Intelligent Poker Player

And an Analysis of its Strength and Success Against Various Styles of Play

Michael Andrusyna
andrusyn@seas.upenn.edu
Bension Dworkin
bension@seas.upenn.edu
Advisor: Prof. Michael Kearns

Abstract

In the past few years, poker has become very popular in the US as well as abroad. In particular, Texas Hold'em has been popularized through its now numerous television programs and gambling sites on the Internet. Texas Hold'em is of interest because it is a game of imperfect information where playing your opponent is just as important as playing your cards. Strategies become very complex and involve analyzing the current state of the game and learning from the past play of opponents. We created a client-server no-limit Texas Hold'em poker game where human and AI players can play against each other. We created several AI players that use various strategies described below. The AI players were designed to play well against human players, but for practical reasons, we evaluated our different players based on their success playing one on one (heads up) against other AI players with various styles of play. Our original goal was to make an AI player that does well in most situations. We did not expect our player to be unbeatable but we wanted the player to show some signs of intelligence. Our results show that certain strategies are more successful in heads up play but we did not analyze their success in a multi-player environment.

Introduction

In the past few years, poker has become very popular in the US and all around the world. In particular, Texas Hold'em has been popularized through its now numerous television programs and gambling sites on the Internet. Just last year it was estimated that about 30 million Americans play this intriguing game of chance and skill. Today, it is estimated that at least double that amount of Americans play Texas Hold'em.

Why is this game so exciting and popular? One of the reasons is that poker is a game of chance as well as skill. To be successful at it, you need to be able to correctly calculate how strong your hand is, bluff when your hand is not so strong, as well as be able to read your opponent and figure out how strong his hand really is. For many, this set of criteria provides a captivating challenge that keeps people interested. In addition, a successful poker player must react to different opponents' strategies. A player can play very loose, very tight, or somewhere in between. He may also play very passively, very aggressively, or again somewhere in between. All of this adds up to a pretty complex sequence of decisions and actions that everyone must consider. However, the final ingredient to poker, luck, makes it all worthwhile especially when things are not going your way.

While the strategy within the game can be complex, the rules to Texas Hold'em are fairly straightforward. Everyone is first dealt two cards face-down. After this, a round of betting occurs and players decide whether or not their cards are worthy enough to play with. During betting, you can check, bet, raise, or fold. Checking means you essentially do nothing and you let the next person decide what they want to do. To end a round of betting, all bets must be called. Those players who do not want to call a bet must fold. After the round of betting, three community cards called the flop are placed in the center of the table face-up. Everyone may use these cards to improve their hand. Then another round of betting occurs. After this, another community card called the turn is placed on the table and another round of betting occurs. Finally, one more community card called the river is placed on the table. One final round of betting then occurs. In the end, players use the five community cards and their two down cards to make their best 5-card poker hand. The player with the best 5-card poker hand wins the pot.

Weakest	High Card
|	One Pair
|	Two Pair
|	Three of a kind
|	Straight
|	Flush
|	Full House
\/	Four of a Kind
Strongest	Straight Flush

Figure 1: Hand Rankings

Our motivation for this project

As avid amateur poker players, this is a topic that interests both of us. Often during play, one wonders what the correct decision is. It would be helpful if an automated program could analyze the hand and give you advice. Playing poker involves a lot of mental calculations that players estimate during play. Many decisions are based on objective criteria such as odds of improving one's hand, odds of holding the best hand, pot odds, etc. A computer program could automate these calculations and give a player better information so he can make an informed decision. When you watch poker on TV, they give the odds of someone winning, but the players obviously don't know that information – they have to calculate it on the fly. Additionally, a computer program could be helpful in analyzing the aggregate decisions of other players. A good player will pay close attention to opponents' styles of play and adjust their own play to take advantage of opponents' weaknesses. Although modeling opponents' behavior is more subjective, a player still needs to make decisions based on aggregate data from past hands. An automated tool could be helpful in making this analysis. When playing against multiple opponents it can be hard to remember a hand played an hour ago but a computer program can help give you more information so you can make that decision. Our goal for the project was to create an AI player that will be able to make independent decisions and play against real players and/or other AI players. Hopefully the AI's winning style of play will give us insight to more accurately analyze the game and make us better players.

Different styles of play

As mentioned earlier, there are many different styles that a player can have in poker. Phil Hellmuth and other poker authors describe these styles in great depth. Although coding these various styles into an AI player was challenging, there are definite characteristics that each style possesses. Once again, examples of different styles are tight, loose, passive, and aggressive. A tight player is one that only plays limited amounts and types of hands. These hands tend to be the best starting hands possible like A,A (2 aces), K,K (2 kings), Q,Q (2 queens), A,K (ace, king), etc. As a result, tight players only put their money into the pot when they have the goods. They tend to play for and win fewer pots than other players but have a higher percentage of pots won in hands they participate in. A loose player, on the other hand, is exactly the opposite. They tend to play a vast number of hands. It does not matter to them whether the hand is very good, mediocre, or even horrendous. Since loose players play in a lot more pots than tight players, they also tend to win more pots. However, loose players also lose a lot more pots.

There are further styles that apply to both tight and loose players. These styles include playing passively and playing aggressively. Playing passively means that whether you have good cards or not, you play your hand passively or slowly, that is, you do not make very big bets. This tends to bring more money into the pot because more players may call your bet since it is not that large. However, this also gives more players a chance to improve their hand with future community cards for a cheap price. Conversely, a player may want to play aggressively. Instead of making small bets, these players tend to make larger bets. The advantage to this is that you drive out people who currently have mediocre or poor hands that may have a good chance of improving them later. In the end, you have a better chance of winning the pot because less people are involved. However, the pot may not grow as large unless you have other opponents who also have strong hands worthy of calling your large bet. Aggressive players also tend to raise and reraise opponents frequently.

Project Description

We created a poker program that allows multiple players – real or automated – to connect to a Texas Hold‘em poker server. Within this playing environment is a regular poker game for real players – no different than any commercial poker game you find online or in a casino. The only difference is that one can insert one or more automated players to compete against real people. We do not expect our automated player to always win. As can be seen in the World Series of Poker, luck is a factor in the game – players can get improbable hands and beat even the best poker players in the world. We expected that our artificial player would play well in many situations, but eventually someone would be able to find a strategy to beat it. One of the great things about Texas Hold’em is that it is a game of imperfect information. Unless you know your opponent’s hand, previously encountered situations may arise where the choice made last time may not be the best choice for this occasion. The best strategies are often ones that are not predictable. Although we know we didn’t create the perfect player, it excels in certain situations.

We wrote our program in Java using the built in socket class to handle the client-server part of the game. Our program is a text-based game so any computer with a java compiler and java interpreter can run it. We wanted to add graphical features, but ran out of time. Initially, we wanted to create a behavioral model of all the opponents so that we could try to predict their play and take advantage of their predicted strategy. We implemented the first part of this by keeping track of certain statistics such as the percentage of hands that were played, how large the player’s bets were compared to the size of the pot, how often did the player raise and reraise an opponent, and what percentage of hands the player actually won. We started to incorporate these statistics into our strategy but after some suggestions from our advisor, we decided to focus on other parts of the project so that we could ultimately evaluate our AI player better.

Rather than place opponents into different styles of play, we hard coded various strategies of AI Players and evaluated them based on their heads up performance. The advantage of this approach is that it is easier to collect data. We ran numerous simulations that ran up to 500 hands in no more than a few minutes. With real players this could take a few hours. Even if we had the time and volunteers to participate, we wouldn't know if real players would keep a consistent strategy, especially if they aren't playing for real money. We wanted to create a controlled environment so we could make minor changes and then replay the same situation.

We created 6 different players largely based on strategies described in Phil Hellmuth's book *Play Poker Like the Pros* as well as books by other poker authors like David Sklansky. The 4 basic strategies are tight, loose, aggressive, and passive. We created 2 additional players which combined the basic strategies: tight/aggressive and loose/aggressive. We didn't try other pairings of styles because of time constraints and because we wanted to limit the scope of the project.

Related Work and Resources Required

The previously mentioned styles of play are discussed in numerous books and papers that have been written about poker. One such book, *Play Poker Like the Pros*, was written by Phil Hellmuth, a very experienced professional poker player. In it he talks about strategies that people use in poker. Hellmuth associates each strategy with a different kind of animal. He equates different styles of play with different animals such as a mouse, jackal, lion, and elephant. Hellmuth then goes on in his book, offering up many different situations you can find yourself in when you are playing a game of poker. He also analyzes how each type of player mentioned above would play the hand and also how you should play against each type of player in that situation. We used the information in this book to help model our strategy for our AI player. We also used a similar book, *Hold'em Poker for Advanced Players*, by David Sklansky and Mason Malmuth. This book goes even more in-depth into poker strategy in various scenarios.

We consulted a few papers that have been written on poker. Specifically, we looked at papers by fellow computer scientists that discuss opponent modeling in poker. There are, however, differences between the research that we are studying in these papers and our own project. The papers in question mainly deal with Limit Texas Hold'em, in which players are limited to how much they can bet at each stage of the hand. Our project deals with No-Limit Texas Hold'em, which does not limit the player to how much he can bet. This difference is not as subtle as one may think. Although the game, at a higher level, is played the same way, the fact that the bets are not limited in our situation makes a big difference in the way that the strategies unfold within the game. As a result, our implementation of our AI player is different than the implementations in the projects discussed in the aforementioned research papers.

Further research and analysis of poker has been done by the University of Alberta. The University of Alberta has a number of research groups dedicated to multiplayer game theory and AI research in those fields. One of these research groups, in fact, is about poker. This research group also has worked on a poker-playing program (for 10 years!) called the "Vex Bot" that has the ability to play in head-to-head matches against other poker players. Although it plays very successfully in head-to-head matches, it has not been extended to take into account a multiplayer environment where more than one player may be in the hand.

Discussion of the Challenges Encountered During the Fall Semester

As we acknowledged in our first paper for this project, there was a strong possibility that we may have been sidetracked by less important aspects of this project namely the coding of the client-server feature. We indeed did encounter this type of hindrance exactly where we predicted it. The reason for the extra time spent on the client-server portion of the program is because of the complexity of the synchronization aspects between the clients and the server. We needed to make sure that each client was receiving the correct information at the correct time.

Once we got everything synchronized the way we wanted it to be, we needed to make sure that the game play went the way it should go. This includes things like making sure the betting scheme is correct, making sure cards are dealt correctly, and making sure that winning hands are calculated correctly. Although these aspects of the game are easy to understand from our perspective, from the program's perspective it is a much different story. As we found out, it is much more difficult to implement these features within our program than it is to understand it for yourself.

Although we encountered the previously mentioned challenges during the fall semester, we felt confident in our progress as well as the potential for the future of the project. These challenges were expected and we felt that we sufficiently rose above them to allow us to move on to the more crucial aspects of the project, namely, the creation and evaluation of the artificial players that are at the center of our goals.

Potential Challenges for the Spring Semester

As discussed in the fall semester section, we felt that we would most likely encounter challenges within the modeling of the opponents for the AI. We predicted that these would probably comprise the brunt of our challenges that we would encounter in the spring semester. The criteria that we used to model the opponents' behavior is somewhat arbitrary and in the end may not be as accurate a predictor of future play as may be necessary for our AI player to be completely successful. Of course, we did expect our AI player to be successful in certain situations but maybe not all. At the beginning of the spring semester, we felt that we would have enough time during the spring semester to adjust our methods of behavior modeling as seemed fit. Although we knew that we would never be able to adjust them to obtain a perfect model, we adjusted them accordingly to attain as good a model as we could obtain.

Progress Made During the Fall Semester

We successfully created a server that allows clients to connect to play a game of Texas Hold'em. The client/server interaction is based on the use of sockets and synchronization. When the server starts, it listens for incoming socket connections on a given port. Once it receives a socket connection, it creates a thread called CasinoSession to handle interactions with the client. All communication between the server and client is handled by a unique CasinoSession thread through the socket that is created between them. Once enough players connect, the game is started. The number of players that determines whether or not to start the game is entered when the server is started up. Once the game has begun, it is played like any other game of Texas Hold'em. The attached readme file shows how to set everything up.

All information about the current state of the game – whose turn it is, how many chips everyone has, what cards everyone has, etc. is stored in a class called table. One essential aspect to making sure that the game is played correctly and fairly is synchronizing the information in table. There are two times when synchronization is needed – first when creating a game and then when updating the game, for example when cards are dealt out or when a player makes a decision. In the first situation, if there are not enough players then the server thread goes to sleep and is woken up when there are enough players. This needs to be synchronized because the threads are concurrently viewing and updating the variable number of players and need to make sure no one else is using the variable at the same time you are. The second situation of updating the game has two methods that need to be synchronized – update and view. This is similar to classic producer/consumer problems. This is especially important in poker because the game entails a specific order of actions – poker is a turn-based game. The game is set up so that one person needs to make a decision before the game can continue. The way we implemented this was to have a variable *hasseentable* which is reset to false on an update to the table and set to true on a view of the table. After a view occurs, the server thread will wait for a response if it is that player's turn. If it is not that player's turn, it will go to sleep and will be woken up when an update to the table occurs.

One of the more time-consuming parts of the game was evaluating each player's hand at the end of a hand. This seems trivial to a human player but translating it to code is a little more difficult. Our hand evaluator determines what hand everyone has (e.g. pair, full house, flush, etc.), information which is used by another method, winner, to determine who has the best hand at the end and who is therefore the winner of the hand. The hand evaluator code is partially reused by the AI player when it decides what to do. Along with evaluating its own hand it will also be able to check for possible hands that may come with future community cards. This is important in making a decision because a poor hand can become the best hand if the right cards come on the board. Conversely, a strong hand can become a second best hand, so betting strategy is partially based on this.

By the end of the fall semester, we were able to complete a skeleton AI player which made naive decisions. It collected data about other players but did not, initially, make full use of the data. For each opponent, an object called playerstats is created that keeps track of statistics of past decisions. The idea is to update an opponent's statistics every time he makes a decision so that the AI can have up-to-date information at its disposal when making its own decision. The AI player also evaluates its own hand strength. As mentioned earlier, the hand strength evaluation process will also include the usage of the hand evaluator method. By the end of the fall semester, we had coded up some of the more common strategies that are discussed in David Sklansky's book. More complicated strategies were incorporated later on. These strategies were obtained from books like Sklansky's as well as from our own personal experience at poker. Moreover, we gained our personal experience through game play as well as reading books by people like Sklansky and Hellmuth.

Progress Made and Challenges Faced During the Spring Semester

During the spring semester we planned on improving the artificial player. We added quantitative depth to the behavior modeling to make better use of the aggregate past actions. This information could be used to guess the opponent's strategy, for example do they play tight, loose, etc. Although our original plans were to focus specifically on the ways in which our player modeled its human counterparts' behaviors, our project took a slightly different course of action.

As discussed earlier, after discussions with our advisor about our progress, we realized that even if we managed to create this player that would be very intelligent in its entire decision-making process, we would not have enough time to scientifically evaluate our AI player's performance against human opponents. In reality, we could play hundreds of hands between our AI player and some human opponent, while keeping track of all of the AI's decisions during the game. We could then point out some "intelligent" aspects to the AI player's game-play. However, there is one problem with this approach, namely, that it is a very subjective approach. The AI player could in fact make some intelligent decisions which would be recorded but it would most likely make many more ill-advised decisions, which we would not really be looking for in our subjective approach to evaluation. This would lead to neither a scientific nor an accurate evaluation of our project. As a result, we took a more objective approach to the conclusion and evaluation of our project.

Instead of focusing our efforts on improving our AI player so that it plays the "best" that it can against human players, we decided to create a number of other AI players, each with its own style of play. Besides, how can you really tell whether the AI player is indeed playing "better" against human opponents with certain modifications and "improvements"? In our original proposal, we mentioned that we may add other AI styles of play, depending on how much time we had left. The shift in our focus provided us with this needed time. We were then able to create 6 new AI players. As mentioned, each of these possessed its own style of play, which included tight, loose, passive, aggressive, tight/aggressive, and loose/aggressive. With these new automated players, objective evaluation of our project would now be more feasible. We were now able to run simulations that could play thousands of hands in a matter of minutes, as opposed to only hundreds of hands played against a human opponent, which could take a few hours. Specifics about these simulations as well as our results obtained from them will be discussed in the following "Results" section of this paper.

Results

To obtain results so that we could evaluate the performances of all of our AI players, we ran simulations for all possible head-to-head combinations. As mentioned, each simulation involved 2 of the 6 AI players and ran for 10 rounds each. A round was over when one of the players lost all of his chips. We realized that our results could be more precise if we ran more than 10 rounds per simulation but time constraints did not allow us to do so. We were still able to draw conclusions from the substantial amount of data that we collected. A summary of all of the results of these simulations can be found in the table below (figure 2).

Match-up (p1 vs. p2)	Rounds won by p1	Rounds won by p2	Aggregate Avg. pot size	Hands Played	% Hands Won by p1	% Hands Won by p2
A vs. A	5	5	167.48	646	49.38%	51.08%
A vs. P	8	2	287.24	1218	53.45%	46.96%
L vs. A	5	5	132.03	873	49.94%	51.2%
L vs. L	5	5	132.0	1031	49.76%	51.12%
L vs. P	8	2	149.36	798	57.89%	42.98%
P vs. P	6	4	192.26	1112	50.09%	50.36%
T vs. A	4	6	115.15	978	47.96%	55.52%
T vs. L	4	6	145.32	651	45.32%	56.22%
T vs. P	6	4	120.1	1241	50.2%	50.28%
T vs. T	7	3	139.63	743	50.87%	49.13%
T vs. TA	5	5	133.26	1454	51.17%	48.97%
T vs. LA	6	4	452.47	375	41.07%	59.73%
TA vs. TA	5	5	101.06	1790	49.78%	50.45%
P vs. TA	5	5	115.19	1169	49.7%	50.56%
P vs. LA	4	6	303.92	1005	43.18%	58.41%
L vs. TA	6	4	149.41	586	56.48%	44.54%
L vs. LA	4	6	167.15	658	50.15%	50.76%
LA vs. TA	7	3	116.76	532	59.21%	40.79%
LA vs. LA	4	6	255.29	585	51.97%	49.06%
A vs. TA	4	6	182.61	1517	51.48%	48.98%
A vs. LA	8	2	149.64	707	42.86%	58.42%

Figure 2: Summary of Results from Head-to-Head Match-ups between the 6 different AI Players

A = Aggressive P = Passive L = Loose T = Tight
TA = Tight/Aggressive LA = Loose/Aggressive

We originally started off with only 4 AI styles of play – tight, loose, aggressive, and passive. After running all possible simulations between these 4 players, we found 2 players that stood out as being the most successful. These were the aggressive player and the loose player. As you can see from the results, both the aggressive and loose players dominated the passive player. Each won 8 out of 10 rounds in their simulations against the passive player and each won a more substantial amount of hands in the simulations.

We also examined how the tight AI player did against the passive one and we found that, if anything, the tight player was only a little better than the passive player, winning 6 out of 10 rounds and actually winning a little less hands than the passive player. You may notice that the percentages of hands won in any given match-up may not add up to 100% as one may expect. The reason for this is that sometimes in poker, you encounter a split pot situation. This means that more than one player had the same winning hand in the end and therefore more than one player won the hand. When this occurred during our simulations, we gave each player a win for that hand. This would then raise the total percentage of hands won to over 100%.

The performances of the aggressive and loose players against the remaining AI player, the tight player, were not as dramatic as those against the loose player but are still noteworthy. These performances were almost identical once again, with both the aggressive and loose player winning 6 of the 10 rounds against the tight player as well as winning many more hands. Once we saw these 2 styles of play stand out like they did, we started to analyze how well they did against each other. Amazingly, both the aggressive and loose players had very similar performances against each other. The match-up ended in a 5-5 tie in the amount of rounds won and each player also won about half the hands in the match-up.

After analyzing the performances of these 4 original AI players, we determined that the aggressive and loose players did the best. From this observation, we decided to create 2 new players to see how they fair against the 2 best so far, as well as against all of the AI players. We figured that if the aggressive and loose styles did best among the original 4, then maybe a combination of the two, loose/aggressive, would turn out to be an even better player. For comparative reasons, we also decided to create a tight/aggressive player to see how it would fair as well. Since we saw in our previous simulations that the passive player did not fair very well against all of the other opponents, we felt that any newly created styles comprised of a combination of the passive style and any other style would also not perform well. So we did not create any new players with a passive aspect to them.

Now focusing on our 2 new players, we once again ran simulations between them and all the other players. To our surprise, the loose/aggressive player did not do as well as was expected. Although it did win many more hands than all of its opponents, it did not dominate the tight and passive players like the original aggressive and loose players did. Also, it performed about the same against the loose player and was dominated in its match-up against the aggressive player, 8 to 2. The only match-up that the loose/aggressive player showed a substantially better performance in was against the new tight/aggressive player, where it won 7 out of 10 rounds. Oddly enough, the same tight/aggressive player that performed considerably worse than the loose/aggressive player, performed quite well against the aggressive player which the loose/aggressive player itself performed very poorly against. This formed a cyclical relationship between the performances of various AI players. All in all, the loose/aggressive player seemed to only be significantly better than the tight/aggressive player, which was not what we had expected. Furthermore, the tight/aggressive player seemed to be better overall against all other opponents than the loose/aggressive player.

One final interesting category in our results is the aggregate average pot size. There are certain pairings of styles of play that one would expect to have an overall larger or smaller average pot size than other pairings. As mentioned earlier, aggressive players tend to bet larger amounts. There is a good possibility for a larger pot size because of this but there is also a good possibility of making people fold their hand because of the larger bet, thereby reducing the average pot size. Whether or not the aggressive bet makes someone fold their hand depends on three things. One is the frequency of the bets. Another is the type of opponent the aggressive player is facing. The final is the strength of the hands of both players.

Many of the aggregate average pot sizes in our results can be explained by one or more of these three factors. Some, however, cannot be explained so easily. No matter what the pairing is in the match-up, if there are a low number of total hands played in that match-up, the average pot size will tend to be higher. This is because there had to be quite a few hands in the simulation where a large number of chips were exchanged, thereby raising the average pot size dramatically. This can be seen in the case of the largest aggregate average pot size in our results. This occurred in the match-up of the tight player and the loose/aggressive player. The average pot size was 452.47. However, if you look at the total hands in that simulation, you will see that it is the lowest out of all simulations, 375. That is an average of 37.5 hands per round of that simulation. It is obvious that since that simulation ended so quickly, there had to be certain rounds that lasted only a few hands or so. This was indeed the case, with some rounds containing only a few hands, some of which had pot sizes in the range of 1000 to 2000 chips. This being the case, it is apparent why the aggregate average pot size was the highest there. An explanation for why a simulation would last so few hands could be the strength of each players hand during a given hand. If both players happen to obtain a very strong hand at the same time, they will both most likely bet it strongly and try to protect it. However, only one of these hands could ultimately win, with the exception of the hands being identical. In the end, a considerable number of chips will be exchanged in the hand and one of the players will most likely be left with a very small amount of chips, if any.

We also observed that many of the simulations with a higher aggregate average pot size involved the loose/aggressive player. Also, many of the simulations with the lowest amount involved the tight/aggressive player. The loose/aggressive player plays a lot of hands and also bets aggressively in a majority of these hands. Taking this into consideration, a match-up with the loose/aggressive player should theoretically end up with a larger average pot size because there are many more chances for his opponent to call a larger bet and increase the pot size. This would also hold true to some extent with the regular aggressive player. The aggressive player just does not play as many hands. This being said, 5 of the 9 highest aggregate average pot sizes involved the loose/aggressive player and 4 of the 9 highest involved the aggressive player. What is not explained by these observations is why a match-up like passive vs. passive had the fifth highest aggregate average pot size. One would expect for passive players to end up with a lower pot size. It may be the case, as it was above, that they had a considerable number of hands where both players had very strong hands, which resulted in larger bets during the hand. Another observation that is hard to explain is why match-ups against passive players resulted in higher average pot sizes than ones against loose players. One would expect loose players to call larger bets more often than passive players. However, it may be the case that if we had a larger number of rounds in each simulation, this gap could have decreased or disappeared altogether.

When looking at the simulations with the lowest aggregate average pot sizes, we see that the three lowest involve the tight/aggressive player. The tight/aggressive player plays few hands but when he does play a hand, he plays it aggressively. This, in fact, does explain why simulations with a tight/aggressive player may end up with lower average pot sizes. When the tight/aggressive does get a hand that he can play aggressively, which is not very often, most of the time his opponent will most likely fold because he himself will not have a good enough hand to call the aggressive bet with. That being the case, the opponent will fold. The "rare" aggressive bet never becomes part of the pot and the average pot size stays low. These observations also would hold true for a tight player.

Overall, the observations above about the aggregate average pot sizes show that the behavior and betting schemes of the various AI players were basically what was expected, with certain exceptions that could partially be explained by the chances of both players having a very strong hand at the same time. We expected tighter players to be involved in fewer hands, thereby resulting in a smaller average pot size in most situations. This was observed in our results. We also expected more aggressive players and looser players to be involved in more hands in which they bet larger amounts, thereby resulting in a larger average pot size. This was also observed. Loose players who are not aggressive should be in a large number of hands, but they will not bet large amounts. This would keep the average pot size down because the number of hands is high but the size of the bets is not. This was seen in the match-ups involving loose players, which had an overall lower aggregate average pot size than other match-ups, as was expected. All in all, our AI players behaved more or less as was expected of them.

Conclusion

To summarize our accomplishments during this project, we were able to create a program that allows multiple players to connect to a poker server. The server can accommodate both real (human) and automated players. We also created 6 different AI players, each possessing its own style of play. Many of these players do play well in many situations as we envisioned in the beginning of this project. We tested our players' capabilities in one-on-one simulations. After analyzing all of our results, it seems that the overall best style of play is one that is either aggressive or loose but not necessarily both. However, that is not to say that a combination between the 2 styles does not perform solidly in many situations. If you decrease the degree of aggression and/or looseness (or maybe some other known or unknown qualities of a poker player), you may indeed find the "perfect" player. In the end, however, even the "perfect" player will lose some hands and matches. It may be the case that successful strategies may only be successful against certain types of opponents. This can be seen in professional poker tournaments where there is a variation between the winner's strategies from year to year. Of course, luck also plays a big part in determining who the winner will be. A very solid player can be playing his hand perfectly but at any moment, a "miracle card" can come up on the board to give another play a very improbable win. Another potential downfall to our AI player can be the number of opponents it faces. We evaluated our various AI players based on their heads-up performance, that is, one-on-one match-ups. The number of opponents is significant because it affects your odds of winning and therefore should affect your strategy. Heads up simplifies the play and allowed us to evaluate each player in that environment. We did not formally test our AI player against multiple opponents, but we can formulate an educated guess of how it would fair. The more opponents you have at your table, the more aware you must be of the different strategies you are facing. This means, for example, that you may have to play a little less aggressive in certain situations and maybe a little looser in others.

Looking back at our results, we concluded that an aggressive or loose strategy did the “best” overall during our simulations. However, even though a certain strategy may consistently lose to other strategies, it does not mean that it is incapable of making intelligent decisions. A really good player may play aggressive a majority of the time but he may also change gears a play a bit more passively in certain situations. When we manually examined some of the transcripts of the match-ups involving the passive player, we found some instances where it did indeed play well but it was difficult to quantify these results because of the fact that it lost in the end most of the time. If some of the qualities of the passive player were modified in one way or another, it is entirely possible that the passive player may have played better than what we observed. After all, there may be other strategies of poker that don’t fit into any of the categories mentioned above. A good human player will purposely make decisions that make it hard to predict his future actions. For example, he may lose a hand on purpose just to confuse an opponent and win a larger pot in the future. Certain strategies like this one could be hard to model in an AI player. It is everyone’s dream to find the “perfect” strategy, but that would be like knowing the winning lottery number everyday.

In retrospect, this project in general was a bit more difficult than was expected. We found that a lot of our time was taken up in the first semester when we had to create the client/server poker environment. Since we originally expected to evaluate our AI player against human players, we spent extra time on making sure that our program could be run over a network. We could have made a much simpler game from the beginning which would not have supported network play. This would have allowed us to test AI players against each other but not against human players. When we realized that a scientific evaluation of our project would be less feasible if we only tested it against human players, we shifted our focus away from enhancing the human behavior modeling features and towards creating different types of AI players. This shift allowed us to make a scientific evaluation more easily, which is an important aspect to any project. If we had more time, we could have done more testing against human players. At this moment, our program is capable of hosting up to 8 players, which could be either human or automated, with no extra programming necessary.

There are also some fundamental aspects of poker that may have hindered our results. An important part of poker is the psychological element of the game. In real games the opportunity to intimidate an opponent with a fierce gaze or notice an anxious opponent quivering can have a large impact on a player’s decision. When you play on the computer you lose that aspect of the game. This means that if someone could play in a live game exactly like our artificial players, they might fair differently. Another, perhaps more important aspect of poker is betting real money. Hosting an online gambling website in Pennsylvania is illegal so we use play money. In real life a player may be very cautious if they could lose a hundred dollars, but if its not real money, they would be willing to tolerate a higher level of risk. This could potentially result in much looser players that our AI players did not account for. In the end, however, we feel that our AI players showed intelligence and were successful in many situations. Although none of their strategies are “perfect”, they do indeed pose a challenge to fellow AI players as well as real human players.

References

Bernstein, Orr, Jonathan Margulies, and Cliff Tsai. *Intelligent Poker Player*.
Cornell University, Department of Computer Science.
http://www.csuglab.cornell.edu/~ob29/poker/algorithm.html

Billings, Darse, et al. *Opponent Modeling in Poker*.
University of Alberta, Department of Computer Science, 1998.
http://www.cs.ualberta.ca/~jonathan/Papers/Papers/aaai98.poker.ps

Hellmuth, Jr., Phil. *Play Poker Like the Pros*.
New York: HarperCollins Publishers Inc., 2003.

Sklansky, David, and Mason Malmuth. *Hold'em Poker For Advanced Players*.
Henderson, NV: Two Plus Two Publishing, 1999.

The University of Alberta Computer Poker Research Group.
http://www.cs.ualberta.ca/~games/poker/