An Evolutionary Game-Theoretic Analysis of Poker Strategies

Marc Ponsen, Universiteit Maastricht, Netherlands
Karl Tuyls, Technische Universiteit Eindhoven, Netherlands
Michael Kaisers, Technische Universiteit Eindhoven, Netherlands
Jan Ramon, Katholieke Universiteit Leuven, Belgium

December 15, 2008

Abstract

In this paper we investigate the evolutionary dynamics of strategic behaviour in the game of poker by means of data gathered from a large number of real world poker games. We perform this study from an evolutionary game theoretic perspective using two Replicator Dynamics models. First we consider the basic selection model on this data, secondly we use a model which includes both selection and mutation. We investigate the dynamic properties by studying how rational players switch between different strategies under different circumstances, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. We illustrate the dynamics using a simplex analysis. Our experimental results confirm existing domain knowledge of the game, namely that certain strategies are clearly inferior while others can be successful given certain game conditions.

1 Introduction

Although the rules of the game of poker are simple, it is a challenging game to master. There exist many books written by domain experts on how to play the game (see, e.g., [3, 5, 12]). A general consensus is that a winning poker strategy should be adaptive: a player should change the style of play to prevent becoming too predictable, but moreover, the player should adapt the game strategy based on the opponents. In the latter case, players may want to vary their actions during a specific game (see, e.g., [2, 10, 13]), but they can also consider changing their strategy over a series of games (e.g., play a more aggressive or defensive style of poker).

In this paper we perform an Evolutionary Game Theoretic analysis of poker strategies based on data from real world poker games played between human players. More precisely, we investigate the strengths of a number of poker strategies facing some opponent strategies using Replicator Dynamics (RD) models

[7, 8, 16, 18]. The RD are a system of differential equations describing how strategies evolve through time. In this paper we investigate two of such models. The first RD model only includes the biological selection mechanism. Studies from game theory and reinforcement learning indicate that people do not behave purely greedy and rational in all circumstances but also explore different available strategies (to discover optimal strategies) for which they are willing to sacrifice reward in the short term [4, 15]. We believe it is critical to include mutation, as an exploration factor, to the RD model to find accurate results. Therefore we also apply a second RD model that includes both selection and mutation.

A complicating factor is that the RD can only be applied straightforwardly to simple normal form games (NFG) as for instance the Prisoner's Dilemma game [4]. The game of poker is too complex to be represented in such a way. Therefore we define heuristic strategies, i.e., strategic behavior over large series of games, and compute a heuristic payoff table that assigns payoffs to these strategies. This approach has been used before in the analysis of behaviour of buyers and sellers in automated auctions [9, 19, 20]. Conveniently, for the game of poker several heuristic strategies are already defined in poker literature and can be used in our analysis.

The innovative aspects of our work are twofold: firstly, although there are good classical game-theoretic studies of poker, they are mainly interested in the static properties of the game, i.e. what the Nash equilibria are and how to explicitly compute or approximate them. Due to the complexity of this computation, usually only some simplified versions of poker are considered (e.g., see [1]). Instead we take an evolutionary perspective towards this game using two different RD models. This allows us to investigate the dynamic properties by studying how rational players switch between different strategies under different circumstances, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. These new insights help to unravel the complex game of poker and may prove useful for strategy selection by human players but can also aid in creating strong artificial poker players.

Secondly, for this analysis we use real world data that we obtained by observing poker games at an online website, wherein human players competed for real money at various stakes. From this real world data the heuristic payoff table is derived, as opposed to the artificial data used in the previously mentioned auction studies. By analyzing real world data we can empirically validate the claims put forward by domain experts on the issue of strategy selection in poker.

The remainder of this paper is structured as follows. We start by explaining the poker variant we focus on in our research, namely No-Limit Texas Hold'em poker, and describe some well-known strategies for this game. Next we elaborate on the RD and continue with a description of our methodology. We end with experiments and a conclusion.

2 Background

In this section we will first briefly explain the rules of the game of poker. Then we will discuss ways to categorize poker strategies as was proposed by domain experts.

2.1 Poker

Poker is a card game played between at least two players. In a nutshell, the objective in poker is to win games (and consequently win money) by either having the best card combination at the end of the game, or by being the only active player. The game includes several betting rounds wherein players are allowed to invest money. Players can remain active by at least matching the largest investment made by any of the players, or they can choose to fold (i.e., stop investing money and forfeit the game). The winner receives the money invested by all the players.

In this paper we focus on the most popular poker variant, namely No-Limit Texas Hold'em. This game includes 4 betting rounds (or phases), respectively called the pre-flop, flop, turn and river phase. During the first betting round, all players are dealt two private cards (what we will now refer to as a player's *hand*) that are only known to that specific player. To encourage betting, two players are obliged to invest a small amount the first round (the so-called small- and big-blind). One by one, the players can decide whether or not they want to participate in this game. If they indeed want to participate, they have to invest at least the current bet. This is known as *calling*. Players may also decide to *raise* the bet. If they do not wish to participate, players *fold*, resulting in possible loss of money they bet thus far. A betting round ends when no outstanding bets remain, and all active players have acted. During the remaining three betting phases, the same procedure is followed. In every phase, community cards appear on the table (respectively 3 in the flop phase, and 1 in the other phases). These cards apply to all the players and are used to determine the card combinations (e.g., a pair or three-of-a-kind may be formed from the player's private cards and the community cards). After the last betting round the card combinations for active players are compared during the so-called showdown.

2.2 Classifying poker strategies

There exists a lot of literature on winning poker strategies, mostly written by domain experts (see, e.g., [3, 5, 12]). These poker strategies may describe how to best react in detailed situations in a poker game, but also how to behave over large numbers of games. Typically, experts describe poker strategies (i.e., behavior over a series of games) based on only a few aggregate features. For example, an important feature in describing a player's strategy is the percentage of times this player voluntarily invests money during the pre-flop phase and then sees the flop (henceforth abbreviated as *VPIP*), since this may give insight in the player's hand selection. If a particular player sees the flop more than, let's

say, 40% of the games, he or she may play with less quality hands (see [12] for hand categorization) compared to players that only see the flop rarely. The standard terminology used for respectively the first approach is a *loose* and for the latter a *tight* strategy.

Another important feature is the so-called *aggression-factor* of a player (henceforth abbreviated as AGR). The aggression-factor illustrates whether a player plays offensively (i.e., bets and raises often), or defensively (i.e., calls often). This aggression factor is calculated as:

$$\frac{\%\text{bet} + \%\text{raise}}{\%\text{calls}}$$

A player with a low aggression-factor is called *passive*, while a player with a high aggression-factor is simply called *aggressive*.

3 Methodology

In this section we explain the methodology we will follow to perform our analysis of poker strategies. We start by explaining the RD and the heuristic payoff table that is used to derive average payoffs for the various poker strategies. Finally, we describe our algorithm for visualizing and analyzing the dynamics of the different strategies in a simplex plot.

3.1 Replicator Dynamics

The RD [16, 23] are a system of differential equations describing how strategies evolve through time. We assume an infinitely large population of "individuals" (i.e., players). Each player may apply one of the available "replicators" (i.e., strategies). The pure strategy i is played with probability x_i, according to the vector $x = (x_1, \ldots, x_k)$. The profit of each player depends on x. At each time step, players may switch their strategies based on the profits received (i.e., they switch to more successful strategies). As a consequence, the probabilities of strategies are changed. This adaptation can be modeled by the RD from evolutionary game theory.

An abstraction of an evolutionary process usually combines two basic elements, i.e., selection and mutation. Selection favors some population strategies over others, while mutation provides variety in the population. In this research, we will consider two RD models for our analysis. The first one is based solely on selection of the most fit strategies in a population. The second model, which is based on Q-learning and is formally derived in [17, 18], includes mutation besides selection. We now formally describe both models.

3.1.1 Replicator Dynamics: the basic model

As it is more convenient for our purposes we will work in continuous time. Therefore we use the continuous time version of the replicator equations. Equation 1

represents this basic form of RD. We suppose there is a single population (considered infinite) of strategies and we consider for simplicity two-player games. Let $A = (a_{ij})_{i,j=1}^{n}$ be the reward matrix (a_{ij} is the reward for the joint strategy (i, j), and n is the total number of possible strategies).

$$\frac{dx_i}{dt} = [(Ax)_i - x \cdot Ax]\, x_i \tag{1}$$

The state x of the population can be described as a probability vector $x = (x_1, x_2, ..., x_n)$ which expresses the different densities of all the different types of replicators (i.e., strategies) in the population, with x_i representing the density of replicator i. As mentioned above, A is the payoff matrix that describes the different payoff values that each individual replicator receives when interacting with other replicators in the population. Hence $(Ax)_i$ is the payoff that replicator i receives in a population with state x, whereas $x \cdot Ax$ describes the average payoff in the population. The growth rate $\frac{dx_i}{dt}/x_i$ of replicator i in the population, equals the difference between the replicator's current payoff and the average payoff in the population. For a more detailed elaboration, we refer to [4, 7, 22].

Usually, we are interested in models of multiple players that evolve and learn concurrently, and therefore in that case we need to consider multiple populations. For ease of exposition, the discussion focuses on only two such learning players. As a result, we need two systems of differential equations, one for each player. This setup corresponds to an RD for asymmetric games, where A and B are the payoff tables for respectively the first and second player, and the available replicators of the players belong to two different populations, respectively p and q. This translates into the following coupled replicator equations for the two populations:

$$\frac{dp_i}{dt} = [(Aq)_i - p \cdot Aq]p_i \tag{2}$$

$$\frac{dq_i}{dt} = [(Bp)_i - q \cdot Bp]q_i \tag{3}$$

Equations 2 and 3 indicate that the growth rate of the types in each population is additionally determined by the composition of the other population, in contrast to the single population (learner) case described by Equation 1. If $A = B^T$ equation 1 would emerge again.

3.1.2 Replicator Dynamics: Selection and Mutation

It is known from game theoretic studies that when we consider human players in a game they usually do not purely select their actions [1] greedily [4]. Once in a while they also randomly explore their possible actions. This closely resembles the theory of Reinforcement Learning where players have to make a trade off between exploration and exploitation [15].

[1] Note that throughout this Section we use the word 'action', as is common in Reinforcement Learning, but in the current study actions represent heuristic poker strategies

In this section we describe the RD model of Q-learning. These equations are derived by constructing a continuous time limit of the Q-learning model, where Q-values are interpreted as Boltzmann probabilities for the action selection. Q-learning is an adaptive value iteration method [15, 21], which bootstraps its estimate for the state-action value $Q_{t+1}(s,a)$ at time $t+1$ upon its estimate for $Q_t(s^{'},a^{'})$ with $s^{'}$ the state where the learner arrives after taking action a in state s:

$$Q_{t+1}(s,a) \leftarrow (1-\alpha)Q_t(s,a) + \alpha(r + \gamma\ max_{a'}Q_t(s^{'},a^{'})) \tag{4}$$

With α the usual step size parameter, γ a discount factor and r the immediate reinforcement.

Action selection in Q-learning is usually based on a stochastic process. Popular choices include ϵ-greedy exploration (select the best action with probability 1-ϵ , or a random action otherwise) and Boltzmann exploration (the selection process further depends on a temperature parameter). Here we assume that agents choose actions by using the Boltzmann selection: an action a_j is chosen with probability

$$x_j = \frac{e^{\frac{Q(s,a_j)}{\tau}}}{\sum_i e^{\frac{Q(s,a_i)}{\tau}}} \tag{5}$$

where τ is a temperature parameter used to balance exploration and exploitation (the agent tends to select actions associated with higher utilities when τ is low). [2]

We again consider games between 2 learning players. The state s can be safely removed from the update rule as we only consider stateless games here. Deriving the equations for Q-learning goes as follows: a difference equation is derived for the Boltzmann probabilities $x_i(k)$ with k the current timestep; next we suppose that the amount of time between two repetitions of the game is given by δ with $0 < \delta \leq 1$. The variable $x_i(k\delta)$ describes the x-values at time $k\delta = t$. Calculating a continuous time limit of these equations, for $\delta \to 0$, leads to the following equations for the first player,

$$\frac{dp_i}{dt} = \frac{\alpha}{\tau}[(Aq)_i - p \cdot Aq]p_i + p_i\alpha\sum_j p_j ln(\frac{p_j}{p_i}) \tag{6}$$

analogously for the second player, we have,

$$\frac{dq_i}{dt} = \frac{\alpha}{\tau}[(Bp)_i - q \cdot Bp]q_i + q_i\alpha\sum_j q_j ln(\frac{q_j}{q_i}) \tag{7}$$

Equations 6 and 7 express the dynamics of both Q-learners in terms of Boltzmann probabilities.

Comparing Equations 6 or 7 with the RD in Equation 1, we see that the first term of (6) or (7) is exactly the same and thus takes care of the selection

[2] Note that in the literature τ occurs as well in the nominator as in the denominator.

mechanism (see [22]). The mutation mechanism for Q-learning is therefore left in the second term, and can be rewritten as:

$$x_i \alpha \sum_j x_j ln(x_j) - ln(x_i) \tag{8}$$

In equation (8) we recognize 2 entropy terms, one over the entire probability distribution x, and one over strategy x_i.

Relating entropy and mutation is not new. It is well known [11, 14] that mutation increases entropy. In [14], it is elucidated that the concepts are familiar with thermodynamics in the following sense: the selection mechanism is analogous to *energy* and mutation to *entropy*. So generally speaking, mutations tend to increase entropy. Therefore our second term is a measure of the mutations in strategy space occuring in the game under consideration. Hence we have a selection-mutation perspective on Q-learning. Exploration from RL then naturally maps to the mutation concept, as both concepts take care of providing variety. Analogously selection maps to the greedy concept of exploitation in RL.

3.2 The Heuristic Payoff Table

The RD equations take as input a payoff matrix (e.g., matrix A in Equation 1) that assigns a reward to each joint action. For a complex game such as No-Limit Poker it is unpractical to assemble all game actions into a normal form game (NGF) matrix, simply because it then has too many dimensions. Therefore, we look at heuristic strategies as outlined in Section 2.2. A heuristic payoff table replaces the NFG matrix, and gives the payoffs for all possible strategies given some known opponent strategies.

Let's assume we have n players and k strategies. This would require k^n entries in our heuristic payoff table. We now make a few simplifications, i.e., we do not consider different types of players, we assume all players can choose from the same strategy set and all players receive the same payoff for being in the same situation. This setting corresponds to the setting of a symmetric game. This means we consider a game where the payoffs for playing a particular strategy depend only on the other strategies employed by the other players, but not on who is playing them.

Now the distribution of n players on k pure strategies is a combination with repetition, hence a heuristic payoff table requires $\binom{n+k-1}{n}$ rows. Each row yields a *discrete profile* $S = (S_1, \ldots, S_k)$ telling exactly how many players play each strategy.

Suppose we have 3 heuristic strategies and 6 players, this leads to a heuristic payoff table of 28 entries, which is a serious reduction from $3^6 = 729$ entries in the general case. Table 1 illustrates what the heuristic payoff table looks like for three strategies S_1, S_2 and S_3. The left-hand side expresses the discrete profile, while the right-hand side gives the payoffs for playing any of the strategies given the discrete profile.

$$P = \left(\begin{array}{ccc|ccc} S_1 & S_2 & S_3 & P_1 & P_2 & P_3 \\ \hline 6 & 0 & 0 & 0 & 0 & 0 \\ & \ldots & & & \ldots & \\ 4 & 0 & 2 & -0.5 & 0 & 1 \\ & \ldots & & & \ldots & \\ 0 & 0 & 6 & 0 & 0 & 0 \end{array}\right)$$

Table 1: An example of a heuristic payoff table

Consider for instance the second row of this table: in this profile there are 4 players that play strategy S_1, none of the players play strategy S_2 and 2 players play strategy S_3. Furthermore, -0.5 is the expected payoff for playing strategy S_1 given these set of opponent strategies (i.e., given this discrete profile). Obviously, when a strategy is not employed by any player, no payoffs are recorded and the resulting expected payoff is then 0. For situations when all players in a discrete profile play identical strategies, the expected payoff is also 0 because no payoffs are made against other strategies. Because poker is zero-sum, the profits and losses are actually divided between the same class of players playing this particular strategy, and the average result (for this strategy) is 0. Therefore, for game of poker, the heuristic payoff table expresses the utility of a strategy in the presence of different opponent strategies.

To determine the payoffs in the table, we compute expected payoffs for each discrete profile from real-world poker data. More precisely, we look in the data for the appearance of each discrete profile and compute from these data points the expected payoff for the used strategies. However, because payoff in the game of poker is non-deterministic, we need a significant number of independent games to be able to compute representative values for our table entries. In Section 4 we provide more details on the data and on the process of computing the heuristic payoff table.

3.3 Simplex Analysis

Using the heuristic payoff table as input to the RD equations, we can now analyze the dynamics of strategies changing. The dynamics can be visualized in a simplex analysis that allows us to graphically and analytically study the dynamics of the system.

Before explaining this analysis, we first introduce a definition of a simplex. Given n elements which are randomly chosen with probabilities $(x_1, x_2, \ldots, x_n)$, there holds $x_1, x_2, \ldots, x_n \geq 0$ and $\sum_{i=1}^{n} x_i = 1$. We denote the set of all such probability distributions over n elements as Σ_n. Σ_n is a $n-1$-dimensional structure and is called a *simplex*. One degree of freedom is lost due to the normality constraint. For example in Figure 1, Σ_2 and Σ_3 are shown. In the figures throughout the experiments we mainly use Σ_3, projected as an equilateral triangle as in Figure 1(b), but we drop the axes and labels.

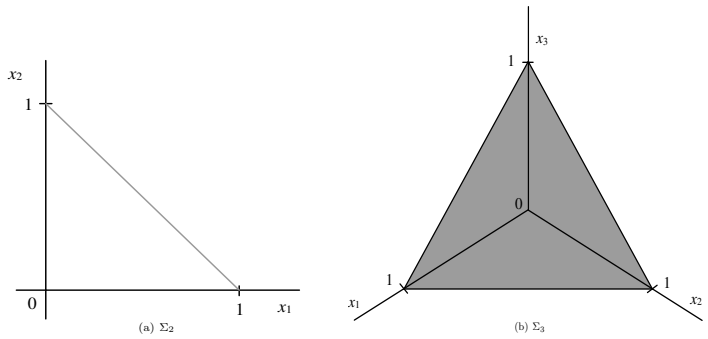

Figure 1: The unit simplices Σ_2 (a; left) and Σ_3 (b; right).

The simplex plots show arrows (as for example shown in Figure 2) or trajectories (as for example shown in Figure 4) that indicate the direction of change of the strategies. To calculate the direction at any point $s = (x, y, z)$ in our simplex, we consider a large number of N runs with mixed-strategy s; x is the percentage of the population playing strategy S_1, y is the percentage playing strategy S_2 and z is is the percentage playing strategy S_3. For each run, each player selects their pure strategy based on this mixed-strategy (i.e., pure strategies are sampled based on the probability distribution of s). Given the number of players using the different pure strategies (S_1, S_2, S_3), we have a particular discrete profile for each run. This discrete profile can be looked up in our heuristic payoff table, yielding a specific payoff for each strategy. The average of the payoffs of each of these N discrete profiles gives the payoffs at $s = (x, y, z)$. Provided with these payoffs we can easily compute the RD by filling in the values of the different variables. This yields us a gradient or direction at the point $s = (x, y, z)$.

Starting from a particular point within the simplex, we can now generate a smooth trajectory (consisting of a piecewise linear curve) by moving a small distance in the calculated direction, until the trajectory reaches an equilibrium. A trajectory does not necessarily settle at a fixed point. An equilibrium to which trajectories converge and settle is known as an attractor, while a saddle point is an unstable equilibrium at which trajectories do not settle. Attractors and saddle points are very useful measures of how likely it is that a population converges to a specific equilibrium. Each attractor consumes a certain amount of the strategy space that eventually converges to it. This space is also called the basin of attraction [6].

4 Experiments and results

We collected a total of 318535 No-Limit Texas Hold'em games played by a total of 20441 human players at an online poker site. In our data we have a variable

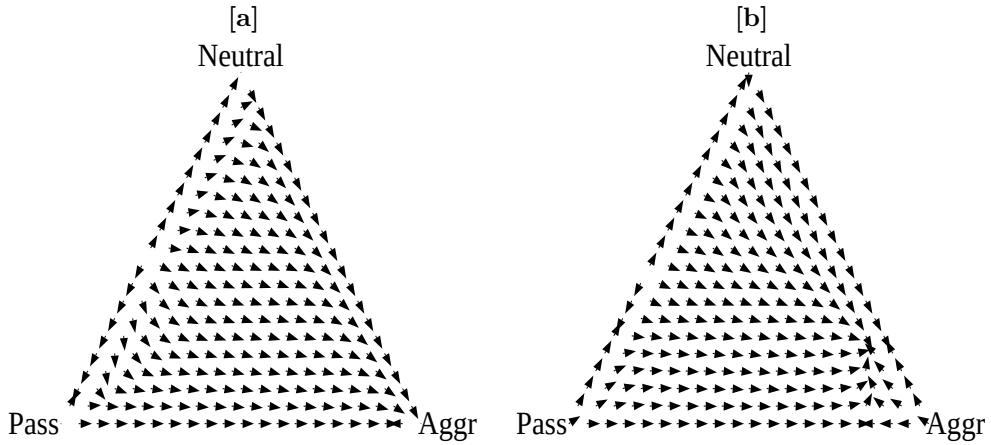

Figure 2: RD plots analyzing post-flop strategies using the replicator dynamics based on selection (a) and selection combined with mutation (b)

number of players participating in a single game, ranging from 2 player games to full-table games with 9 players. As a first step we needed to determine the strategy for a player for any given game. If a player played less than 100 games in total, we argue that we do not have sufficient data to establish a strategy, and therefore we ignore this player (and game). If the player played at least 100 games, we use intervals of 100 games to collect statistics for this specific player, and then determine the $VPIP$ and AGR values (see Section 2.2). Based on these computed values, we are then able to label the player with a strategy. The resulting strategy was then associated with the specific player for all games in the interval. Having estimated all players' strategies, it is now possible to determine the discrete profile (i.e., the number of players playing any of the available strategies) for all games. Finally, we can compute the average payoffs for all strategies given a particular discrete profile.

We will apply the RD based on selection (see Section 3.1.1), which will favor the most fit strategy in the population. To further sharpen our analysis with a more elaborate human like model, we also apply the RD model derived from the popular Q-learning algorithm (see Section 3.1.2) that contains the sum of the basic RD equations (selection) and additionally an entropy term mapping to mutation. Using the RD with this selection-mutation model we facilitate explorative behavior instead of pure greedy behavior. For all described selection-mutation experiments, we have chosen a fixed temperature τ of 0.1. We will now highlight several experiments with varying strategy classifications.

4.1 Analyzing pre-flop and post-flop Play

For our first two experiments we analyze pre-flop and post-flop play in isolation. To be more specific, we label players with strategies based solely on either the

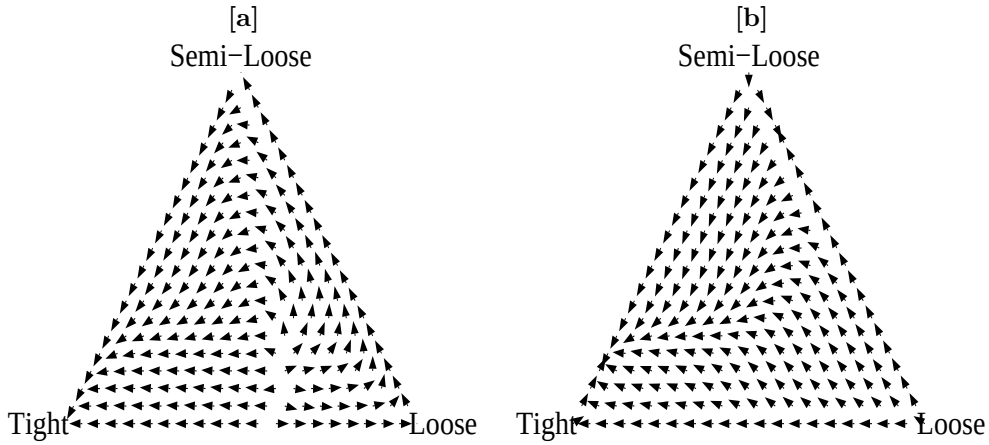

Figure 3: RD plots analyzing pre-flop strategies using the replicator dynamics based on selection (a) and selection combined with mutation (b)

VPIP or *AGR* values that were computed. Table 2 gives the rules for the strategy classification. These rules were derived from domain knowledge and are common for classifying strategies in a No-Limit Texas Hold'em game (see e.g., [3, 5, 12]).

The *VPIP* determines the pre-flop strategy, and gives insight in the player's card selection. A loose player plays a wider range of cards whereas a tight player will wait for more quality cards (i.e., those that have a higher probability of winning the game at showdown when cards are compared). The *AGR* value determines the post-flop strategy, and denotes the ratio between aggressive (i.e., betting and raising) and passive (i.e., calling) actions.

It is often claimed by domain experts that aggressive strategies dominate their passive counterparts. The rules of the poker game, and in particular the fact that games can be won by aggressive actions even when holding inferior cards, seem to backup this claim. In Figure 2a (selection) we can see one strong attractor that lies at the pure strategy AGGRESSIVE. Figure 2b (selection-mutation) shows a mixed equilibrium strategy mainly between AGGRESSIVE and NEUTRAL. Again, the AGGRESSIVE strategy is played 3 out of 4 games. These results nicely confirm the claim that aggressive strategies dominate passive ones.

For the pre-flop strategy, the tight strategy is often assumed to be best,

pre-flop-strategy	Rule	post-flop-strategy	Rule
Tight	$VPIP < 0.25$	Passive	$AGR < 1$
Semi-Loose	$0.25 <= VPIP < 0.35$	Neutral	$1 <= AGR < 2$
Loose	$VPIP >= 0.35$	Aggressive	$AGR >= 2$

Table 2: Strategy classification for pre-flop and post-flop play

in particular for less skillful players. Although it is also claimed that the pre-flop strategy should depend on the strategies played by the opponents. If the majority of players play a tight strategy, then a looser strategy pays of and vice versa.

In Figure 3a (selection) we see an attractor lying in the pure strategy TIGHT. When we apply the selection-mutation model in 3b, we find a mixed strategy between TIGHT and SEMI-LOOSE. Still, the TIGHT strategy is dominant and is played 8 out of 10 games. These findings seem to contradict the claim that one should mix their pre-flop play according to the opponent strategies. However, we need to take into account that we are currently ignoring the post-flop strategy. We already showed in our previous experiment that aggression is important for the utility of the overall strategy. So one could say that on average the TIGHT strategy is optimal, given a random post-flop strategy. Mixing up pre-flop play may only be reasonable when always playing aggressive after the flop.

4.2 Analyzing Complete Poker Strategies

For our next series of experiments, we combine both *VPIP* and *AGR* features for strategy classification. The rules used are shown in Table 3. Again note that these strategy classifications are derived from poker literature, although we reduced the number of attributes per feature to two so we have exactly four strategies, namely tight-passive (a.k.a. ROCK), tight-aggressive (a.k.a. SHARK), loose-passive (a.k.a. FISH) and loose-aggressive (a.k.a. GAMBLER).

Experts argue that the SHARK strategy is the most profitable strategy, since it combines patience (waiting for quality cards) with aggression after the flop, while the FISH strategy is considered as the worst possible strategy.

Recall from Section 3.3 that our simplexes show the dynamic behavior of the participating players having a choice from three strategies. For this experiment we actually have a total of four strategies. We exclude in this case one strategy per plot, by only considering discrete profiles in our heuristic payoff table where we have no players playing the excluded strategy. This leaves us with four different combinations of three strategies. We plotted the results with the RD for both the selection and selection-mutation model.

What we can see from plots in Figure 4a, Figure 5a and Figure 7a, is that both passive strategies, i.e., the FISH and ROCK strategies, are dominated by the two aggressive strategies SHARK and GAMBLER.

We also see that the attractors in Figure 4a and Figure 5a lie close to the

Strategy	Rule
Rock	$VPIP < 0.25$, Passive $AGR < 2$
Shark	$VPIP < 0.25$, Passive $AGR >= 2$
Fish	$VPIP >= 0.25$, Passive $AGR < 2$
Gambler	$VPIP >= 0.25$, Passive $AGR >= 2$

Table 3: Simple strategy classification

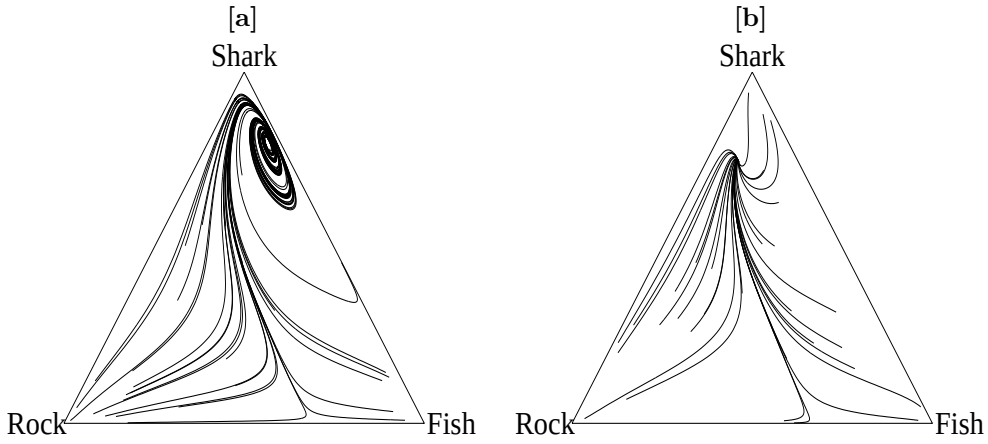

Figure 4: Trajectory plots analyzing the Rock, Shark and Fish strategies using the RD based on selection (a) and selection-mutation (b)

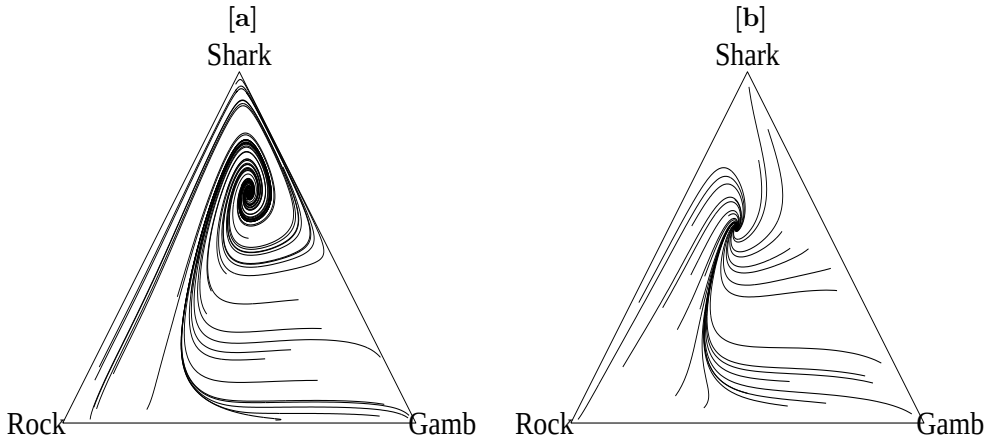

Figure 5: Trajectory plots analyzing the Rock, Shark and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

SHARK strategy, namely this strategy is played respectively around 80% and 65% of the times. In Figure 7a the GAMBLER strategy is slightly preferred over the SHARK strategy, that is played 40% of the times. Based on our analysis we can say that SHARK is a strong strategy, as was suggested by domain experts. Only in Figure 7 is SHARK slightly dominated by GAMBLER. It is also obvious from the plots that the FISH strategy is a repeller, with the exception of Figure 6, where the equilibrium is actually a mix with the ROCK strategy.

For the selection-mutation plots we see similar results with mixed strategies close to the SHARK. In general, the equilibria found through selection-mutation

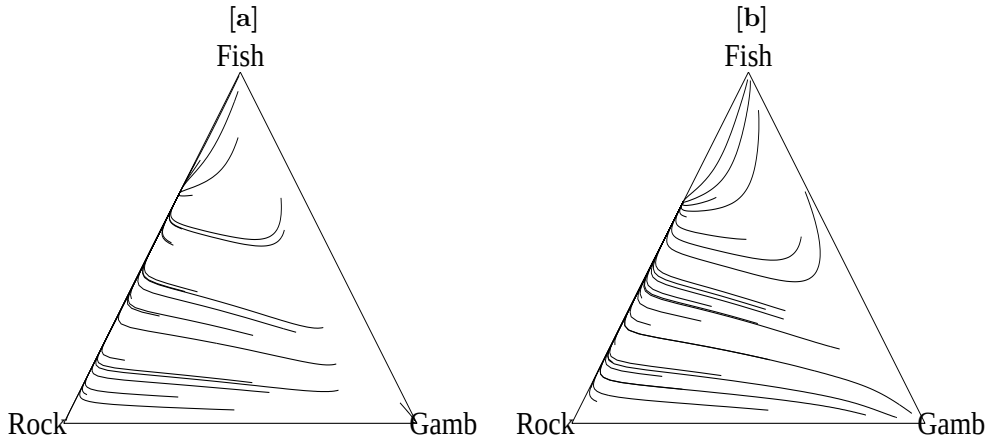

Figure 6: Trajectory plots analyzing the Rock, Fish and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

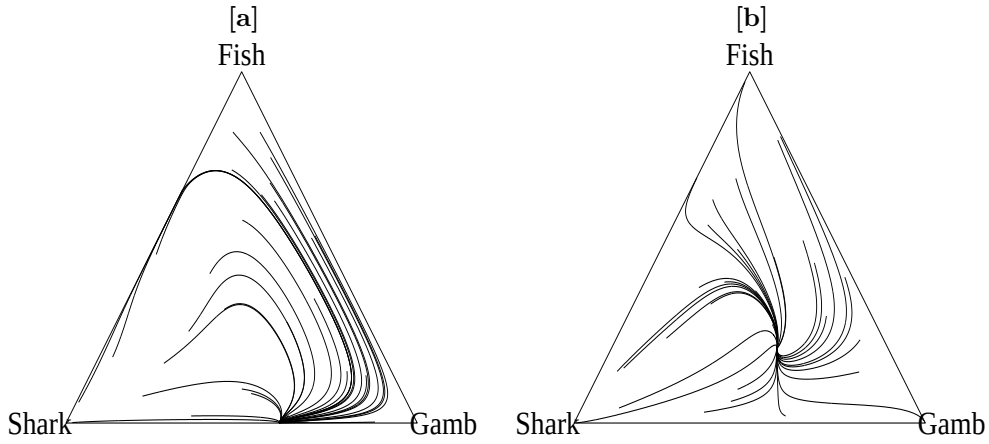

Figure 7: Trajectory plots analyzing the Shark, Fish and Gambler strategies using the RD based on selection (a) and selection-mutation (b)

lie closer to the center of the simplex and therefore mixes more between the available strategies. This comes as no surprise, when we apply the selection-mutation model a player will explore his available actions more, which pulls the dynamics more to the center of the simplex but also allows to find more optimal solutions. An interesting observation in Figure 4 is that for the mixed strategy using the selection model the Fish strategy is played more compared to the the Rock strategy (respectively 17% to 3%), while for the selection-mutation model we see the opposite. Now the Rock strategy is played more with 17% to 10%. Since domain experts believe the Fish strategy is inferior over all other

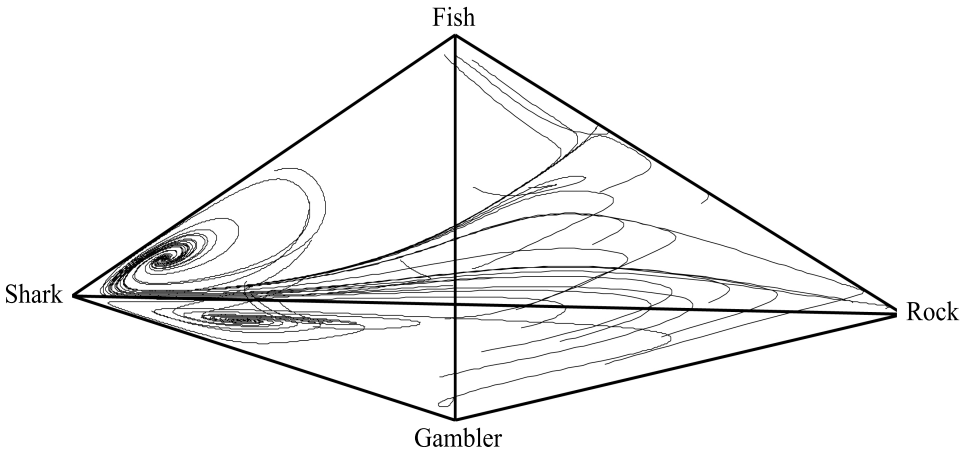

Figure 8: Trajectory plots in 3-dimensional space analyzing dynamics for all 4 strategies

strategies, the results from the selection-mutation model seem to approximate reality better.

A shortcoming of the leave-one-out approach, is that we always dismiss large portions of the data because we put constraints on the strategy that must be excluded. We also analyzed the dynamics among all four strategies at once. The result (2-dimensional snapshot in 3-dimensional space) is represented in Figure 8 (for the selection model). The dynamics are similar to our previous plots, but there are differences. As for example, we can now see that only two attractors remain both near the SHARK strategy, and that the attractor found in Figure 7a is actually lost. The attractors near the SHARK strategy clearly have a stronger basin of attraction, i.e. trajectories are more likely to end up in one of these equilibria.

For the results including mutation we only report the results here. We see one attractor, namely near the mixed strategy $56\%, 25\%, 17\%$ and 2%, for respectively the SHARK, ROCK, GAMBLER and FISH strategy. The FISH strategy nearly went extinct.

5 Conclusion

In this paper we investigated the evolutionary dynamics of strategic behaviour of players in the game of No-Limit Texas Hold'em poker. We performed this study from an evolutionary game theoretic perspective using two Replicator Dynamic models, one that is purely driven by selection, and another that also contains mutation. Using these models we analyzed the dynamic properties by studying how rational players switch between different strategies under different circumstances. For our analysis we observed poker games played at an online

poker site and used this as our data. Based on domain knowledge, we identified several strategies (with varying levels of detail) in the game of poker. We then computed the heuristic payoff table to which we applied the Replicator Dynamic models. The results are visualized in simplex plots, these show where the equilibria lie, what the basins of attraction of the equilibria look like, and what the stability properties of the attractors are. Our results confirm that what is claimed by domain experts, namely that usually aggressive strategies dominate their passive counterparts. We also noticed that when we apply an RD model that includes mutation to the data, we do see results that better reflect what domain experts claim, compared to results obtained with the basic model of selection.

For future work, we will examine the interactions between the strategies among several other dimensions. For example, we could look at more detailed strategy classifications (i.e., based on more features) or represent strategies in a continuous way. Although our evolutionary game theoretic approach to the game drops the hyper-rationality assumption of players, and produces a more human-like model of the dynamics involved, currently players are still assumed to be rational in the sense that they aim to optimize their payoffs and are also capable of doing so (i.e., they have full information on expected payoffs for playing any of the available strategies). In many human domains, and certainly in poker, this rationality assumption does not necessarily hold. Therefore, we are interested in applying other behavioral models that might describe more accurately the behavior of players in the game of poker.

References

[1] D. Billings, N. Burch, A. Davidson, R. C. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *IJCAI*, pages 661–668, 2003.

[2] A. Davidson, D. Billings, J. Schaeffer, and D. Szafron. Improved opponent modeling in poker. In *Proceedings of The 2000 International Conference on Artificial Intelligence (ICAI'2000)*, pages 1467–1473, 2000.

[3] D. Doyle Brunson. *Doyle Brunson's Super System: A Course in Power Poker*. Cardoza, 1979.

[4] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2001.

[5] D. Harrington. *Harrington on Hold'em Expert Strategy for No Limit Tournaments*. Two Plus Two Publisher, 2004.

[6] M. W. Hirsch, S. Smale, and R. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press, 2004.

[7] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[8] J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[9] S. Phelps, S. Parsons, and P. McBurney. Automated trading agents versus virtual humans: an evolutionary game-theoretic comparison of two double-auction market designs. In *Proceedings of the 6th Workshop on Agent-Mediated Electronic Commerce*, New York, NY, 2004.

[10] M. Ponsen, J. Ramon, T. Croonenborghs, K. Driessens, and K. Tuyls. Bayes-relational learning of opponent models from incomplete information in no-limit poker. In *Twenty-third Conference of the Association for the Advancement of Artificial Intelligence (AAAI-08)*, pages 1485–1487, Chicago, USA, 2008.

[11] T. Schneider. Evolution of biological information. *journal of NAR*, 28: 2794–2799, 2000.

[12] D. Slansky. *The Theory of Poker*. Two Plus Two Publisher, 1987.

[13] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI '05)*, pages 550–558, 2005.

[14] D. Stauffer. Life, love and death: Models of biological reproduction and aging. *Institute for Theoretical physics, Köln, Euroland*, 1999.

[15] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[16] P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Math. Biosci.*, 40:145–156, 1978.

[17] K. Tuyls, K. Verbeeck, and T. Lenaerts. A Selection-Mutation model for Q-learning in Multi-Agent Systems. In *The second International Joint Conference on Autonomous Agents and Multi-Agent Systems. ACM Press, Melbourne, Australia*, 2003.

[18] K. Tuyls, P. 't Hoen, and B. Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *The Journal of Autonomous Agents and Multi-Agent Systems*, 12:115–153, 2006.

[19] P. Vytelingum, D. Cliff, and N. R. Jennings. Analysing buyers and sellers strategic interactions in marketplaces: an evolutionary game theoretic approach. In *Proc. 9th Int. Workshop on Agent-Mediated Electronic Commerce*, Hawaii, USA, 2007.

[20] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In P. Gymtrasiwicz and S. Parsons, editors, *Proceedings of the 4th Workshop on Game Theoretic and Decision Theoretic Agents*, 2001.

[21] C. Watkins. *Learning with Delayed Rewards.* PhD thesis, Cambridge University, 1989.

[22] J. W. Weibull. *Evolutionary Game Theory.* MIT Press, 1996.

[23] E. Zeeman. Dynamics of the evolution of animal conflicts. *Journal of Theoretical Biology*, 89:249–270, 1981.