Evolutionary Optimization of Player Traits in Poker Using Genetic Algorithms

Isha Bhandari
EECS
University of Tennessee, Knoxville
Knoxville, TN
ibhandar@vols.utk.edu

Jason Choi
EECS
University of Tennessee, Knoxville
Knoxville, TN
jchoi38@vols.utk.edu

Shashank Bandaru
EECS
University of Tennessee, Knoxville
Knoxville, TN
sbandar1@vols.utk.edu

***Abstract*—This project explores using genetic algorithms (GAs) to evolve strategic traits in simulated poker players competing in no-limit Texas Hold'em tournaments. Each player is defined by a set of quantifiable behavioral traits—such as aggressiveness, bluff tendency, chip size awareness, and others—that influence in-game decision-making. Players are evaluated through simulated games and selected using tournament selection; crossover and mutation are used to produce the next generation. Over 50 generations and hundreds of parameter combinations, we analyze how traits evolve, which traits correlate with success, and how fitness scores respond to algorithmic configurations. Results indicate that balanced players with slightly elevated risk tolerance and bluff tendency achieve the highest performance. The findings demonstrate the viability of evolutionary methods in identifying effective strategic behavior in complex, imperfect-information games.**

***Index Terms*—Genetic Algorithms, Poker, Player Traits**

I. Introduction and Motivation

Poker is a game enjoyed by many, with aspects of both skill and luck intertwined throughout every hand. The 2024 World Series of Poker Main Event had just over ten thousand contestants, with the winner walking away with an astonishing 94 million dollars. With so much money on the line, poker players need to hone their skills, especially during hands where luck is not on their side.

Success in poker is rarely about simply getting good cards. Instead, it often comes down to how players use strategy, psychology, and risk management to gain an edge over their opponents. There are countless traits and tactics that players develop over time to navigate the uncertainty of the game. Some traits, such as the ability to effectively mask one's emotions, can be vital for actions like bluffing, where hiding weakness can turn a losing hand into a winning one. Other traits, like understanding chip sizes, position at the table, and knowing when to play aggressively versus cautiously, can have just as much of an impact on long-term success. Given the complexity of decision-making in poker, it becomes an interesting and valuable challenge to study which traits are the most effective.

II. Related Work

Diving into related work, we discover that there are many uses of genetic algorithms in order to better understand game and player mechanics in different games. Some of these works specifically deal with poker in a similar manner to our project, while others aim to identify other quirks of the game.

A. *Poker Related Works*

Carter et. al. [1] focused on the information aspect of poker, attempting to determine whether a player is more successful in surviving a round of poker if they have access to information about the table or are limited to their hands. The paper goes further into mentioning certain aspects of table-based information that provide players with an advantage, such as chip stack size and position in the table. This reflects our project as we vary traits such as chip size awareness, similar to chip stack size, to determine if it is a useful trait in poker.

Likewise, another work focuses on the opponents rather than the player. By utilizing neural networks and genetic algorithms to train them, Xu et. al. [2] created networks to determine traits that opponents exhibit during a poker game to exploit them. This is an interesting take as it primarily focuses on finding flaws and weaknesses in fellow players rather than finding optimal traits to have during a poker game.

Furthermore, we see many works that focus on the core question that our project is trying to answer: what traits make a good poker player? Using genetic algorithms, many of these papers come to similar conclusions as our project, where some traits are hard to simulate or determine whether or not they are effective [3]. Alternatively, some works aim to develop models that train and adapt independently to maximize poker winnings, some even going further to tackle the issue of deliberate disinformation that is present in poker gameplay [4], [5], [6].

B. *General Game-Related Works*

More generally, genetic algorithms are used in many game-related studies, often aimed at determining how the player should or could act or how the game does overall [7]. One such use is in fighting games where Zuin et. al. use genetic algorithms to discover effective combos (combination of inputs) [8]. In this approach, the player is not evolved; rather, it is the combos found that are evolved, based on how well they perform in an actual match.

Another paper looks at the card-based fighting game, Hearthstone, in which Sánchez et. al. point out the issues in testing for flaws in the game when new cards are introduced [9]. They then go on to propose using genetic algorithms to create decks of cards that can be tested, forgoing the traditional method of iteratively creating all possible decks and testing each one out. Weber, et. al. take this approach a little differently, using genetic algorithms to dynamically change the difficulty of a game to ensure that the player does not become bored or frustrated [10].

These works represent how genetic algorithms can be used similarly to determine certain aspects of game behavior, from testing to the actual gameplay.

III. Methods

This project aims to determine which traits are the most effective when it comes to winning poker. Traits for poker vary widely, so it was crucial to choose a set of traits that could be easily quantifiable. As such, we chose the following traits to test using genetic algorithms:

- Aggressiveness: a measure of how likely a player is to raise more chips rather than check or call.
- Risk tolerance: a measure of how likely a player is to take chances.
- Bluff tendency: a measure of how likely a player is to play as if they have strong hands, despite the fact that they are holding weak cards.
- Position awareness: how much a player adjusts their behavior based on table position.
- Chip size awareness: how much a player adjusts the amount of chips they are betting based on the number of chips that they currently have.

Using these chosen traits, we implemented a playground where populations can be initialized with random traits and allowed to evolve throughout 50 generations using genetic algorithms.

A. Texas Hold'em Poker Background

Texas Hold'em is a variant of poker that combines community cards (shared by all players) with hole cards (cards held by the player). The game's objective is to win chips by either having the best hand at showdown or by forcing opponents to fold before reaching showdown. Chips are a representation of money denominations and are pre-determined throughout the game. The strategy portion of poker comes from incomplete information (players cannot see opponents' hole cards) and the sequential nature of betting decisions. It becomes both a psychological and logical battle between players to come out on top.

a) *Game Structure:* A game of poker typically starts with 2-10 players at a table. At a table, certain positions hold important roles in the game. The dealer will shuffle and deal the cards to the other players after each game. The small-blind (SB) forces the player to place a small bet before seeing any community cards, and the big-blind (BB) forces a bigger bet before seeing community cards. The SB is positioned next to the dealer in the clockwise direction and is the player who is first in the betting round. The BB comes after the SB. After every showdown, the positions are rotated clockwise so that new players will hold these key positions. When players place bets, they are placed into the "pot", which is the total sum of all the players' bets that have been placed throughout the rounds.

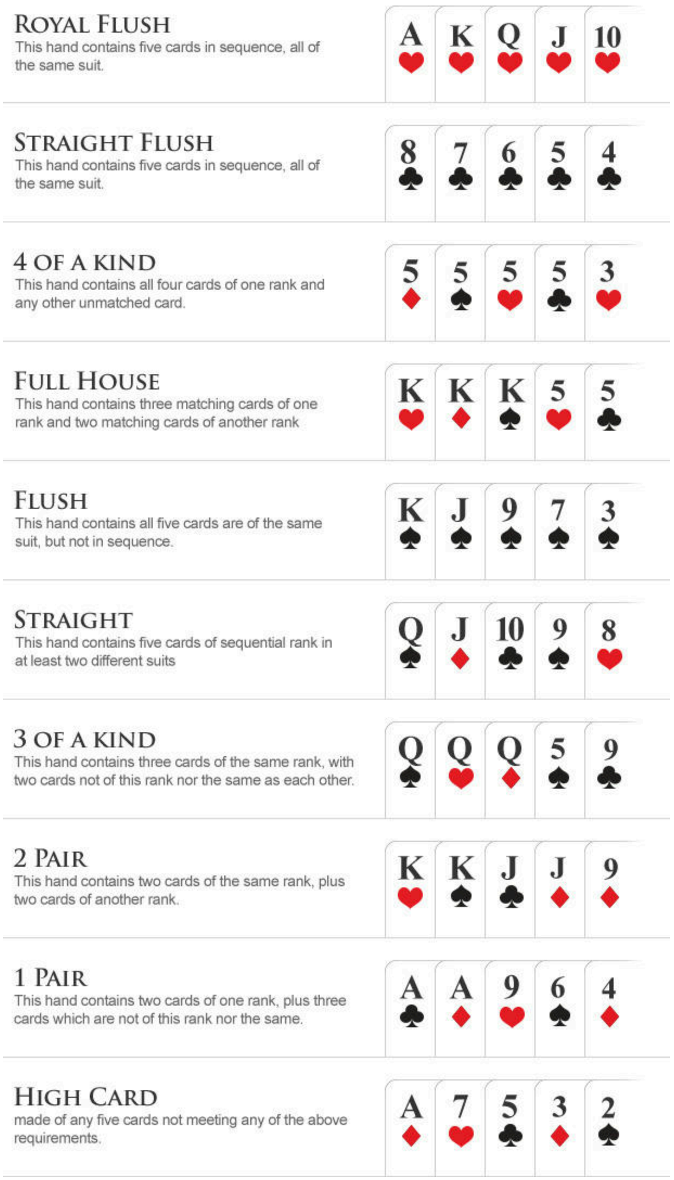
b) *Hand Structure:* Each hand of Texas Hold'em consists of four betting rounds:

- Pre-flop: Each player receives two private cards (hole cards), followed by a betting round
- Flop: Three community cards are dealt face-up, followed by a betting round
- Turn: A fourth community card is dealt, followed by a betting round
- River: A fifth and final community card is dealt, followed by a final betting round
- Showdown: If multiple players remain, they reveal their hands, and the best five-card hand wins

c) *Player Actions:* During each betting round, players can choose from these actions:

- Fold: Discard hand and forfeit any chance at the current pot.
- Check: Pass the action to the next player (only available if no bet has been made).
- Call: Match the current bet amount.
- Bet/Raise: Increase the amount that subsequent players must call to stay in the hand.

d) *Hand Rankings:* A player's hand, combined with the community cards, determines the rank of their hand. Within a table at showdown, the player with a higher rank will win the pot. A breakdown of the possible hands ranking from top down is shown in Fig. 1.



ROYAL FLUSH	This hand contains five cards in sequence, all of the same suit.	A♥ K♥ Q♥ J♥ 10♥
STRAIGHT FLUSH	This hand contains five cards in sequence, all of the same suit.	8♣ 7♣ 6♣ 5♣ 4♣
4 OF A KIND	This hand contains all four cards of one rank and any other unmatched card.	5♦ 5♠ 5♥ 5♣ 3♥
FULL HOUSE	This hand contains three matching cards of one rank and two matching cards of another rank	K♥ K♦ K♠ 5♥ 5♣
FLUSH	This hand contains all five cards are of the same suit, but not in sequence.	K♠ J♠ 9♠ 7♠ 3♠
STRAIGHT	This hand contains five cards of sequential rank in at least two different suits	Q♠ J♦ 10♣ 9♠ 8♥
3 OF A KIND	This hand contains three cards of the same rank, with two cards not of this rank nor the same as each other.	Q♠ Q♥ Q♦ 5♠ 9♣
2 PAIR	This hand contains two cards of the same rank, plus two cards of another rank.	K♥ K♠ J♣ J♦ 9♦
1 PAIR	This hand contains two cards of one rank, plus three cards which are not of this rank nor the same.	A♣ A♦ 9♥ 6♠ 4♦
HIGH CARD	made of any five cards not meeting any of the above requirements.	A♦ 7♥ 5♣ 3♦ 2♠

Fig. 1: Rankings of Poker Hands Ranking from best at the Top to Worst at the Bottom

In the case where there are multiple players at showdown with the same highest hand, we determine the winner by comparing the values of the hands. When the highest hands have both the same rank and the same value, a tie occurs, and the pot is split between the winning players.

e) *Player Considerations:* When a player considers what move to make on their turn, there are some key concepts that they typically consider:

- Position: How many players make a move before or after their turn to gauge opponents' actions.
- Pot Odds: The ratio of the current pot size to the cost of a contemplated call.
- Expected Value: The average amount a player can expect to win or lose on a bet in the long run.
- Hand Ranges: The spectrum of possible hands an opponent might hold based on their actions.
- Hand Possibilities: The possibility of getting a better hand through more community cards being revealed.
- Bluffing: Betting or raising with a weak hand to represent strength and induce folds.
- Value Betting: Betting with a strong hand to extract maximum value from weaker hands.

B. *Texas Hold'em Poker Simulation*

In our simulation, we've added various changes that make our poker simulation playable through simulated players while minimally impacting the fundamental game mechanics. For the purposes of this project, we've implemented a couple of rules that limit the complexity of the development. For example:

- $5000 Buy-In
- $100 Big-Blind
- $50 Small-Blind
- No Wrap-Around (ace can only be used as the highest card)
- Playing with table stakes only (players can't add more money than they started with)
- No "cash-out" (players cannot leave a table unless all their money is spent or they've won all the money from other players)

To keep the simulation as close to tournament-style poker, players will be playing with chips of this denomination:

- (20) White: $50
- (10) Red: $100
- (4) Green: $250
- (2) Blue: $500
- (1) Black: $1000

where the number before the color of the chip indicates how many of that chip they receive.

In order to facilitate the simulated players to make decisions and perform actions, we need to make sure that the game runs smoothly in all cases without intervention.

One case would be trading in chips accurately if a player can't bet using the same denomination of chips. This process is automatically calculated to give trade-in chips to match the denomination needed as best as possible.

We must also consider the case of multiple winners where there is a tie in hand ranking and value. In this case, the pot will be split between the winners, but the complication comes when the split pot cannot be represented through the denomination of chips (i.e., splitting $100 between 3 players). In this case, the pot will be split as evenly as it can, given the denomination, and the remaining chips that cannot be split will be awarded to the first player who won that comes counter-clockwise from the dealer.

Moreover, in the case of all-ins, special rules are made so that a player who goes all-in is not better off or worse off than the other players based on the money that they have. For instance, when two players are playing and player one has significantly more money available than player two, if player two goes all-in and wins, they can only win the pots that they have contributed to. So in the implementation of a poker game, whenever a player goes all-in, a side-pot is created, and if more players can bet more money than what the player went all-in for, then that money gets added to these side-pots. Then, if the player that went all-in wins, they can only win the pots that they have contributed to, and the rest of the pots are distributed across the winners of the pots that contributed to those side pots.

C. Player Implementation

In our genetic algorithm framework, each simulated poker player is represented as a collection of six fundamental traits that influence decision-making throughout gameplay. These traits are encoded as floating-point values ranging from 0.0 to 1.0, where higher values indicate stronger expression of that particular characteristic. This normalization allows for consistent genetic operations while providing sufficient granularity to model complex behavioral patterns. Each trait was selected to model a key aspect of poker strategy observed in human players.

a) *Decision Making:* When it comes time for a player's turn, their cards will be evaluated to see how good their cards are with the cards on the table. Based on this and their values for each feature, they will make a decision.

The way that decisions are made is as follows. If a player has high-ranking cards, they will bet higher if they have a higher aggressive trait. Otherwise, if a player has medium-ranking cards, they will bet higher if they have a higher probability of risk tolerance. Lastly, if a player has low-ranking cards, they will bet higher if they have a high bluff tendency; otherwise, they will fold.

After it has been determined whether a player will raise, chip size awareness and position awareness come into play. The player will be willing to bet more chips if they have a higher chip stash, if they are more chip aware; on the other hand, they will bet fewer chips if they have a lower chip stash and are chip aware. If a player has higher position awareness, they will bet more if they are the last person to bet.

D. Genetic Algorithm

Genetic algorithms (GAs) are a class of evolutionary optimization techniques inspired by natural selection. They work by iteratively evolving a population of candidate solutions through processes such as selection, crossover, and mutation. In our project, we use a genetic algorithm to evolve simulated poker players, each defined by a set of behavioral traits. The GA enables us to explore how combinations of strategic tendencies—such as aggressiveness, bluffing, and adaptability, etc.—affect long-term performance in simulated poker tournaments, to discover high-performing strategies without relying on explicit game-theoretic models.

a) *Evolutionary Cycle:* The evolutionary process follows a standard genetic algorithm workflow:

- Evaluation: Players compete in Texas Hold'em tournaments to determine fitness
- Selection: Tournament selection identifies promising individuals
- Recombination: Crossover and mutation create the next generation
- Statistical Collection: Performance metrics are gathered for analysis

This cycle repeats for a configurable number of generations.

b) *Parameter Configurations:* For our genetic algorithm, we follow typical parameter configurations:

- Population Size: Total number of players per generation
- Tournament Size: Number of individuals in each selection tournament
- Crossover Probability: Likelihood that crossover occurs between two parents
- Mutation Probability: Likelihood that a trait undergoes mutation
- Generations: Number of evolutionary cycles to simulate

But we also include our simulation-specific parameters, specifically:

- Round Cutoff: Maximum number of poker rounds per table. Used to terminate unending games and ensure a timely simulation.
- Max Players per Table: Limits the number of players per game table to maintain realistic gameplay dynamics.
- Iterations: Number of tournaments played per generation per table.

c) *Population:* The population is initialized by creating a set of poker players. Each player receives a unique identifier for tracking, randomized trait values (aggressiveness, risk_tolerance, bluff_tendency, adaptability, position_awareness, chip_size_awareness) between 0.0 and 1.0, a unique lineage identifier for genealogical tracking, and initial poker game state parameters. This initialization ensures population diversity in the first generation, providing a broad sampling of the strategy space for evolutionary optimization. Every player is given a new set of chips at the beginning of each generation.

d) *Fitness:* The fitness function in our genetic algorithm is critical for determining which player traits are advantageous and should be propagated to future generations. We developed a fitness evaluation that considers short-term performance, long-term viability, and evolutionary consistency. This approach ensures that successful traits are not merely the result of fortunate card distribution but represent genuinely superior poker strategies. Our fitness function employs a weighted multi-metric approach that balances three key performance indicators:

- Survival Rank (20% weight): Players are ranked based on their longevity in the tournament and final chip count. This metric rewards players who survive longer and accumulate more chips:

  $$R = \frac{\{N-r\}}{\{N-1\}}$$

  Where:

  - N: Total number of players at the table
  - r: Player's rank position (0 for first place, $N-1$ for last place)
  - R: Normalized rank score between 0.0 (first eliminated) and 1.0 (winner)

  This produces a normalized score between 0.0 (first eliminated) and 1.0 (tournament winner).

- Chip Efficiency (30% weight): This metric evaluates how effectively a player accumulates chips relative to the number of rounds they have played:

$$G = \frac{\mathrm{Cf} - \mathrm{Ci}}{\mathrm{Ci} \times S}$$

Where:
- Ci: initial chip stack
- Cf: final chip stack
- S: rounds survived
- G: normalized chip growth per round

This metric identifies players who consistently make profitable decisions, even if they don't ultimately survive the longest.

- Lineage History (50% weight): This metric tracks ancestral performance across generations, implementing a form of evolutionary memory:

$$\mathrm{L} = d \times \mathrm{Lp} + (1 - d) \times P$$

Where:
- L: current lineage fitness
- Lp: previous generation's lineage fitness
- P: current performance score (combined R + G)
- d: decay factor (0.9)

The decay factor (0.9) ensures that recent performance has a greater impact while still maintaining continuity with successful ancestors. This encourages the development of stable, consistently successful trait combinations rather than high-variance strategies that occasionally succeed through luck.

The final fitness score is calculated as a weighted combination of these three metrics:

$$F = 0.5L + 0.2R + 0.3G$$

This balanced approach ensures that players must demonstrate multiple positive attributes to achieve high fitness scores.

e) *Tournament Selection:* We use random sampling to select k number of players within the population, and within the selected players, the two players with the best fitness are selected as parents for the next generation of children.

This method balances exploitation (by favoring high-performing individuals) and exploration (through random sampling), helping prevent premature convergence to local optima.

f) *Crossover:* In order to maintain various traits between previous generations, we employ arithmetic crossover to generate a new offspring for the next generation.

$$\text{child\_trait} = \frac{\text{parent1\_trait} + \text{parent2\_trait}}{2}$$

Each new offspring is assigned a unique identifier, a marked reference to the parents they inherited from, and traits calculated from the arithmetic mean of their parents' traits. Crossover is dictated by the crossover probability parameter. If crossover does not happen, then a child's traits are copied from one of the parents chosen at random.

g) *Mutation:* To maintain genetic diversity and avoid premature convergence, we apply a mutation step after crossover. Mutation introduces small random changes to an individual's traits, allowing the population to explore new regions of the strategy space that may not be reachable through crossover alone. However, this mutation step only has a set probability of occurring.

In our implementation, each trait of the offspring has its value altered by adding a random floating-point number drawn uniformly from the range 0.0 - 1.0. To ensure that the resulting value remains within the normalized trait bounds [0.0, 1.0], we use circular wrapping via the modulo operator:

$$\text{trait\_new} = (\text{trait\_old} + \text{mutation\_amount}) \bmod 1.0$$

This strategy ensures that even evolved strategies with highly tuned trait values are subject to occasional variation, which supports long-term adaptability and helps escape local optima. Moreover, when a player mutates, we assign the player a new lineage ID as well to introduce lineage diversity.

h) *Experimental Setup:* To evaluate the performance and adaptability of evolved poker strategies, we conducted a series of simulations under varying genetic algorithm configurations. The goal was to analyze the effect of key parameters—such as population size, mutation rate, and tournament size—on the emergence of effective poker-playing behaviors.

We systematically varied the following parameters, testing all combinations of their values:
- Population Size: 25, 50, 75
- Crossover Probability: 0.2, 0.5, 0.8
- Mutation Probability: 0.2, 0.5, 0.8
- Max Players per Game Table: 6, 8
- Tournament Size (k): 5, 10, 20, 30, 40, 50
- Round Cutoff: 3000 (fixed)

For each configuration, a single tournament was played per generation over 50 generations.

This exhaustive combination of parameters enabled us to explore the algorithm's sensitivity to genetic and environmental conditions and to determine which configurations best support the emergence of high-performing poker strategies.

E. Data collection

To support rigorous analysis of evolutionary dynamics and poker-playing behavior, our simulation framework collects detailed data at both the population and individual levels across all generations and experimental configurations. a) *Population-Level Statistics:* For each generation, we compute and store the following aggregated statistics:

- Average Fitness: Mean fitness score across all players

Average Rounds Lasted: Mean number of poker rounds survived

- Average Trait Values: Mean value for each strategic trait:
  - aggressiveness
  - risk_tolerance
  - bluff_tendency
  - adaptability
  - position_awareness
  - chip_size_awareness
- Average Action Frequencies: Mean number of times players performed each action:
  - bluff_attempts
  - fold
  - raise
  - call
  - all_in
  - check

These metrics help assess the overall behavioral trends of the evolving population and detect convergence or diversification in strategy.

b) *Lineage and Individual Player Histories:* Each player is uniquely identified and tracked throughout their lineage. For every generation and individual, we record:

- Unique player ID and lineage ID
- Fitness and lineage-level average fitness
- Number of rounds survived
- Final table position
- Full trait vector
- Detailed action counts (e.g., number of raises, folds, etc.)

This fine-grained tracking allows us to analyze how successful traits propagate over generations, how individual strategies evolve, and how lineage performance fluctuates over time.

c) *Data Storage:* For each experiment configuration, two CSV files are saved:

- population_stats.csv: Contains generation-level averages for each population
- lineage_history.csv: Contains individual-level performance and trait/action data for genealogical analysis

Each file includes an iteration field, enabling multi-run aggregation and statistical comparison across independent runs of the same configuration. Data is preprocessed before being saved into a file. First, we convert nested generation-level dictionaries into a flat tabular format, we also flatten individual-level lineage data for export.

This structure simplifies data processing for visualization, statistical modeling, and trend analysis.

IV. Results

Visualizing the results would aid in answering the ultimate question of the project

A. The Top 5% of Players

The first inspection of the data included answering the question of the most effective traits, which included data from the top 5% of players (top 5% of fitness scores from all the produced CSV files):

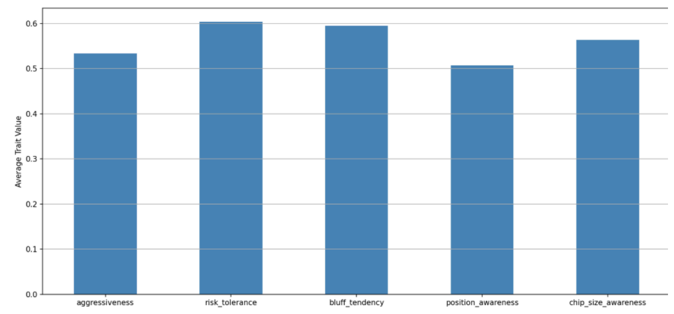


Fig. 2: Average Traits of Top 5% of Players Across All Experiments

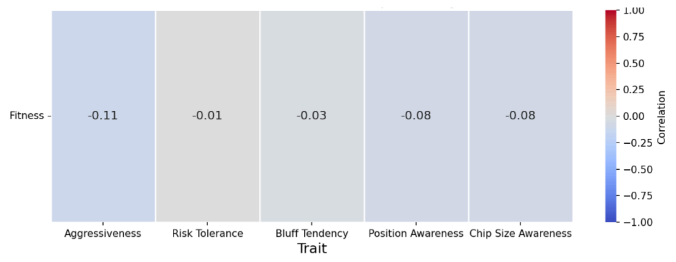


Fig. 3: Traits vs. Fitness Correlation (Top 5% of Players)

Through the data collection and visualizations, the research question has now been answered. Based on Fig. 2, having a relatively higher risk tolerance and bluff tendency, but relatively lower traits of aggressiveness, chip size awareness, and position awareness, makes for the best poker player. However, all of the traits are around the 0.5 trait value, with some slightly higher and some slightly lower. It makes logical sense that a player with average traits wins the most, as they are playing their hands a bit more safely, thus saving them from losing too much of their money.

Observing Fig. 3, one may see that all the traits have a very slight negative correlation with fitness, with aggressiveness having the most negative correlation. In other words, players who had higher traits of aggressiveness were slightly less likely to have higher fitness scores. Risk tolerance and bluff tendency had very minimal correlation with fitness scores. This is quite interesting because of the bar graph in Fig. 2.

B. Studying the Fitness Scores

Another aspect of the data that was looked at included how long it took for the genetic algorithm variables to stabilize:

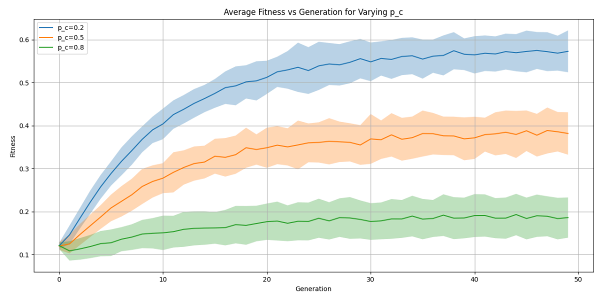


Fig. 4: Average Fitness vs. Generation for Varying Crossover Probability

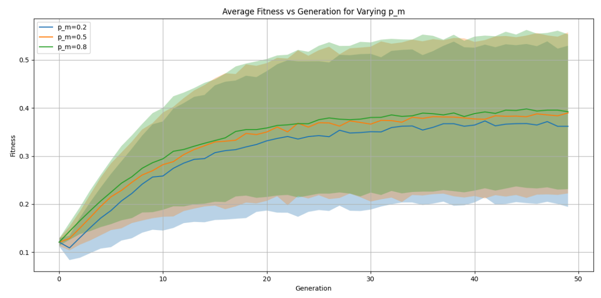


Fig. 5: Average Fitness vs. Generation for Varying Mutation Probability

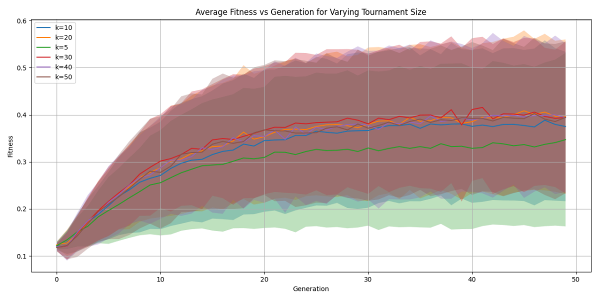


Fig. 6: Average Fitness vs. Generation for Varying Tournament Size

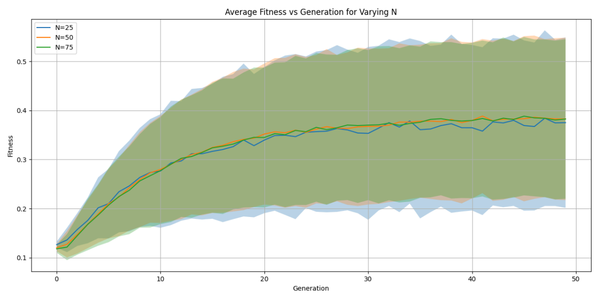


Fig. 7: Average Fitness vs. Generation for Varying Population Size

It was important to see how the fitness changed across generations to make sure that it was increasing as was expected:

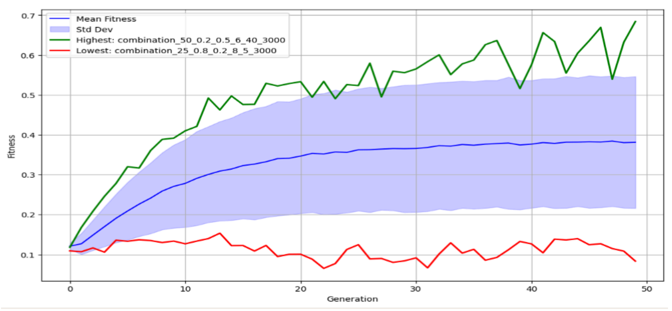


Fig. 8: Mean Fitness per Generation Across All Combinations

It is helpful to study the graphs of varying parameters to see how these parameters impact fitness over generations. Figs. 4-7 show the crossover probability, mutation probability, population size, and tournament size parameters varying. In Fig. 4, one may see that the crossover probability values do not converge in the slightest. Thus, it may be deduced that lower crossover probabilities lead to higher fitness scores. Too much crossover can disturb traits from evolving and getting better.

For the mutation probabilities in Fig. 5, it can also be seen that higher mutation probabilities lead to higher fitness scores. Tournament sizes, shown in ts, for the most part, converge except when that tournament size is too small a value. As shown in Fig. 7, population sizes also seem to converge. Thus, these values do not seem to impact the fitness value too much.

As one may see in Fig. 8, the mean fitness is increasing as the generations go on. This is a sanity check, as well, to ensure that the traits are evolving in the way one would expect them to (ensuring higher fitness scores). It also shows that the best combination increases its fitness score by a major amount, while the worst combination does not improve but actually worsens.

C. *Lineage History*

Lastly, looking at lineage history would help observers see how traits evolved:

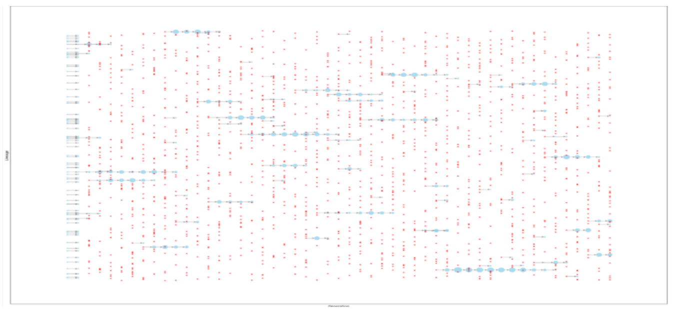


Fig. 9: Lineage Evolution Timeline Example

In Fig. 9, blue dots and blue lines indicate lineages that have survived and did not mutate. Red dots, on the other hand, show mutations that do not live on to produce more offspring. There are quite a few red dots, but also a good amount of blue dots and lines. This is another check to ensure the traits are changing, but also prioritizing higher fitness values.

V. Discussion

Because of the results obtained from Figs. 2-3, it seems as if a balanced player is more likely to win a game of poker. As stated previously, the traits are all close to 0.5, and there is minimal correlation between the traits and the fitness value (with more correlation between the traits and aggressiveness). It is interesting to see because one may predict that chip size awareness and position awareness would have positive correlations. However, these simply control the amount someone is betting. Thus, if that person loses, they are losing that number of chips. This will even out with winning that many chips.

Figs. 4-7 may be important when reproducing the study. One should use values of the crossover probability and mutation probability that produce the highest fitness scores to achieve even better traits to study. Smaller crossover probabilities have higher fitness scores, most likely because there are fewer combinations of the genes when producing offspring; it will, instead, keep better genes. On the other hand, larger mutation probability values have higher fitness scores because there is more diversity, leading to more diverse traits being tested. The best curve for optimal fitness values across generations is also now known as improving over time.

Lastly, Fig. 9 explores how lineages evolve. Mutation helps with the diversity of traits, but traits that continue evolving can lead to higher fitness scores.

VI. Conclusion

AI agents simulated playing rounds of Texas Hold'em poker with genetic algorithms used for evolving traits to find the best traits for a poker player to have. All in all, this project showed that a poker player with balanced traits (of aggressiveness, risk tolerance, bluff tendency, chip size awareness, and position awareness) is the best one to be; a slightly higher risk tolerance and bluff tendency trait is also good to have. The findings suggest that evolving traits through genetic algorithms is an effective way for identifying high-performing strategies within competitive games.

VII. Future Work

If more time had been available, we would have implemented multi-processing in the simulation to run more iterations in less time. Additionally, we would have improved the hand evaluation method to include not only what was in the player's hand and on the table but also the probabilities of what was still left in the deck of cards.

One promising direction for expanding our project is the integration of Counterfactual Regret Minimization (CFR) [11], a well-established algorithm in game theory and poker AI research. While our current simulation operates in the more complex and realistic domain of no-limit multi-player Texas Hold'em, where players can bet any amount of their available chips at any time. CFR has already nearly "solved" limit Texas Hold'em by finding unbeatable strategies under fixed betting rules. Although this project focuses on no-limit poker, CFR methods could still be useful for comparisons or experiments. Specifically, we propose several avenues for future exploration:

- Benchmarking Against CFR Agents: By implementing simplified CFR-based agents in a heads-up limit Hold'em environment, we could compare the evolutionary outcomes of our genetic algorithm (GA) against theoretically grounded CFR strategies. This would provide a baseline to assess how well evolved traits approximate optimal play in constrained game settings.
- Trait Evolution Under CFR Pressure: We could pit CFR agents against our GA-evolved players in limit Hold'em settings to observe how genetic traits adapt in response to theoretically optimal opponents. This could yield insights into which behavioral traits are reinforced or suppressed under near-optimal strategic pressure.
- Hybrid Models: There is potential to develop hybrid strategies by seeding initial populations with CFR-inspired agents or traits (e.g., bluff frequency calibrated from regret values), blending theory-driven and evolution-driven approaches to strategy formation.

This project also opened up new research questions. One includes how combinations of traits (i.e., high risk tolerance and high bluff tendency) affect players' performance. Another involves how varying only the crossover and mutation probability parameters affects the fitness scores and traits of the players. Lastly, it opens the door to finding the best traits for other decision-making games, such as blackjack. Addressing these questions could provide more insight into how genetic algorithms shape strategic decision-making in complex games like poker.

References

[1] R. G. Carter and J. Levine, "An Investigation into Tournament Poker Strategy using Evolutionary Algorithms," in *2007 IEEE Symposium on Computational Intelligence and Games*, 2007, pp. 117–124. doi: 10.1109/CIG.2007.368087.

[2] J. Xu and S. Chen, "A Neuroevolutionary Approach for Opponent Modeling and Exploitation in No-limit Texas Hold'em Poker," in *2021 China Automation Congress (CAC)*, 2021, pp. 2270–2275. doi: 10.1109/CAC53003.2021.9727922.

[3] H. Quek, C. Woo, K. Tan, and A. Tay, "Evolving Nash-optimal poker strategies using evolutionary computation," *Frontiers of Computer Science in China*, vol. 3, no. 1, pp. 73–91, Mar. 2009, doi: 10.1007/s11704-009-0007-5.

[4] L. Barone and L. While, "An adaptive learning model for simplified poker using evolutionary algorithms," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, pp. 153–160. doi: 10.1109/CEC.1999.781920.

[5] D. Billings, "Algorithms and assessment in computer poker," doi: 10.7939/R3-W2XJ-AV70.

[6] G. Nicolai and R. J. Hilderman, "No-Limit Texas Hold'em Poker agents created with evolutionary neural networks," in *2009 IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 125–131. doi: 10.1109/CIG.2009.5286485.

[7] S. Lucas and G. Kendall, "Evolutionary computation and games," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 10–18, 2006, doi: 10.1109/MCI.2006.1597057.

[8] G. L. Zuin, Y. P. Macedo, L. Chaimowicz, and G. L. Pappa, "Discovering Combos in Fighting Games with Evolutionary Algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, in GECCO '16. Denver, Colorado, USA: Association for Computing Machinery, 2016, pp. 277–284. doi: 10.1145/2908812.2908908.

[9] P. García-Sánchez, A. Tonda, A. M. Mora, G. Squillero, and J. J. Merelo, "Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone," *Knowledge-Based Systems*, vol. 153, pp. 133–146, 2018, doi: https://doi.org/10.1016/j.knosys.2018.04.030.

[10] M. Weber and P. Notargiacomo, "Dynamic Difficulty Adjustment in Digital Games Using Genetic Algorithms," in *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 2020, pp. 62–70. doi: 10.1109/SBGames51465.2020.00019.

[11] T. W. Neller and M. lanctot, [Online]. Available: http://modelai.gettysburg.edu/2013/cfr/cfr.pdf