response equations (see, e.g., [1] for a detailed derivation), we compute the unique Nash equilibrium numerically:

$$\alpha = \frac{8}{9}, \quad \beta = \frac{2}{3}.$$

At this equilibrium, Player 1's expected value is:

$$\mathbb{E}[u_1] = -\frac{1}{9},$$

and Player 2's expected value is:

$$\mathbb{E}[u_2] = \frac{1}{9}.$$

These values confirm that Player 2 enjoys a *positional advantage*, a common trait in poker variants, reflecting the informational benefit of acting after Player 1. The uniqueness of the equilibrium and its zero-sum nature ensure that no other pair of strategies can yield better payoffs for either player.

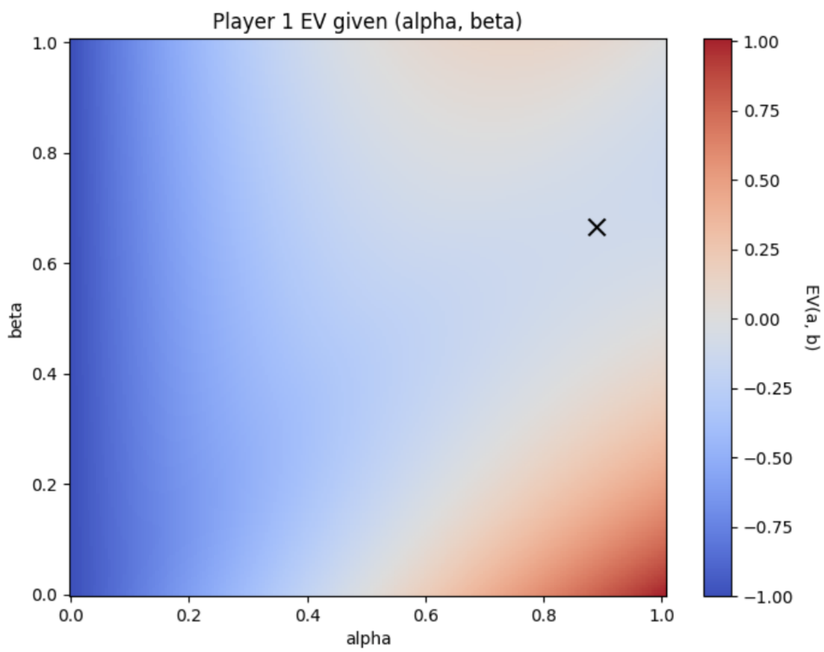

Figure 1: Heatmap of Player 1's expected value as a function of α and β. The x-axis represents α, the y-axis represents β, and the color indicates the expected utility $\mathbb{E}[u_1(\alpha, \beta)]$, ranging from -1.0 (blue) to 1.0 (red). The black 'X' denotes Nash Equilibrium.

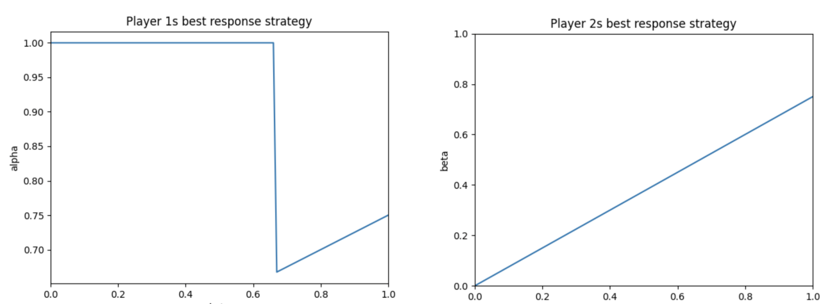

Figure 2: Best response strategies for both players. The left plot shows Player 1's best response strategy as a function of β, while the right plot shows Player 2's best response strategy as a function of α. The x-axes represent the opposing player's parameter, and the y-axes represent the optimal response for each player.

4.5 Equilibrium Derivation

Let Γ be our extensive-form game with threshold strategies parameterized by frequency $\alpha, \beta \in [0,1]$. We derive the unique Nash equilibrium.

Expected Value Decomposition. The expected value $\mathbb{E}[u_1(\alpha,\beta)]$ can be decomposed based on three disjoint events:

1. Player 1 folds with probability $(1-\alpha)$: - Player 2 automatically calls - Net payoff: -1 chip (lost ante)

2. Player 1 calls with probability α and Player 2 folds with probability $(1-\beta)$: - Player 1 invests 2 chips (ante + call) - Player 2 loses 1 chip (ante) - Net payoff: $+1$ chip for Player 1

3. Both players call with probability $\alpha\beta$: - Each player invests 2 chips (ante + call) - Winner takes entire 4-chip pot - Net payoff: $+2$ chips for winner, -2 chips for loser

Showdown Probability. Given both players call, we must compute $\mathbb{P}(x_1 > x_2)$:

$$\begin{aligned}
\mathbb{P}(x_1 > x_2) &= \int_{1-\beta}^{1} \int_{x_2}^{1} \frac{1}{\alpha\beta} dx_1 dx_2 \\
&= \int_{1-\beta}^{1} \frac{1-x_2}{\alpha\beta} dx_2 \\
&= \frac{1}{\alpha\beta} \int_{1-\beta}^{1} (1-x_2) dx_2 \\
&= \frac{1}{\alpha\beta} \left[x_2 - \frac{x_2^2}{2} \right]_{1-\beta}^{1} \\
&= \frac{1}{\alpha\beta} \left(1 - \frac{1}{2} - (1-\beta) + \frac{(1-\beta)^2}{2} \right) \\
&= \frac{1}{\alpha\beta} \left(\frac{\beta^2}{2} \right) \\
&= \frac{\beta}{2\alpha}
\end{aligned}$$

Expected Value During Showdown. When both call, Player 1's expected value is:

$$\begin{aligned}
\mathbb{E}[u_1|\text{call,call}] &= (+2)\mathbb{P}(x_1 > x_2) + (-2)(1 - \mathbb{P}(x_1 > x_2)) \\
&= 2\frac{\beta}{2\alpha} + (-2)(1 - \frac{\beta}{2\alpha}) \\
&= \frac{2\beta}{\alpha} - 2
\end{aligned}$$

Total Expected Value. Combining all cases:

$$\begin{aligned}
\mathbb{E}[u_1(\alpha,\beta)] &= (1-\alpha)(-1) + \alpha(1-\beta)(+1) + \alpha\beta(\frac{2\beta}{\alpha} - 2) \\
&= -1 + \alpha + \alpha - \alpha\beta + 2\beta^2 - 2\alpha\beta \\
&= -1 + 2\alpha - 3\alpha\beta + 2\beta^2
\end{aligned}$$

First-Order Conditions. For Nash equilibrium:

$$\begin{aligned}
&\frac{\partial}{\partial\alpha}\mathbb{E}[u_1] = 2 - 3\beta = 0 \\
&\frac{\partial}{\partial\beta}\mathbb{E}[u_2] = -\frac{\partial}{\partial\beta}\mathbb{E}[u_1] = -(4\beta - 3\alpha) = 0
\end{aligned}$$

Equilibrium Solution. From $2 - 3\beta = 0$:

$$\beta = \frac{2}{3}$$

From $4\beta - 3\alpha = 0$:

$$\beta = \frac{3\alpha}{4}$$

Substituting:

$$\frac{2}{3} = \frac{3\alpha}{4} \implies \alpha = \frac{8}{9}$$

Verification. At $(\alpha^*, \beta^*) = (\frac{8}{9}, \frac{2}{3})$:

$$\begin{aligned}
\mathbb{E}[u_1(\alpha^*, \beta^*)] &= -1 + 2\left(\frac{8}{9}\right) - 3\left(\frac{8}{9}\right)\left(\frac{2}{3}\right) + 2\left(\frac{2}{3}\right)^2 \\
&= -1 + \frac{16}{9} - \frac{16}{9} + \frac{8}{9} \\
&= -\frac{1}{9}
\end{aligned}$$

Therefore $\mathbb{E}[u_2(\alpha^*, \beta^*)] = \frac{1}{9}$, confirming Player 2's positional advantage.

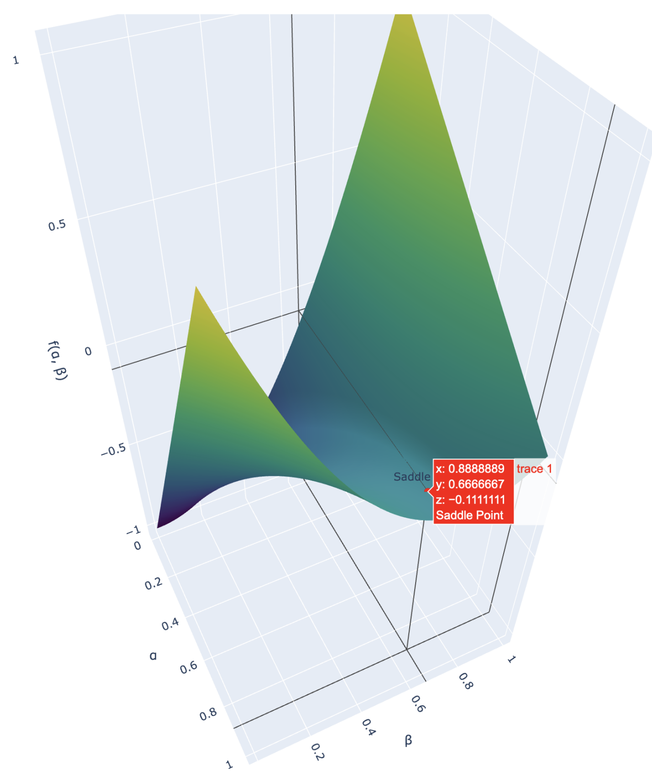

Figure 3: Saddle point of the function $\mathbb{E}[u_1(\alpha, \beta)] = -1 + 2\alpha - 3\alpha\beta + 2\beta^2$. The saddle point is marked at $(\alpha^*, \beta^*, f(\alpha^*, \beta^*)) = (0.888, 0.667, -0.111)$.

5 Multiplayer Guts: ChipEV Analysis

5.1 Game Definition and Structure

Consider an extensive-form game $\Gamma = (n, \mathcal{H}, \mathcal{I}, \mathcal{A}, u)$ with $n > 2$ players, indexed by $[n] := \{1, 2, \ldots, n\}$:

1. **Players:** The set of players is $[n]$.

2. **Histories:** The set of histories $\mathcal{H}$ includes:

- A root node representing the start of the game.
- Chance nodes corresponding to the independent draws of private hand values $h_i \sim \text{Unif}(0, 1)$ for each player $i \in [n]$.
- Action nodes where each player $i \in [n]$ decides, either simultaneously or in sequence, to take one of two actions: fold or call.
- Terminal nodes corresponding to outcomes after all players have either folded or called.

3. **Information Sets:** Each player's information sets $\mathcal{I}_i$ partition their action nodes such that each information set groups nodes with identical private hand values h_i. Since each player receives a unique private signal h_i, the information sets reflect the continuous nature of the strategy space. In other words, a player's strategy is a continuous mapping from $[0, 1]$ to a binary action. A common parametrization of strategies is via a threshold $\alpha_i \in [0, 1]$.

4. **Actions:** For each information set $\mathcal{I} \in \mathcal{I}_i$, the action space is $\mathcal{A}(\mathcal{I}) = \{\text{fold}, \text{call}\}$. Calling costs 1 chip.

5. **Payoffs:** Let N_c be the set of players who call. If $|N_c| = 1$, that player wins the entire pot (including all antes and calls). If $|N_c| > 1$, the player among N_c with the highest hand value wins the pot. The payoff to a player who folds is 0. A caller who loses pays 1 chip. A caller who wins against k other callers receives the pot of size $n + k$ (the original antes plus the k additional calls), netting a positive profit. Thus, each player's expected utility u_i depends on their threshold α_i and the thresholds α_j of all other players $j \neq i$.

5.2 Complexities and Challenges

While the two-player version of Guts poker admits a unique Nash equilibrium with threshold strategies, the multiplayer setting ($n > 2$) introduces complexity:

- Nash equilibrium is not well-defined. The game is no longer strictly zero-sum, and the notion of a global equilibrium is not straightforward[3].
- Standard no-regret algorithms and iterative procedures (e.g., Fictitious Play, Online Mirror Descent) do not guarantee convergence in multiplayer continuous-action games[4, 5].
- Although no closed-form Nash equilibrium solution is known for $n > 2$, prior work suggests that approximate equilibria can often be found numerically[6].

5.3 Continuous State-Space and Best-Response Computation

The game's state space in an n-player setting can be described by at most $2n - 1$ distinct states where each player is either calling or folding in sequence. At each such decision point, one player chooses between calling or folding based on their threshold strategy. Computing a best response reduces to solving a polynomial equation derived from the player's ChipEV as a function of all players' thresholds.

Example: Last-to-Act Player. Consider a player who is last to act facing k previously committed callers with calling thresholds $\{c_1, c_2, \ldots, c_k\}$. Let the last player's threshold be α. The opponents' hands are $h_i \sim \text{Unif}(0, c_i)$ and the last player's hand is $h \sim \text{Unif}(0, \alpha)$. The probability of the last player winning is the probability that $h < \min\{h_1, \ldots, h_k\}$. Calling costs 1 chip. If the player wins, the payoff is the pot size $n + k$ minus the 1 chip cost; losing results in a net -1. Folding yields 0. This structure allows best-response conditions to be computed by integrating over the continuous distribution of hand values and solving for the optimal threshold α.

5.4 Numerical Techniques and Policy Iteration

Because the problem reduces to finding roots of a polynomial, we can apply standard numerical methods (e.g., `scipy.optimize.root_scalar` in Python) to obtain the best response for each player given the thresholds of the others. The initial guess for $\pi_i^{(0)}$ can be the policy computed in the previous iteration, ensuring continuity and faster convergence. Removing integral computations in favor of direct polynomial evaluation often yields substantial computational speedups (we observed over 200x improvement in practice).

To approximate stable strategies in the multiplayer setting, one can employ a form of *policy iteration*:

1. Initialize each player's calling threshold $\pi_i^{(0)}$ randomly.
2. At iteration i, for each state s, identify the terminal states reachable under the current profile $\{\pi_j^{(i-1)}\}$.
3. For each terminal state t, compute the probability p_t of reaching t and derive the EV polynomial for the acting player at s.
4. Solve the polynomial equation to find the best response π_s^{BR} for the acting player at state s.
5. Update:

$$\pi_i^t = \left(1 - \frac{1}{t}\right) \pi_i^{t-1} + \frac{1}{t} \pi_i^{BR}$$

This *smoothed fictitious play* approach dampens oscillations and encourages convergence.

6 Some Multiplayer Guts ChipEV Results

In this section, we present illustrative results from applying the methods described above to multiplayer Guts poker scenarios. By incorporating continuous threshold strategies and polynomial root-solving techniques, we are able to compute approximate equilibria for configurations with up to 8 players. This substantially extends prior work (e.g., [7], [8]) that typically considered only a few players or relied on discretized action spaces.

6.1 Representative Example

Figure 4 shows a small extract of the game tree for a three-player Guts scenario, with each node annotated by a player index and the calling threshold c learned through our iterative procedure. The depicted thresholds represent stable policies after a number of iterations, indicating that Player 1's optimal calling threshold converges to approximately 0.66, Player 2's to approximately 0.92 on one branch and 0.44 on another, and Player 3's thresholds range from 0.31 to 1 depending on the path. Despite the complexity of multi-branching outcomes, these threshold values stabilize and serve as reasonable approximations to equilibrium strategies.

6.2 Sample Expected Values and Performance

The equilibrium-like thresholds yield the following expected values for the three players:

$$\mathbb{E}[u_1] \approx -0.140, \quad \mathbb{E}[u_2] \approx 0.015, \quad \mathbb{E}[u_3] \approx 0.125.$$

These results indicate distributional differences in chip EV, likely attributable to positional advantages and the non-symmetric equilibrium structure that emerges in multiplayer settings.

6.3 Computational Complexity and Runtime

While computing best responses in a two-player scenario is relatively straightforward, the complexity grows significantly with the number of players. The number of states increases combinatorially as each player can call or fold, resulting in approximately $2n - 1$ unique states and branching factors. The computational complexity scales as:

$$O(3^n \cdot n \cdot T),$$

where n is the number of players and T is the number of iterations required for convergence.

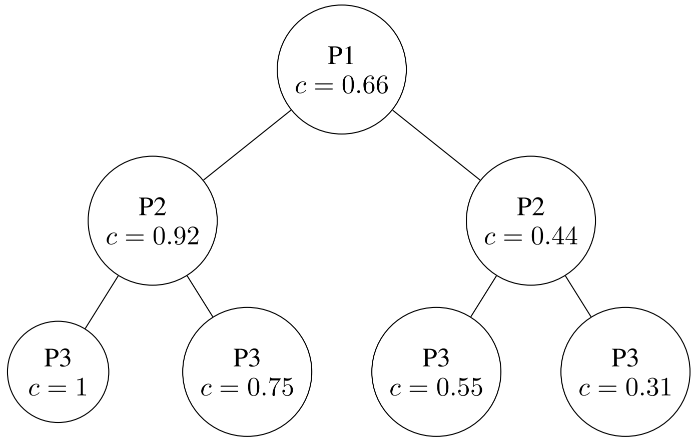

Figure 4: A partial game tree showing learned calling thresholds for a 3-player Guts configuration. Each node corresponds to a state where a single player chooses a calling threshold, potentially leading to multiple downstream branches.

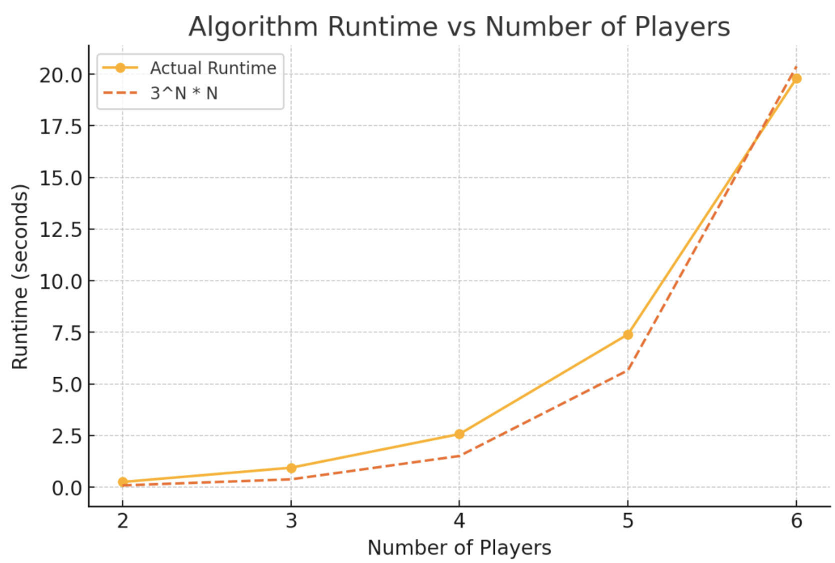

Figure 5: Algorithm runtime as a function of the number of players. The solid orange line represents the actual runtime in seconds, while the dashed red line shows the theoretical complexity $3^N \cdot N$, where N is the number of players.

Table 1 reports sample runtimes for 10 iterations of our policy iteration procedure on different game sizes. Although the complexity grows quickly, we have been able to compute equilibria for games up to $N = 8$ players by leveraging efficient polynomial solving, memoization, and careful pruning of the game tree.

6.4 Discussion and Extensions

Our results demonstrate that, despite the theoretical challenges associated with defining and computing a Nash equilibrium in multiplayer continuous-action settings, practical approaches yield stable and meaningful approximations. By focusing on continuous threshold strategies rather than discretizing the action space, we avoid large-scale combinatorial explosions in state-action representation. This approach leads to more tractable computations and better scaling to larger n. We extend the prior literature by pushing the boundary from previously studied small games (e.g., $n \leq 3$) to larger ones (e.g., $n \leq 8$), allowing to study strategic behavior in more realistic tournament-like scenarios.

Number of Players	Algorithm Runtime (10 Iterations)
2	0.25s
3	0.94s
4	2.57s
5	7.40s
6	19.80s

Table 1: Empirical runtime scaling for iterative approximate equilibrium computation in multiplayer Guts.

While not providing closed-form Nash equilibria or formal convergence guarantees in every setting, these techniques are suitable to explore large, continuous, and convex normal-form games such as multiplayer Guts poker.

7 Tournament Poker as a Stochastic Game

Beyond single-hand models of poker, tournament play introduces an inherently dynamic, multi-stage decision process. As players win or lose chips over multiple hands, their changing stack sizes influence strategic considerations. Modeling tournament poker as a stochastic game enables us to capture these evolving conditions and analyze long-run equilibria or optimal strategies using dynamic programming techniques.

7.1 State Representation

A state in the tournament model encodes all relevant information needed to determine future payoffs and feasible actions. We consider the following factors as part of the state description:

1. **Stack Sizes:** Each player $i \in [n]$ has a current stack size $s_i \geq 0$. The total number of chips in play is constant, denoted by C. Over the course of the tournament, players gain and lose chips, but the sum $\sum_{i=1}^{n} s_i = C$ remains invariant.
2. **Information Sets:** The current hand's partial action sequence—such as which players have folded, called, or raised—constitutes the *information set* for that decision point. While no community cards or evolving betting rounds exist in simplified Guts-style models, the actions taken so far still matter, as they determine which players remain in contention and who will act next.
3. **Button Position:** In poker, a crucial element is the position of the "button," which designates the nominal dealer and establishes the order of play.Without loss of generality, we can fix the button at position $n-1$ by permuting players.

Two states are considered *permutation equivalent* if they differ only by a permutation of player indices that preserves the relative configuration of stack sizes and button position. For instance, the states $(\text{stack} = [5, 8, 3], \text{button} = 2)$ and $(\text{stack} = [8, 3, 5], \text{button} = 0)$ can be mapped to one another by permuting players appropriately.

7.2 Game Definition and Structure

Consider an extensive-form stochastic game

$$\Gamma = \left(n, \mathcal{H}, \{\mathcal{I}_i\}_{i=1}^{n}, \mathcal{A}, u, P\right)$$

with the following components:

1. **Players:** The set of players is $[n] = \{1, 2, \ldots, n\}$, representing all participants in the tournament.
2. **States:** A state $s \in S$ encodes the current distribution of chips among players, the current button position, and any relevant information sets. Specifically, a state can be represented as:

   $$s = (\mathbf{s}, b, \mathcal{I})$$

   where:

- $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ denotes the stack sizes of each player, with $\sum_{i=1}^{n} s_i = C$, the total number of chips.
- $b \in [n]$ indicates the current button position.
- $\mathcal{I}$ represents the current information set, encapsulating the history of actions taken in the current hand.

3. **Actions:** At each state, players with non-zero stacks can choose from a set of actions, typically including *fold* or *call*. The action set for player i at state s is:

$$\mathcal{A}_i(s) = \begin{cases} \{\text{fold}, \text{call}\}, & \text{if } s_i > 0, \\ \{\emptyset\}, & \text{if } s_i = 0. \end{cases}$$

4. **Transition Function:** The transition probabilities $P(s'|s, a)$ determine how the game moves from state s to state s' given the action profile $a = (a_1, a_2, \ldots, a_n)$. Transitions account for the outcomes of hands, such as chip transfers based on actions and resolved payoffs.
5. **Utility Functions:** Each player i has a utility function $u_i : S \times \mathcal{A} \to \mathbb{R}$ that assigns payoffs based on the outcomes of actions and terminal states. Utilities reflect the players' chip stacks and eventual tournament rankings.

7.3 State Transitions and Dynamic Programming

The evolution of the tournament is governed by the state transition probabilities $P(s'|s, a)$, which capture the stochastic nature of hand outcomes and player actions. To analyze the game, we define a continuation value function $V_i : S \to \mathbb{R}$ for each player i, representing the expected utility from state s onward under a given strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$.

The continuation value satisfies the Bellman equation:

$$V_i(s) = \mathbb{E}_\sigma \left[u_i(s, a) + \sum_{s' \in S} P(s'|s, a) V_i(s') \right],$$

where the expectation is taken over the strategies σ and any inherent randomness in the game (e.g., card dealing).

Dynamic programming techniques, such as backward induction, can be employed to solve for the equilibrium strategies by recursively computing $V_i(s)$ for all states s starting from terminal states and moving backward.

7.4 Threshold Structure in a Tournament Context

In a tournament or multi-hand scenario, threshold strategies extend the single-hand analysis by incorporating the dynamic state of the game. Each player i may adopt a *state-dependent threshold* $\theta_i(s)$ that determines their decision to call or fold based on the current state s. Specifically, a threshold policy for player i is defined as:

$$\sigma_i(x_i, s) = \begin{cases} \text{call}, & \text{if } x_i \geq \theta_i(s), \\ \text{fold}, & \text{if } x_i < \theta_i(s). \end{cases}$$

Here, x_i represents player i's private signal (e.g., hand strength), and s encapsulates the current state of the game, including stack sizes and the action history.

A *Nash equilibrium* in this extensive-form game is a strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$ such that no player can improve their expected continuation value $V_i(s; \sigma_i^*, \sigma_{-i}^*)$ by unilaterally deviating from their equilibrium threshold strategy σ_i^*. Formally, for all players i and for all possible deviations σ_i,

$$V_i(s; \sigma_i^*, \sigma_{-i}^*) \geq V_i(s; \sigma_i, \sigma_{-i}^*).$$

Identifying or proving the existence of such equilibria in a general n-player tournament setting is significantly more complex than in the two-player zero-sum case. The challenges arise from the increased dimensionality of the state space, the interdependencies of players' strategies, and the potential for multiple equilibria.

7.5 Terminal and Non-Terminal States

A tournament ends when fewer than n players remain with non-zero chips, i.e., some players have been eliminated. At such a point, the prize distribution is determined by the finishing positions. Thus, any state in which the count of players with non-zero stacks is less than n is *terminal*. The payoffs at these terminal states can be pre-computed using a backward induction or a dynamic programming (DP) approach, as the continuation value from that point forward is fully determined.

7.6 Counting the State Space

Given n players and C total chips, consider how to count the number of non-terminal states. Since each state is defined by a vector of stack sizes that sum to C, the number of distinct stack configurations (up to permutation) can be computed combinatorially. The count of all non-negative integer solutions to:

$$s_1 + s_2 + \cdots + s_n = C$$

with $s_i \geq 1$ (ensuring all players are still in contention) is:

$$\binom{C-1}{n-1}.$$

Moreover, consider the possible action histories or information sets. Even in a simplified Guts setting, each player in a hand can either fold or call at each decision node, leading to a combinatorial explosion of possible histories. In a single betting round guts variant, the number of such action configurations (excluding the trivial case where no one acts) scales as $(2^n - 1)$, corresponding to all non-empty subsets of players calling. Combining the stack configurations and the action sequences, the total number of non-terminal states can be approximated as:

$$\binom{C-1}{n-1} \cdot (2^n - 1).$$

7.7 Implications for Computation

The exponential growth in the number of states—both in terms of chip distributions and action histories—poses significant computational challenges. While for small n and manageable C, one can attempt a direct MDP[9] or Stochastic Game solution[10], scaling to large tournaments remains computationally intensive.

Despite these challenges:

- **Dynamic Programming:** Terminal payoffs can be solved exactly using DP, as the game reduces to subproblems with fewer active players.
- **Approximations and Heuristics:** For large n and C, approximation techniques, sampling, or heuristic-based policies may be necessary.
- **Leveraging Symmetry and Structure:** Recognizing permutation equivalences and using canonical forms of states can reduce the effective state space, improving tractability.

Modeling tournament poker as a stochastic game lays the groundwork for analyzing more complex strategic interactions, from multi-street variants to multi-table formats. While exact equilibrium solutions remain elusive for large-scale instances, this framework enables structured exploration of policies, approximate equilibrium solutions, and strategic principles in increasingly realistic models of tournament play.

8 Challenges with Fictitious Play and Mixed-Strategy Equilibria

The complexity of computing equilibria in continuous-action poker variants poses significant challenges. Even in the relatively simple two-player model, the process of identifying a Nash equilibrium can yield unintuitive or "messy" solutions when one relies on discrete approximation rather than closed-form analysis. For instance, consider the discrete approximation of the two-player Guts Nash equilibrium strategy vector:

$$[0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.6667,\ 0.3333]$$
$$[0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.00,\ 0.6667,\ 0.3333,\ 0.00,\ 0.00].$$

Interpreting this solution is troublesome. It essentially says that, for Player 1, two-thirds of the time they set their calling threshold to 0.875, and one-third of the time to 1.0—an odd mixture that does not correspond to a clean, pure threshold. By contrast, the analytical solution we derived previously gave a pure threshold at $\alpha = \frac{8}{9} \approx 0.8889$. Such pure solutions are preferable as they are more interpretable and stable.

8.1 Mixed Strategies in Larger Games

For more complex multiplayer tournament Guts scenarios, no known closed-form equilibrium exists. Obtaining equilibrium strategies often necessitates iterative methods like Fictitious Play (FP). However, these can yield mixed threshold strategies that are difficult to interpret and may not align with practical poker strategies. While mixed strategies are theoretically valid, in a game like Guts—where the action space is structurally continuous but economically simple—having a "pure" threshold (a single cutoff point) is far more desirable.

8.2 Issues with Standard Fictitious Play

Standard FP assumes each player best responds to the empirical distribution of opponents' strategies. Yet in continuous and complex payoff landscapes, best-response computations can be highly non-trivial. As a result, we frequently observe "nonsensical" updates—e.g., the best response might jump unpredictably due to slight changes in the strategy profile, leading to erratic mixed strategies. Such instability is problematic both for convergence guarantees and for the interpretability of results.

8.3 Monte Carlo Fictitious Play

To address these difficulties, we implemented a Monte Carlo-based variant of FP, inspired by the Monte Carlo Fictitious Play approach (see Kiatsupaibul et al. [11] for a detailed analysis). The idea is intuitive:

1. Begin at the root state.
2. Sample a set of random hands and simulate plays down to terminal states.
3. Given these sampled outcomes, compute a best response (BR) for the player in question—either by direct polynomial root-solving or by numerical approximation.
4. Update the player's policy using a smoothed FP update:
$$\pi_i^t = \left(1 - \frac{1}{t}\right)\pi_i^{t-1} + \frac{1}{t}\pi_i^{BR}$$

This Monte Carlo approach reduces reliance on fully enumerating the state space and computing exact best responses at every iteration, thereby improving stability and yielding more sensible policies.

The logic is analogous to Monte Carlo Counterfactual Regret Minimization (MC-CFR)[12] techniques but adapted to continuous threshold strategies. By operating on samples rather than full expansions, we reduce computational overhead and achieve more stable updates.

8.4 Empirical Observations

Two-Player Tournament ($n = 2$, $C = 20$, payouts $p = [100, 0]$): In a simplified two-player tournament setting, we observed that the calling threshold tends to decrease as a player's stack size diminishes—a result that aligns well with standard poker intuition. Specifically:

- At $s_i = 1$ chip, the calling frequency is effectively **100%**.
- At $s_i = 2$ chips, it drops to **96%**.
- At $s_i = 10$ chips, it hovers around **90%**, compared to the pure chip-EV threshold of **88.88%**.

This near-linear relationship between expected value and stack size corroborates the well-established Independent Chip Model (ICM), reinforcing its relevance in analyzing tournament equity distribution, even in reduced settings.

Three-Player Tournament ($n = 3$, $C = 9$, payouts $p = [30, 10, 0]$): In a scenario with three players, we observed "middle stack pressure," a phenomenon well-documented in actual poker tournaments. Consider the stack configuration $\mathbf{s} = (s_1, s_2, s_3) = (2, 6, 1)$:

- Player 1, with $s_1 = 2$ chips, calls only **78%** of the time.
- Contrast this with a $(s_1, s_2) = (2, 6)$ two-player scenario where the same player would call **96%**.

The presence of a third player with just $s_3 = 1$ chip—who is essentially forced to go all-in in the next hand—creates strategic tension. Player 1 must fold some weaker hands to avoid finishing third, reflecting a subtle interplay of risk aversion and positional advantage. In fact, changing the order of stacks (e.g., to $\mathbf{s} = (1, 6, 2)$) can increase Player 1's EV from **10.1** to **11.8**, demonstrating how stack order influences incentives and calling thresholds.

8.5 Conclusion

Our findings suggest that while standard Fictitious Play can struggle to produce stable and interpretable mixed-strategy equilibria in continuous-action games like tournament Guts, a Monte Carlo-based approach can yield more stable and intuitive outcomes. The observed behaviors—stack-size-dependent aggressiveness in two-player settings and "middle stack pressure" in three-player scenarios—align with widely recognized poker concepts.

Acknowledgements

This work was done as a final project for 15-888 (Fall'24): Computational Game Solving. We thank out instructors Brian Zhang and Tuomas Sandholm for helpful feedbacks and insightful discussions.

References

[1] Trojanrabbit. New algorithm to calculate icm for large tournaments. https://forumserver.twoplustwo.com/15/poker-theory-amp-gto/new-algorithm-calculate-icm-large-tournaments-1098489/, 2011.

[2] Sit-n-go power tools. http://www.sitngo-analyzer.com/, 2005.

[3] Sam Ganzfried, Andrew Nowak, and Juan Pinales. Successful nash equilibrium agent for a 3-player imperfect-information game. In *Proceedings of the AAAI Workshop on Computer Poker and Imperfect Information Games*. AAAI Press, 2018.

[4] Yurong Chen, Xiaotie Deng, Chenchen Li, David Mguni, Jun Wang, Xiang Yan, and Yaodong Yang. On the convergence of fictitious play: A decomposition approach. *arXiv preprint arXiv:2205.01469*, 2022.

[5] Zhengyuan Zhou, Panayotis Mertikopoulos, Aris L. Moustakas, Nicholas Bambos, and Peter Glynn. Mirror descent learning in continuous games. *arXiv preprint arXiv:1711.00143*, 2017.

[6] Kevin Buck, Jae Hwan Lee, Jacob Platnick, Aric Wheeler, and Kevin Zumbrun. Continuous guts poker and numerical optimization of generalized recursive games. *arXiv preprint arXiv:2208.02788*, 2022.

[7] Sam Ganzfried and Tuomas Sandholm. Computing an approximate jam/fold equilibrium for 3-player no-limit texas hold'em tournaments. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, AAMAS '08, pages 919–925, 2008.

[8] Tuomas Sandholm. The state of solving large incomplete-information games, and application to poker. *AI Magazine*, 31(4):13–32, 2010. URL https://www.cs.cmu.edu/~sandholm/steeringEvolution.csd50.pdf.

[9] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[10] Daniel S. Bernstein and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2004. URL https://aaai.org/papers/00709-aaai04-112-dynamic-programming-for-partially-observable-stochastic-games/.

[11] Seksan Kiatsupaibul, Giulia Pedrielli, Christopher Thomas Ryan, Robert L. Smith, and Zelda B. Zabinsky. Monte carlo fictitious play for finding pure nash equilibria in identical interest games. *INFORMS Journal on Optimization*, 6(3-4):155–172, 2024. doi: 10.1287/ijoo.2024.0063.

[12] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems*, volume 22, 2009. URL https://proceedings.neurips.cc/paper/2009/file/00411460f7c92d2124a67ea0f4cb5f85-Paper.pdf.

A Appendix

A.1 Notation

1. **Probability Simplex.** $\Delta(S)$ is the probability simplex on set S, defined as:

$$\Delta(S) := \left\{ \mathbf{x} \in \mathbb{R}^S_{\geq 0} : \sum_{s \in S} x(s) = 1 \right\}.$$

For $\mathbf{x} \in \Delta(S)$, $\operatorname{supp} \mathbf{x}$ denotes the support of $\mathbf{x}$.

2. **Big-O Notation.** The notations $f \lesssim g$, $f \gtrsim g$, and $f \sim g$ mean $f = O(g)$, $f = \Omega(g)$, and $f = \Theta(g)$, respectively. Additionally, $\widetilde{O}$, $\widetilde{\Omega}$, and $\widetilde{\Theta}$ hide logarithmic factors.

3. **Histories and Nodes.** $\mathcal{H}$ is the set of all nodes (or histories) in an extensive-form game tree. Each history h represents a unique path from the root $\varnothing$ to h. Terminal nodes are denoted $\mathcal{Z} \subseteq \mathcal{H}$.

4. **Strategies.**
   (a) *Pure Strategies:* A pure strategy $\mathbf{x}_i$ for player i is a binary vector that selects exactly one action at every decision point.
   (b) *Mixed Strategies:* A mixed strategy π_i is a distribution over pure strategies, where the realization form is $\mathbb{E}_{\mathbf{x} \sim \pi_i} \mathbf{x}_i$.
   (c) *Behavioral Strategies:* Mixed strategies where actions are independent across decision points.

5. **Expected Value under Profile.** The expected value of player i under a strategy profile π is given by:

$$u_i(\pi) := \mathbb{E}_{z \sim \pi} \, u_i(z),$$

where z is a terminal node sampled according to the probability distribution induced by the strategy profile π. For uncorrelated profiles $\pi = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the expected value can be expressed as:

$$u_i(\mathbf{x}) = \sum_{z \in \mathcal{Z}} x_C(z) u_i(z) \prod_{i=1}^{n} \mathbf{x}_i[z],$$

where $x_C(z)$ is the probability that chance plays all actions leading to z, and $\mathbf{x}_i[z]$ is the realization probability of player i taking actions leading to z.

6. **Equilibria.**
   (a) *Nash Equilibrium:* An ϵ-Nash equilibrium is a strategy profile $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ where no player can improve by more than ϵ using a unilateral deviation:

$$u_i(\mathbf{x}'_i, \mathbf{x}_{-i}) \leq u_i(\mathbf{x}_i, \mathbf{x}_{-i}) + \epsilon.$$

   (b) *Correlated Equilibria:* Defined by a tuple of transformation sets $\Phi = (\Phi_1, \ldots, \Phi_n)$, where $\Phi_i \subseteq (\text{co } \mathcal{X}_i)^{\mathcal{X}_i}$. An ϵ-Φ-equilibrium satisfies:

$$\mathbb{E}_\pi \left[u_i(\phi_i(\mathbf{x}_i), \mathbf{x}_{-i}) - u_i(\mathbf{x}_i, \mathbf{x}_{-i}) \right] \leq \epsilon.$$

7. **Tree-Form Decision Making.** A tree-form decision problem consists of nodes representing *decision points* $j \in \mathcal{J}$ (where actions $a \in A(j)$ are selected) and *observation points* Σ (where players make observations). The decision tree alternates between these nodes, with $\varnothing$ as the root.