Faculdade de Engenharia da Universidade do Porto

FEUP

Opponent Modelling in Texas Hold'em:

Learning Pre-flop Strategies in Multiplayer Tables

João Carlos Leite Ferreira

Master's Thesis submitted for the degree of
Master's in Electrical and Computers Engineering
Major Automação

Supervisor: Prof. Luís Paulo Reis

June 2008

Faculdade de Engenharia da Universidade do Porto

FEUP

Opponent Modelling in Texas Hold'em:

Learning Pre-flop Strategies in Multiplayer Tables

João Carlos Leite Ferreira

Dissertação do Projecto realizado no âmbito do
Mestrado integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Luís Paulo Reis

Junho de 2008

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES 2007/2008

A Dissertação intitulada

"Opponent Modelling in Texas Hold'em: Learning Pre-flop Strategies in Multiplayer Tables"

foi aprovada em provas realizadas em 01/Julho/2008

o júri

Presidente Professor Doutor Armando Jorge Miranda de Sousa
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

Professor Doutor João Carlos Balsa Silva
Professor Auxiliar da Faculdade de Ciências da Universidade de Lisboa

Professor Doutor Luis Paulo Gonçalves dos Reis
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - João Carlos Leite Ferreira

Faculdade de Engenharia da Universidade do Porto

"Our deepest fear is not that we are inadequate. Our deepest fear is that we are powerful beyond measure. It is our light, not our darkness that most frightens us. We ask ourselves, who am I to be brilliant, gorgeous, talented, fabulous? Actually, who are you *not* to be? You are a child of God. Your playing small does not serve the world. There is nothing enlightened about shrinking so that other people won't feel insecure around you. We are all meant to shine, as children do. We were born to make manifest the glory of God that is within us. It's not just in some of us; it's in everyone. And as we let our own light shine, we unconsciously give other people permission to do the same. As we are liberated from our own fear, our presence automatically liberates others."

Marianne Williamson

Abstract

In the last years poker changed from a simple card game to a world phenomenon attracting people from all over the world. This increasing admiration is not limited to the general population but also to the academic society. Many universities embraced research using poker as the main subject, most of them with the objective of designing an intelligent agent capable of playing poker at a world class level. Although the designed agents have became increasingly better every year they are still not able to compete with the best human players.

The academic society motivation to study the game of poker comes mainly from the fact that, contrary to other games, in poker, a world class player needs to judge, model and draw conclusion of the opponent playing style. This capacity will reflect directly on the results achieved by each player. The better the player may use it, the better results he will achieve. This means that poker provides a very good platform for developing intelligent agents with opponent modelling capabilities because the agent performance will correlate with the capacity to model his opponents.

The investigation on poker has been made essentially in limit heads-up poker. The main concerns were in opponent modelling that is needed, on this poker variant, to make a fast model of the opponent and create an appropriate playing strategy to play against that opponent. However, for this thesis, a different approach to the problem was made. It was considered that the opponent players, being intelligent, will adjust to the game, changing strategy when necessary. The thesis consists in determining which factors promote changes in that strategy and measure the important of different factors in the strategy. The work presents a causal model of the game of poker. This means, an abstract model that uses cause and effect logics to describe the behaviour of the system. It is established a group of possible causes (factors) and the effect in measure in groups of tests.

The results achieved show that factors like position of the player, number of players at the table, money and other player's actions, are indeed relevant for the strategy of the players. From these factors the actions of the others players is the one that causes the biggest changes of strategy. From the results it is also possible to conclude that the changes are not random but indeed follow a specific pattern.

Resumo

Nos últimos anos o poker passou de um mero jogo de cartas a um fenómeno mundial atraindo entusiastas de tudo o mundo. Este entusiasmo não se limitou ao público em geral mas também ao mundo académico. O desejo de construir um agente inteligente capaz de superar jogadores humanos instalou-se no seio da comunidade académica. Várias universidades entraram na competição apresentando todos os anos os seus resultados. Embora todos os anos os agentes desenvolvidos se tornem mais eficientes ainda não são capazes de superar os melhores jogadores humanos.

O grande fascínio para a comunidade científica vem do facto, ao contrário de em outros jogos, um bom jogador tem necessariamente de ser capaz de avaliar rapidamente o seu adversário. A sua capacidade de analisar e tirar conclusões sobre o adversário vai-se reflectir directamente nos resultados. Isto significa que o poker fornece uma plataforma incrível para o desenvolvimento de agentes inteligentes. O agente que melhor analisar o seu adversário terá os melhores resultados.

A investigação tem-se focado essencialmente em limit heads'up poker onde as principais preocupações, no que diz respeito ao oponente, são fazer um modelo o mais rápido possível e adaptar-se rapidamente caso o jogador mude de estratégia. Para esta tese seguiu-se uma linha de pensamento ligeiramente diferente. Foi considerado que um jogador, funcionando como um ser inteligente, vai-se adaptar às circunstâncias do jogo, mudando de estratégia quando o considere necessário. A tese consiste em determinar os factores que, influenciam uma mudança de estratégia do jogador e medir a sua influência. Este trabalho apresenta um modelo causal do jogo de poker, ou seja, uma lógica de causa efeito para descrever o comportamento do sistema. É estabelecido um conjunto de factores e o seu efeito é medido em testes.

Os resultados obtidos mostram que os factores posição, número de jogadores na mesa, dinheiro e acções dos outros jogadores, são de facto relevantes para a estratégia dos jogadores. Analisando os resultados é possível concluir que as alterações das estratégias não são aleatórias seguindo geralmente padrões.

Acknowledgements

I'd like to thank first to Prof. Luís Paulo Reis who supervised this work. For the motivation he always provided me. Without his dedication and guidance this thesis would not be possible.

I'd also to thank all my friends that keep interested and help on the thesis.

Index

1 Introduction 1

1.1 - Categorization of Games ... 2

1.1.1 - Perfect information and deterministic ... 2

1.1.2 - Imperfect information and stochastic ... 2

1.2 - Motivation ... 3

1.2.1 - Economy ... 3

1.2.2 - Science ... 4

1.2.3 - Psychology ... 4

1.3 - Goals ... 4

1.4 - Summary of contents ... 5

2 Introduction to Poker 7

2.1 - Hand Ranking ... 7

2.2 - Hand Probability ... 12

2.3 - Texas Hold'em ... 13

2.3.1 - Betting structure ... 14

2.3.2 - Play of the hand ... 15

2.4 - Conclusions ... 16

3 State of the Art 17

3.1 - Nash Equilibrium Strategies ... 17

3.2 - Best Response ... 18

3.3 - Restricted Nash Response ... 20

3.4 - Conclusions ... 20

4 Collecting Poker Data 21

4.1 - BetAndWin to Database ... 22

4.2 - Downloading the hands ... 23

4.3 - Joining the Hands 27
4.4 - Converting from Html to Text 27
4.5 - Converting from Text to a Database 29
4.6 - Conclusions 31

5 Game Models 33
5.1 - Theoretical Game Model 33
5.2 - Simplified Model 35
5.3 - Model Validation 36
5.3.1 - Search Tree algorithm 36
5.3.2 - Testing 47
5.3.2.1 - Number of players 47
5.3.2.2 - Position 50
5.3.2.3 - Actions 50
5.3.2.4 - Money 58
5.4 - Conclusions 60

6 Opponent Modelling 61
6.1 - Player Statistical Behaviour 61
6.1.1 - Algorithms 62
6.1.2 - Example 64
6.2 - Player Analysis 65
6.2.1 - Statistical Model 66
6.2.2 - Game Model 67
6.3 - Conclusion 69

7 Conclusion and Future work 70

Bibliography 71

Appendix A – Glossary of Poker Terms 74

List of Figures

FIG 1: STRAIGHT FLUSH 8
FIG 2: FOUR OF A KIND 9
FIG 3: FULL HOUSE 9
FIG 4: FLUSH 9
FIG 5: STRAIGHT 10
FIG 6: THREE OF A KIND 10
FIG 7: TWO PAIR 11
FIG 8: ONE PAIR 11
FIG 9: HIGH CARDS 12
FIG 10: THE POSITIONS AT THE TABLE [WIKIMEDIA 2008, ADAPTED] 14
FIG 11: BET AND WIN LOG ON 23
FIG 12: BET AND WIN PLAY NOW 23
FIG 13: BET AND WIN SELECTING TABLE 24
FIG 14: BET AND WIN SELECTING TABLE HISTORY 25
FIG 15: BET AND WIN TABLE HISTORY 25
FIG 16: DOWNLOADING TABLE HISTORY USING GETRIGHT 26
FIG 17: FOLDER WITH HTML FILES OF THE TABLE HISTORY 26
FIG 18: JOINING THE HTML FILES 27
FIG 19: CONVERTING HTML TO TXT 28
FIG 20 –FINAL RESULT - CONVERTED BETANDWIN GAME INTO A TXT FILE 29
FIG 21: GAME IN DATABASE 31
FIG 22: TREE STRUCTURE 37
FIG 23: SEARCH TREE ALGORITHM REACHING THE LAST BRANCH 38
FIG 24: SEARCH TREE ALGORITHM MEASURING THE IMPACT. 45
FIG 25: THREE PLAYERS' PRE-FLOP STRATEGY 47
FIG 26: FOUR PLAYER'S PRE-FLOP STRATEGY 48
FIG 27: FIVE PLAYER'S PRE-FLOP STRATEGY 49
FIG 28: DIFFERENCES BETWEEN THE PRE-FLOP STRATEGIES 49
FIG 29: FIVE PLAYERS PRE-FLOP STRATEGY 50

FIG 30: STRATEGIES ADOPTED BY THE PLAYERS WHEN FIRST PLAYER RAISES 51
FIG 31: IMPACT OF THE FIRST PLAYER RAISE 52
FIG 32: STRATEGIES ADOPTED BY THE PLAYERS WHEN FIRST PLAYER CALLS 52
FIG 33: IMPACT OF THE FIRST PLAYER CALLS 53
FIG 34: STRATEGIES ADOPTED BY THE PLAYERS WHEN FIRST PLAYER FOLDS 53
FIG 35: IMPACT OF THE FIRST PLAYER FOLDS 54
FIG 36: STRATEGIES ADOPTED BY THE PLAYERS WHEN SECOND PLAYER FOLDS 54
FIG 37: IMPACT OF THE SECOND PLAYER FOLDS 55
FIG 38: STRATEGIES ADOPTED BY THE PLAYERS WHEN SECOND PLAYER RAISES 55
FIG 39: IMPACT OF THE SECOND PLAYER RAISES 56
FIG 40: STRATEGIES ADOPTED BY THE PLAYERS WHEN FIRST AND SECOND PLAYER FOLD 56
FIG 41: IMPACT OF THE FIRST AND SECOND PLAYER FOLDING 57
FIG 42: STRATEGIES ADOPTED BY THE PLAYERS WHEN FIRST AND SECOND PLAYER RAISE 57
FIG 43: IMPACT OF THE FIRST AND SECOND PLAYER RAISE 58
FIG 44: JOHN PRE-FLOP STRATEGY BASED ON MONEY 58
FIG 45: PETER PRE-FLOP STRATEGY BASED ON MONEY 59
FIG 46: FRED PRE-FLOP STRATEGY BASED ON MONEY 59

List of Tables

TABLE 1: HAND PROBABILITY [WIKIPEDIA, 2008 ADAPTED] 13
TABLE 2: BETANDWIN TO DATABASE DOWNLOAD AND CONVERSION PROCESS 22
TABLE 3: PLAYER BETTING STRATEGY 64
TABLE 4: PLAYER MONEY DISTRIBUTION 65
TABLE 5: JIMMY RESULTS 66
TABLE 6: JIMMY MONEY DISTRIBUTION 67
TABLE 7: JIMMY PRE-FLOP STRATEGY 68
TABLE 8: JIMMY STRATEGY AFTER 1 PLAYER FOLD 68
TABLE 9: JIMMY STRATEGY AFTER FIRST AND SECOND PLAYER FOLDED 69

Chapter 1

1 Introduction

Artificial Intelligence may be defined as "the study and design of intelligent agents" , where an intelligent agent is a computational system that is able to perceive its environment and take appropriate actions which maximize its chances of success [Russel et al., 2003].

While there are accepted traits that are considered essential for Intelligence [Luger et al, 2004], there is not still a Universal accepted definition of Intelligence [McCarthy, 2007].

Artificial Intelligence (AI) in Games appeared in the 1970s, back then it consisted in small patterns that where used to create enemies. In the 1990s prompted the use of formal AI tools like finite state machines. Real-Time Strategy games taxed the AI with many objects, incomplete information, path finding problems, real-time decisions and economic planning, among other things [Schwab, 2004]. Recent games use nondeterministic AI methods, ranging from the first use of neural networks to the emergent behaviour and evaluation of player actions.

In History besides AI being used in games, there are games that were used as a research topic for AI cases like chess, scrabble or rochambeau made significant advances for AI.

AI research largely ignored Poker until recently, although it was a subject of research in other Academic Fields. Researchers like John von Neumann [Neumann, 1941] and John Nash [John Nash, 1951], used simplified poker to illustrate the fundamental principles of Game Theory [Billings et al., 2002]. Nevertheless Poker provides a number of attributes that make it an interesting domain for AI research. These properties include incomplete knowledge, multiple competing agents, risk management, opponent modelling, deception and dealing with unreliable information. All these properties make Poker a new and highly potential research topic for AI.

1.1 - Categorization of Games

Games may be categorized mainly using two parameters: The accessibility of the game state to the players; and whether or not contains chance elements. If all players in a game have complete knowledge of the entire game state, it is called a game with complete information. Otherwise, it is a game with incomplete information. If a game contains elements of chance, it is referred to as a stochastic game. Otherwise, it is refereed to as a deterministic game.

Chess, checkers and backgammon are all games of perfect information because all the players have access to the entire game state by just looking at the board. On the other hand, poker, bridge, and scrabble are examples of imperfect information games because each player has their own cards or tiles that are kept hidden from the other players in the game.

1.1.1 - Perfect information and deterministic

Research from 1970 to 1990 focused mainly on games of two players with perfect information and deterministic, which made chess the main game of AI research. Chess is a game with relatively simple logistic but complex strategy which realised in a deep look-ahead search of the game tree using efficient search algorithms, combined with domain-specific evaluation function applied at the nominal leaf nodes of the search frontier. The strategy of game is quite simple, make the best move possible. This approach made an increased emphasis on computer-oriented solutions, and diminishing reliance on human knowledge [Billings et al., 2002][Papp, 1998]. Although it is impossible to make in the near future the perfect program of chess as the game tree is 10^{40}[Shannon, 1950], research on Chess lost significance as the strategy for improvement is known and Chess programs have achieved dominance over human players.

1.1.2 - Imperfect information and stochastic

There are quite a few games that admit some elements of chance. The traditional game tree representation, which was researched from 1970 to 1990, can easily be extended to handle stochastic outcomes by incorporating changes nodes. Each branch will represent a finite number of random outcomes. To analyse the branch an Expected Value will be determined. Search algorithms like Minimax [Hauk et al.,2004], and simulation methods [Tesauro, 2002] are able to handle the search properly. The property of stochasticity has not been the major impediment to world-class computer performance for poker in practice [Billings, 2006]. The game of Backgammon is a perfect example

of a game with perfect information and elements of stochasticity (rolling the dice) that was solved using evaluation functions learned using self-play. The resulting programs were able to be on pair with the best human players.

The major difference between Poker and other games is the property of imperfect information. Similar to Backgammon, in Poker the best strategy aims to make the play the move which provide the best Expected Value (EV), but in opposite with Backgammon in Poker the best EV is deeply correlated with the opponent strategy. In Poker there is also need to change strategies as a static strategy will be read by the opponent and possibly exploited.

These properties make it necessary for a complete poker strategy to incorporate a certain degree of deception (Bluffing and Trapping), perfect odds calculation, opponent modelling and mixed strategies.

1.2 - Motivation

Poker has become a very interesting field of research in three major areas: economic, scientific and psychology, mainly. This *phenomena* it's easily understood when analysing the importance of this game in the entertainment world (eg.: Poker is a source of huge profit in casinos, etc...) and its huge indirect impact in many other areas of society. All this conditions make the increasing investment on Poker's research very profitable and useful in the future (as the advances in this area of knowledge will always have a potential utility and application in other fields like economy or marketing).

1.2.1 - Economy

The direct Economic impact in Poker gambling is estimated to US$ 200 million per month worldwide [Wikipedia, 2008]. Although it is illegal to use agents (bots) in most on-line Poker playing servers to gain profit from human players, it is possible to develop programs to help and teach the players. This new market has made research in Poker profitable. Among the most well-known poker programs is the software "Poker Academy inspector [Poker Academy, 2008]" of the University of Alberta. The program analyses players in order to find potential weakness in their game play.

The research of poker has also economic benefits in the developing of agents and opponents. The research of modelling players with scarce information is very interesting for marketing and publicity. The main objective is to be able to know costumers with less information possible, and to be able to keep up with market

changes. This is easily compared with the poker domain where little information about the opponent is known and he his constantly changing strategy.

1.2.2 - Science

Poker has very high potential for scientific research not only in AI but also in Economy and Operational Research. The game involves challenges that are commonly presented in the real world. Features like incomplete knowledge, competition, risk management, opponent modelling, deception, dealing with unreliable information are real aspects of the business world. Studding them provides superior odds to win. In the words of Sun Tzu "The general who wins the battle makes many calculations in his temple before the battle is fought. The general who loses makes but few calculations beforehand" [Sun Tzu, 6th century BC]. Although society general belief is that poker is a game of luck it as been proven both practical [DeDonno, 2008] and theoretical [Billings, 2006] that poker is a game of skill. Poker can be compared with an investment where many companies compete in a series of steps. None of the companies knows clearly which value the other company has, only it's own. They have to select a strategy that tries to know other companies value without revealing their own. This while never forgetting their real value and always consider the risk of pretending to have something they do not have. This small analogy demonstrates how poker research and game theory can be important for the business work.

1.2.3 - Psychology

Poker has been abroad by psychologist and some books have been written about the subject. The new topic in research will be combining studies of opponent modelling with psychology [McKenna, 2005]. This will provide a new and ambitious research that will bring researches closer to understand the emotions behind the opponents.

1.3 - Goals

The goal of this project, similar to all poker projects, is to help building an intelligent agent capable of playing poker at a world class level. In this thesis the players pre-flop strategy in multiplayer tables will be analysed. This will contribute for the opponent modelling in multiplayer tables which is essential component of an intelligent agent.

The project will be separated in four main objectives.

- *Gathering real games from online multiplayer poker tables;*
- *Creation of a game model for poker;*

- *Validation of the game model;*
- *Using information of saved games in the game model to analyse players.*

The main goal of analysing pre-flop strategies in multiplayer tables will be achieved if the tools developed provide relevant statistic information that can be used by an agent.

1.4 - Summary of contents

This document is structured in seven chapters.

The first one is the introduction to the thesis. Games and Artificial Intelligence are presented in a simple approach, and it's explained the motivation for this kind of work.

In the second chapter the game of Poker is briefly explained. Some basic aspects of the game are covered along with the most common variations.

The third chapter categorizes the state of the art of the concepts used in the thesis. The chapter reviews the research done in Poker and in the variant Texas Hold'em.

The fourth chapter describes the procedure used for collecting data from online poker servers.

The fifth chapter presents a game model, along with its validation and some tools developed.

In the sixth chapter the game model is tested for opponent modelling, and the results achieved are analysed and discussed.

The conclusions and future work of the project are presented in the seventh and last chapter.

Chapter 2

2 Introduction to Poker

Poker is a popular type of card game in which players bet on the value of the card combination ("hand") in their possession, by placing a bet into a central pot. The winner is the one who holds the hand with the highest value or otherwise the player who remains "in the hand" after all others have folded (the player who makes an un-called bet.).

The three most popular poker variants are spread in casinos and poker rooms worldwide and can be divided into the following groups:

- Draw Poker: Games in which players are dealt a complete hand, hidden, and then improve it by replacing cards. The most common of these is Five-card-draw.
- Stud Poker: Games in which each player receives a combination of face-up cards and face-down cards in multiple betting rounds. The most common of these are Seven-card-stud (7 Stud and 7 Stud H/L) and Five-card-stud.
- Community Card Poker: Games in which each player's incomplete hidden hand is combined with shared face-up cards. The most common of these are Texas Hold'em and Omaha (Omaha Hi and Omaha H/L).

2.1 - Hand Ranking

In poker, certain combinations of cards, or *hands*, outrank other hands. The player with the best poker hand at the showdown wins the pot.

The following general rules apply to evaluating poker hands, whatever set of hand values are used:

- Individual cards are ranked A (high), K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2, A. Aces only appear low when part of an A-2-3-4-5 straight or straight flush. Individual

card ranks are used to compare hands that contain no combination.

- Suits have no value. The suits of the cards are mainly used in determining whether a hand fits a certain combination (specifically the flush and straight flush hands). In most variants, if two players have hands that are identical except for suit, then they are tied and split the pot (so 3♠ 4♠ 5♠ 6♠ 7♠ does not beat 3♦ 4♦ 5♦ 6♦ 7♦).
- A hand always consists of five cards. In games where more than five cards are available to each player, the best five-card combination of those cards plays.
- Hands are ranked first by combination, then by individual card ranks: even the lowest qualifying hand in a certain combination defeats all hands in all lower combinations. The smallest two pair hand (2♦ 2♠ 3♦ 3♣ 4♠), for example, defeats all hands with just one pair or high card. Only between two hands in the same category are card ranks used to break ties.

The combinations of cards in Poker are described next, in order from the highest to lower.

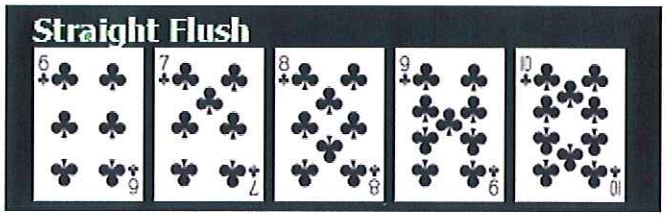


Fig 1: Straight Flush

A straight flush is a poker hand such as Q♣ J♣ 10♣ 9♣ 8♣, which contains five cards in sequence, all of the same suit. Two such hands are compared by their highest card; since suits have no relative value, two otherwise identical straight flushes tie (so 10♣ 9♣ 8♣ 7♣ 6♣ ties with 10♥ 9♥ 8♥ 7♥ 6♥). Aces can play low in straights and straight flushes: 5♦ 4♦ 3♦ 2♦ A♦ is a 5–high straight flush, also known as a "steel wheel". An ace-high straight flush such as A♠ K♠ Q♠ J♠ 10♠ is known as a royal flush, and is the highest ranking standard poker hand.

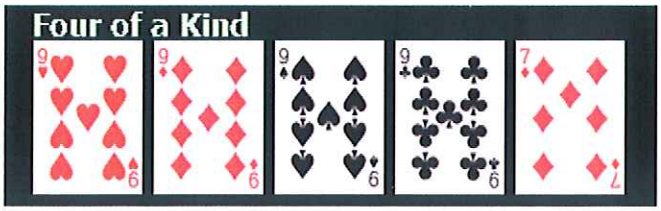


Fig 2: Four of a kind

Four of a kind, also known as quads, is a poker hand such as 9♣ 9♠ 9♦ 9♥ J♥, which contains four cards of one rank, and an unmatched card of another rank. It ranks above a full house and below a straight flush. Higher ranking quads defeat lower ranking ones. In community-card games (such as Texas Hold'em) or games with wildcards it is possible for two or more players to obtain the same quad; in this instance, the unmatched card acts as a kicker, so 7♣ 7♠ 7♦ 7♥ J♥ defeats 7♣ 7♠ 7♦ 7♥ 10♣.

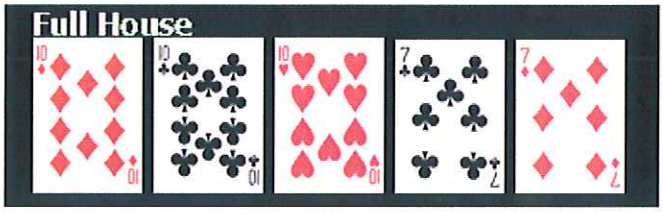


Fig 3: Full House

A full house, also known as a full boat, is a hand such as 3♣ 3♠ 3♦ 6♣ 6♥, which contains three matching cards of one rank, and two matching cards of another rank. It ranks below a four of a kind and above a flush. Between two full houses, the one with the higher ranking set of three wins, so 7♠ 7♥ 7♦ 4♠ 4♣ defeats 4♠ 4♥ 4♦ 7♠ 7♣. If two hands have the same set of three (possible in wild card and community card games), he hand with the higher pair wins, so **5♣ 5♦ 5♠ J♠ J♦** loses to **5♥ 5♦ 5♠ Q♥ Q♣**.

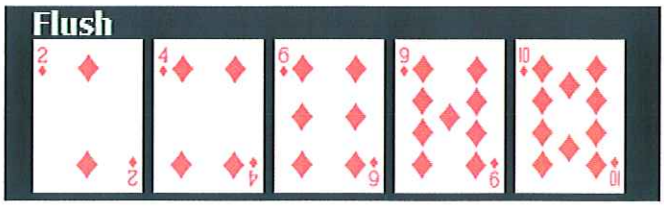


Fig 4: Flush

A flush is a poker hand such as Q♣ 10♣ 7♣ 6♣ 4♣, which contains five cards of the same suit, not in rank sequence. It ranks above a straight and below a full house. Two flushes are compared as if they were high card hands; the highest ranking card of each is compared to determine the winner. If both hands have the same highest card, then the second-highest ranking card is compared, and so on until a difference is found. If the two flushes contain the same five ranks of cards, they are tied – suits are not used to differentiate them.

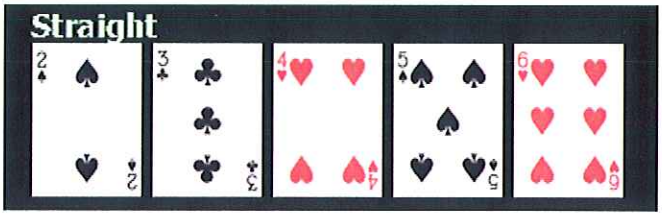


Fig 5: Straight

A straight is a poker hand such as Q♣ J♠ 10♠ 9♥ 8♥, which contains five cards of sequential rank but in more than one suit. It ranks above three of a kind and below a flush. Two straights are ranked by comparing the highest card of each. Two straights with the same high card are of equal value, suits are not used to separate them.

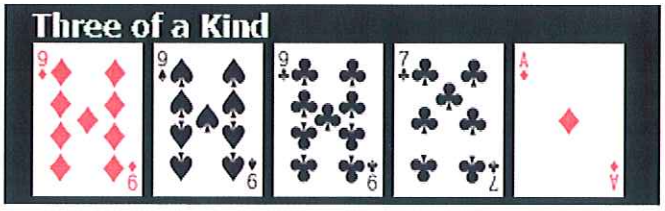


Fig 6: Three of a Kind

Three of a kind, also called trips, is a poker hand such as 2♦ 2♠ 2♣ K♠ 6♥, which contains three cards of the same rank, plus two unmatched cards. It ranks above two pair and below a straight. In Texas Hold'em and other flop games, a "set" refers specifically to a three of a kind composed of a pocket pair and one card of matching rank on the board. Higher-valued three of a kind defeat lower-valued three of a kind, so Q♠ Q♥ Q♦ 7♠ 4♣ defeats J♠ J♣ J♦ A♦ K♣. If two hands contain threes of a kind of the same value, possible in games with wild cards or community cards, the kickers are compared to break the tie, so 4♦ 4♣ 4♠ 8♦ 6♣ defeats 4♦ 4♣ 4♠ 6♣ 5♦.

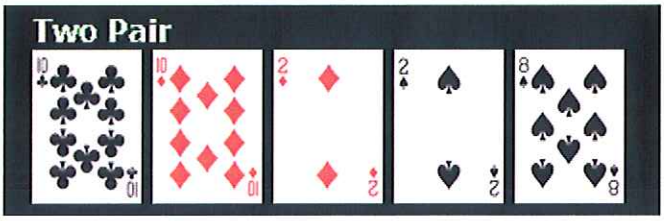


Fig 7: Two Pair

A poker hand such as J♥ J♣ 4♣ 4♠ 9♥, which contains two cards of the same rank, plus two cards of another rank (that match each other but not the first pair), plus one unmatched card, is called two pair. It ranks above one pair and below three of a kind. To rank two hands both containing two pair, the higher ranking pair of each is first compared and the higher pair wins (so 10♠ 10♣ 8♥ 8♣ 4♠ defeats 8♥ 8♣ 4♠ 4♣ 10♠). If both hands have the same "top pair", then the second pair of each is compared, such that 10♠ 10♣ 8♥ 8♣ 4♠ defeats 10♠ 10♣ 4♠ 4♥ 8♥). Finally, if both hands have the same two pairs, the kicker determines the winner.

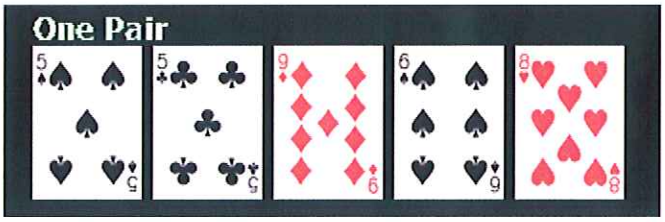


Fig 8: One Pair

One pair is a poker hand such as 4♥ 4♠ K♠ 10♦ 5♠, which contains two cards of the same rank, plus three other unmatched cards. It ranks above any high card hand, but below all other poker hands. Higher ranking pairs defeat lower ranking pairs; if two hands have the same pair, the non-paired cards (the kickers) are compared in descending order to determine the winner.

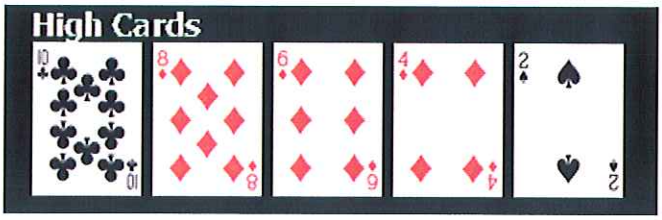


Fig 9: High Cards

A high-card or no-pair hand is a poker hand such as K♥ J♣ 8♣ 7♦ 3♠, in which no two cards have the same rank, the five cards are not in sequence, and the five cards are not all the same suit. It is also referred to as "no pair", as well as "nothing", "garbage," and various other derogatory terms. High card ranks below all other poker hands; two such hands are ranked by comparing the highest ranking card. If those are equal, then the next highest ranking card from each hand is compared, and so on until a difference is found.

2.2 - Hand Probability

The combination order chosen in poker is connected with the probability of occurrence. The straight flush is the most improbable and the no-pair is the most likely to occur. The hand probability is not the same in all poker games. Some poker games use 5 cards as other like Texas Hold'em use 7 cards from each the player should make the best 5 hand possible. This significant changes the probability of the combination. In table 1 it is possible to see the probability of the combinations.

Table 1: Hand Probability [Wikipedia, 2008 adapted]

	7 cards				5 cards			
Hand	Frequency	Probability	Cumulative	Odds	Frequency	Probability	Cumulative	Odds
Straight flush	41,584	0.0311%	0.0311%	3216 : 1	36	0.00154%	0.00154%	72192.33 : 1
Four of a kind	224,848	0.168%	0.199%	594 : 1	624	0.0240%	0.0256%	4164 : 1
Full house	3,473,184	2.60%	2.80%	37.5 : 1	3744	0.144%	0.170%	693.2 : 1
Flush	4,047,644	3.03%	5.82%	32.1 : 1	5,108	0.197%	0.367%	507.8 : 1
Straight	6,180,020	4.62%	10.4%	20.6 : 1	10,2	0.392%	0.76%	253.8 : 1
Three of a kind	6,461,620	4.83%	15.3%	19.7 : 1	54,912	2.11%	2.87%	46.3 : 1
Two pair	31,433,400	23.5%	38.8%	3.26 : 1	123,552	4.75%	7.62%	20.03 : 1
One pair	58,627,800	43.8%	82.6%	1.28 : 1	1,098,240	42.3%	49.9%	1.37 : 1
No pair	23,294,460	17.4%	100%	4.74 : 1	1,302,540	50.1%	100%	0.995 : 1
Total	133,784,560	100%	100%	0 : 1	2,598,960	100%	100%	0 : 1

It is possible to see that in the 7 card Poker some combinations are more probable to occur than in the five card poker. It is also possible to see that change in the same way, for example the straight in the 5 cards is 8 times less probable than the three of a kind, in the 7 cards is 1.2 less probable.

2.3 - Texas Hold'em

In Texas Hold'em, like all variants of poker, individuals compete for the pot. Because the cards are dealt randomly and outside the control of the players, each player attempts to control the amount of money in the pot based on the hand the player holds.

The game is divided into a series of hands or deals; at the conclusion of each hand the pot is awarded to one or more players. A hand ends either at the showdown (when the remaining players compare their hands), or when all but one player have folded and abandoned their claims to the pot. The pot is then awarded to the player(s) who have not folded and have the best hand. (This is usually only one player, but can be more in the case of a tie.)

The objective of winning players is not winning every *individual* hand, but rather making mathematically correct decisions regarding when and how much to bet, raise, call or fold. By making such decisions, winning poker players maximize long-term winnings by maximizing their expected utility on each round of betting.

2.3.1 - Betting structure

Hold'em is normally played using small and big blind bets – forced bets by two players.. A dealer button is used to represent the player in the dealer position; the dealer button rotates clockwise after each hand, changing the position of the dealer and blinds. The small blind (Fig 10) is posted by the player to the left of the dealer and is usually equal to half of the big blind. The big blind , posted by the player to the left of the small blind, is equal to the minimum bet.

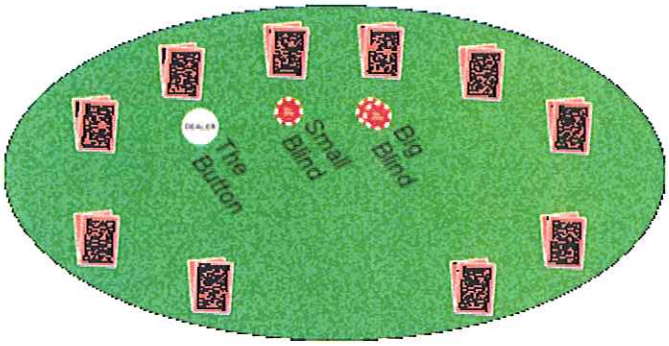


Fig 10: The positions at the table [Wikimedia 2008, adapted]

In tournament poker, the blind/ante structure periodically increases as the tournament progresses. (In some cases, the small blind is some other fraction of a small bet, e.g. $10 is a common small blind when the big blind is $15. The double-blind structure described above is a commonly used and more recent adoption.)

When only two players remain, special 'head-to-head' or 'heads up' rules are enforced and the blinds are posted differently. In this case, the person with the dealer button posts the small blind, while his/her opponent places the big blind. The dealer acts first before the flop. After the flop, the dealer acts last for the remainder of the hand.

The three most common variations of Hold'em are limit Hold'em, no-limit Hold'em and pot-limit Hold'em. Limit Hold'em has historically been the most popular form of Hold'em found in casino live action games in the United States. In limit Hold'em, bets and raises during the first two rounds of betting (pre-flop and flop) must be equal to the

big blind; this amount is called the small bet. In the next two rounds of betting (turn and river), bets and raises must be equal to twice the big blind; this amount is called the big bet. No-limit Hold'em is the form most commonly found in televised tournament poker and is the game played in the main event of the World Series of Poker. In no-limit Hold'em, players may bet or raise any amount over the minimum raise up to all of the chips the player has at the table (called an all-in bet). If someone wishes to re-raise, they must raise at least the amount of the previous raise. For example, if the big blind is $2 and there is a bet of $6 to a total of $8, a raise must be at least $6 more for a total of $14. If a raise or re-raise is all-in and does not equal the size of the previous raise, the initial raiser can not re-raise again. This only matters of course if there was a call before the re-raise. In pot-limit Hold'em, the maximum raise is the current size of the pot (including the amount needed to call).

2.3.2 - Play of the hand

Play begins with each player being dealt two cards face down. (Like most poker games, the deck is a standard 52 card deck, no jokers.) These cards are the player's hole or pocket cards. These are the only cards each player will receive individually, and they will only (possibly) be revealed at the showdown, making Texas Hold'em a closed poker game.

The hand begins with a "pre-flop" betting round, beginning with the player to the left of the big blind (or the player to the left of the dealer, if no blinds are used) and continuing clockwise. A round of betting continues until every player has folded, put in all of their chips, or matched the amount put in by all other active players. See betting for a detailed account. Note that the blinds are considered "live" in the pre-flop betting round, meaning that they contribute to the amount that the blind player must contribute, and that, if all players call around to the player in the big blind position, that player may either check or raise.

After the pre-flop betting round, assuming there remain at least two players taking part in the hand, the dealer deals a flop, three face-up community cards. The flop is followed by a second betting round. This and all subsequent betting rounds begin with the player to the dealer's left and continue clockwise.

After the flop betting round ends, a single community card (called the turn or fourth street) is dealt, followed by a third betting round. A final single community card (called the river or fifth street) is then dealt, followed by a fourth betting round and the showdown, if necessary.

In all casinos, the dealer will burn a card before the flop, turn, and river. Because of this burn, players who are betting cannot see the back of the next community card to come, which might be marked.

If a player bets and all other players fold, then the remaining player is awarded the pot and is not required to show his hole cards. If two or more players remain after the final betting round, a showdown occurs. On the showdown, each player plays the best five-card poker hand he can make from the seven cards comprising his two hole cards and the five community cards. A player may use both of his own two hole cards, only one, or none at all, to form his final five-card hand. If the five community cards form the player's best hand, then the player is said to be playing the board and can only hope to split the pot, since each other player can also use the same five cards to construct the same hand.

If the best hand is shared by more than one player, then the pot is split equally among them, with any extra chips going to the first players after the button in clockwise order. It is common for players to have closely-valued, but not identically ranked hands. Nevertheless, one must be careful in determining the best hand; if the hand involves fewer than five cards, (such as two pair or three of a kind), then kickers are used to settle ties (see the second example below). Note that the card's numerical rank is of sole importance; suit values are irrelevant in Hold'em.

2.4 - Conclusions

This chapter presented the game of Poker and its variants. It was explained the hand raking system and the different probably of each group of hands. The variant Texas Hold was analysed in more detail.

This chapter provides basic knowledge of the game and of the variant Texas Hold'em that is needed for the full understanding of the following chapters.

Chapter 3

3 State of the Art

The University of Alberta Computer Poker Research Group (CPRG) is the major contributor to the academic literature on poker game-playing Artificial intelligence (AI) [University of Alberta Games, 2008]. The research team at the University of Alberta has published the most renowned works about AI applied to poker. More recently, the papers from Billings [Billings, 2006], Johanson [Johanson, 2007] and Nolan Bard in March 2008 [Nolan Bard, 2008] deserve reference. Most of the work is aimed to build a world class artificial poker player. Poker Research is not limited to the CPRG, as many other researchers have published well-known papers in very good journals and conferences [Gilpon et al,. 2006] [Gilpin et al., 2007] [Félix et al, 2008] Nevertheless is important to state that as CPRG is the leader in poker research and any paper concerning poker should have in mind the research in CPRG and if possible compare it with their results.

In order to build a world class poker player two different approaches have been studied. The first one is to build a poker player that correctly simulates a Nash-Equilibrium [John Nash, 1951]; the second is one that exploits the opponent's weakness in order to make the best move possible [Billings et al., 2004].

3.1 - Nash Equilibrium Strategies

An equilibrium strategy plays close to the Nash equilibrium in its abstraction, making it near-unbeatable by any other strategy. If the abstraction is strategically similar to the real game, then these equilibrium strategies can also be very difficult to defeat in the full game [Johanson, 2007]. As poker does not consists in one single decision per game but several and due to presence of stochasticity a perfect equilibrium strategy is considered difficult and no project in the near future aims to achieve that. Nevertheless programs based on Nash equilibrium achieved excelent results even against strong

poker players each new project surpassing his ancestor. CPRG has developed PsOpti4, PsOpti6, PsOpti7, Smallbot 2298 and more recently "Counterfactual Regret Minimization" [Johanson, 2007]. Gilpin and Sandholm produced G1, G2 and G3 [Gilpin et al., 2006] [Gilpin et al., 2007].Programs based on a Nash Equilibrium consist in study poker game in death, often by simulation millions of hands. From the result of million of hands they aim to achieve the best possible strategy. The result is a static strategy that consists in the best average strategy. In order to this static strategy not be discovered by the opponents the designers of the programs often implemented a random variable in the decision process. The first results (PsOpti4) show that Equilibrium strategies to be effective against intermedium poker players, but vulnerable against professional players. In 2007 in AAAI computer poker competition PsOpto7 had the best results followed by G3.

Although a Nash Equilibrium represents a great achievement, is also the safest one. The theory in which is based assumes that the other player is perfect and therefore is going to make the best move possible, which not always is correct. To better explain this weakness he should analyse the game of rochambeau (scissors, rock, paper).

Rochambeau (sometimes spelled roshambo) is a two player game here both players select at the same time with their hands an action. There are three actions rock, scissor and paper. The winner is decided comparing the actions. Rock wins against scissors, scissors wins against paper and paper wins against rock. If both players select the same action there is a tie. Contrary to truly random selections, this game can be played with skill, as a player can often recognize and exploit the non-random behaviour of an opponent.

In Rochambeua, the perfect Nash Equibrium would be to random select a move. The random selection strategy will guarantee a 1/3 to win 1/3 to love and 1/3 of draw. Using this strategy in the long run the player will win 33,3% of the times and draw 33,3%.

.

3.2 - Best Response

The best response theory consists in the paradigm that for every strategy, there is the best possible response. Picking up the idea of rochambeua the best counterstrategy against a strategy that always plays rock, would be to always play paper. This is a good example were a Nash Equilibrum strategy would be negative. In poker to determine the best counterstrategy is more complicated, but the challenge does not rely there. As many papers and programs have already demonstrated a stationary poker playing

strategy is easily exploited. The challenge is to determine the player strategy fast and be able to keep up with changing strategies.

Adaptive programs are designed to build an opponent model of the player and build a best strategy against him. Vexbot [Billings et al., 2004] and BRPlayer [Schauenberg, 2006] were developed by UCLA with that objective. Both use miximix search [Johanson, 2007]. This is similar to the expectimax search which is used for stochastic games, extended to accommodate imperfect information games. In a minimix search the program game tree with an opponent model that represents the player decisions at the choice nodes [Johanson, 2007].

The opponent model needed by miximix search must predict two types of information: opponent action probabilities at their choice nodes, and the utility for reaching each terminal node. To predict the opponent action probabilities, Vexbot and BRPlayer measure the frequency of the opponent's actions for each betting history [Johanson, 2007].

The main advantage of adaptive agents is their ability to learn online and adapt their play to exploit their opponent. In experiments performed by Billings [Billings et al., 2004] Vexbot was shown to defeat PsOpti4, and to defeat a variety of opponents by a much greater margin than PsOpti4 did.

This shows the value of an adaptive strategy over a Nash equilibrium strategy. As we mentioned the goal is not necessary not lose, but to win as much as possible from each opponent.

The adaptive agents developed have several disadvantages. As they start with no knowledge of the opponent or the game, they can develop impossible beliefs about the game. The knowledge about the possible hands the opponent can have is learned through observation, and is represented in the histograms at the terminal nodes. If the game starts with a lucky set of cards for the opponent, for example, adaptive agents can learn that their opponent always wins at showdowns and never folds. These observations can affect the miximix search such that it avoids lines of play where it (incorrectly) believes it will always lose; it may take many more hands of observations to correct this mistake [Johanson, 2007].

Also these techniques require a considerable amount of data to be effective. In experiments with BRPlayer against PsOpti4, BRPlayer took between 20,000 to 175,000 hands before it was able to break even [Billings, 2006]. Obviously, real players never play that long making it impossible to use these techniques to create useful models of them.

3.3 - Restricted Nash Response

Restricted Nash response is a new strategy created by Johansson [Johanson, 2007]. This strategy combines the approaches of the Nash equilibrium and the Best response. Both Nash equilibrium and best response algorithms were enhanced in his thesis. The enhanced algorithms were called Counterfactual Regret Minimization and Frequentist Best Response. The restricted Nash Response represents a new strategy that trains itself to exploit opponents but has a parameter to determine how much emphasis to give to the exploitation compared to minimizing its own exploitability. The higher the parameter gets the more the programs moves away from the Nash equilibrium in order to exploit the opponent.

3.4 - Conclusions

Although many approaches exist poker research was focused mainly in Nash Equilibrum and Best Response strategies for Head'up Poker. The last poker strategy consisted in joining the two main strategies. The result surpassed the previous strategies both still does not mach world class poker players.

The research done in this thesis differs from the current state of the art. Instead of focusing on Heads'up poker we will consider a multiplayer table. While a significant part of Heads'up research applies to multiplayer tables there are significant changes. The most significant one will be the growth of the space of possible outcomes. As the choices grow the space of the game tree grows exponential. This results in a game tree too big to considered conventional search tree algorithms. For the project we will approach multiplayer opponent modelling focusing in the pre-flop strategy.

Chapter 4

4 Collecting Poker Data

In order to develop the proposed research it was necessary to collect information regarding a large number of hands played in multiplayer tables. In the internet it is possible to find information regarding history of poker hands from many sources. For this thesis three different sources could be potentially used:

- *Mirc Database of Alberta* – The Mirc Database at the University of Alberta contains the IRC poker games from 1995-2001 [Irc poker database, 2008] .
- *PokerTracker* – Poker Tracker is a software enabling collecting and analyse of poker history hands from Online Poker sites.
- *Individual Online Poker Sites* – There are many individual online poker playing sites, including BetAndWin, Poker Stars, Party Poker , Everest Poker and Absolute poker. All sites allow access to history but only BetAndWin allows access as an observer.

One of the objectives of the thesis is to analyse real players, for that it is necessary that the database hands be equal to a real poker game. For that reason Mirc Database was rejected as it consisted in Poker games played with virtual money.

PokerTracker is a program that collects hands from different online poker sites and analyses it [Poker Tracker, 2008]. The main advantage of the program is that the format in which the information is stored is common for all sites. The program itself does not contain any database so it's needed to import it from the websites. This is problematic because though most poker sites provide access to the historical of the hands, most of them only allow access to the games where the player takes part. This makes it necessary to visit all Poker sites in order to find one that allowed saving information as a spectator.

The Online Poker site of BetAndWin allows everyone to check past hands played in the server. Opposite to other sites the hands are stored in a HTML format online for a period of 1 month, after that being deleted. BetAndWin is not compatible with PokerTracker which make it impossible to import the hands from BetAndWin to PokerTracker.

Other Major online Poker sites were consulted like Poker Stars, Party Poker, Everest Poker and Absolute Poker. All of these apparently did not allow to save information of the hands as spectators.

For that reasons, from the possibilities gathered, the only one suited for building the intended database of hands from multiplayer poker tables was BetAndWin.

4.1 - BetAndWin to Database

The website BetAndWin does not provide any tool to save the information, so a procedure was developed to collect the data. In BetAndWin every hand is represented in a webpage. For example a game would consist in all the pages of: http://www.pokernetwork.co.uk/games/handhistory/?sRoundReference=P4-67135066-[1-255].html, being 1 the first hand and 255 the last.

In order to transform this information to a useful database a number of steps had to be taken. Figure shows all the steps necessary.

Table 2: BetAndWin to database download and conversion process

Step 1: Log on to bet and win
Step 2: Select the table
Step 3: Download the hands
Step 4: Join the Hands
Step 5: Convert from html to txt
Step 6: Convert to database

To access the games of BetAndWin is necessary to join and login to the BeTandWin website. In the website after choosing to play poker, it is possible to select the table to observe.

In the table in right corner a link to the historical will appear. All the Hands are record in individual html pages. It is necessary to download and join all the hands.

After joining all the hands in order to make it easier the process of converting to a database the html file is converted to a txt file. After that step we are ready to the final step were the file is converted to a database

4.2 - Downloading the hands

The procedure to download the hands from BetAndWin is simple. In figure 11 the main page of BetAndWin website can be seen, to access the poker games it is necessary to log on first.

Fig 11: Bet and win log on

After login on and selecting poker the webpage represented by figure 12 will appear.

Fig 12: Bet and win play now

BetAndWin provides the options of downloading a small poker program of playing online. It was selected playing online "jogue agora" in Portuguese. At this moment step 1 is complete.

As figure 13 shows, once inside the poker server it is necessary to select a table. To select a table just click on it. It was selected the table pyoengyan.

Fig 13: Bet and win selecting table

In figure 14 it is possible to see the pyoengyan table with five players. To select the previous hands a link in the upper right corner will appear. Clicking on it will give access to all the hands played in the table so far.

Fig 14: Bet and Win selecting table history

After this step 2 will be complete.

Figure 15 shows the history of one hand. In this particular case the hand number 488 "upper right corner".

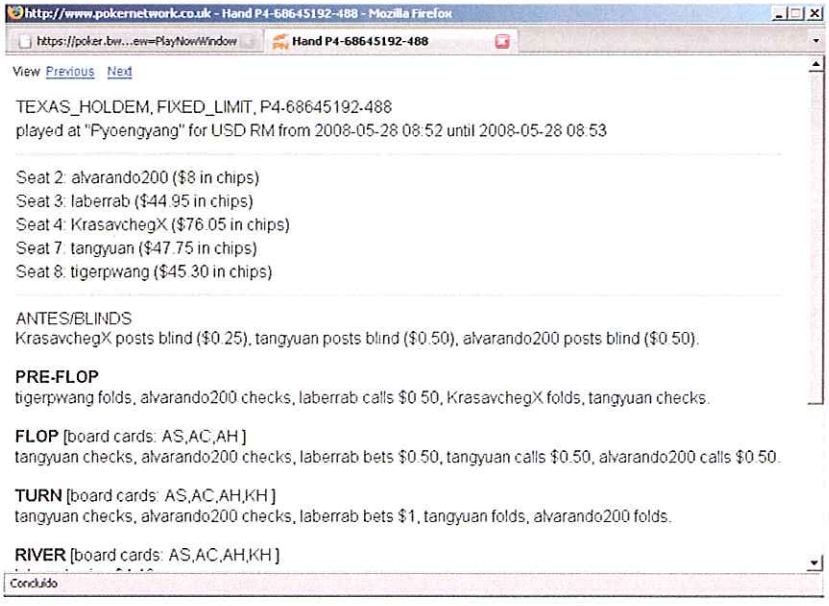


Fig 15: Bet and Win table history

To download all the hands the software Getright [GetRight] was used. Getright provides a utility to download a sequence of Pages which was very useful. Figure 16 shows the procedure to download all the hands using Getright.

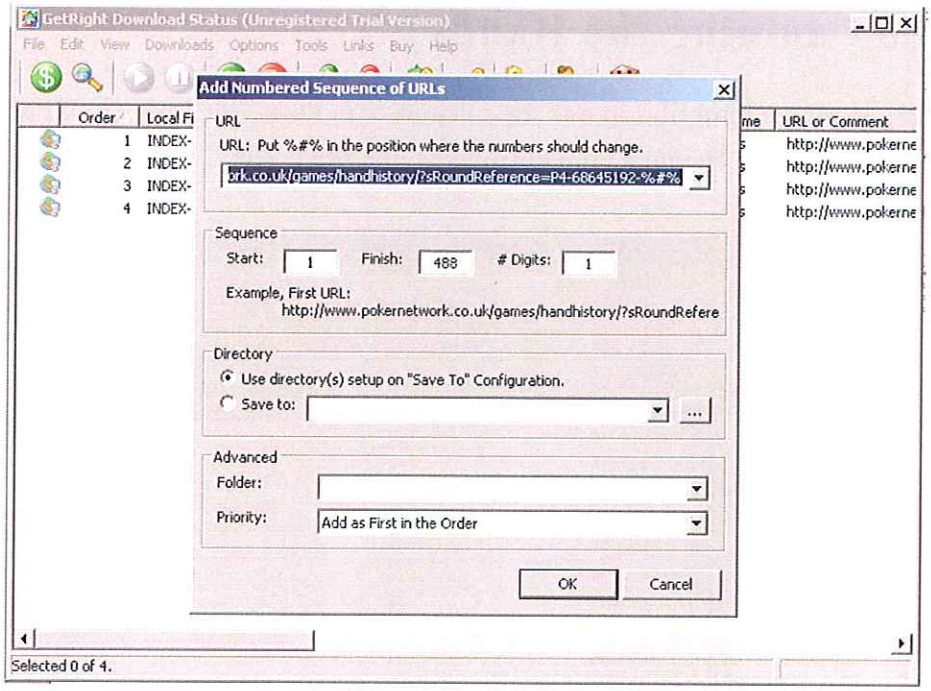


Fig 16: Downloading table history using Getright

The result will be a folder with all the hands played in the game which can be seen in figure 17.

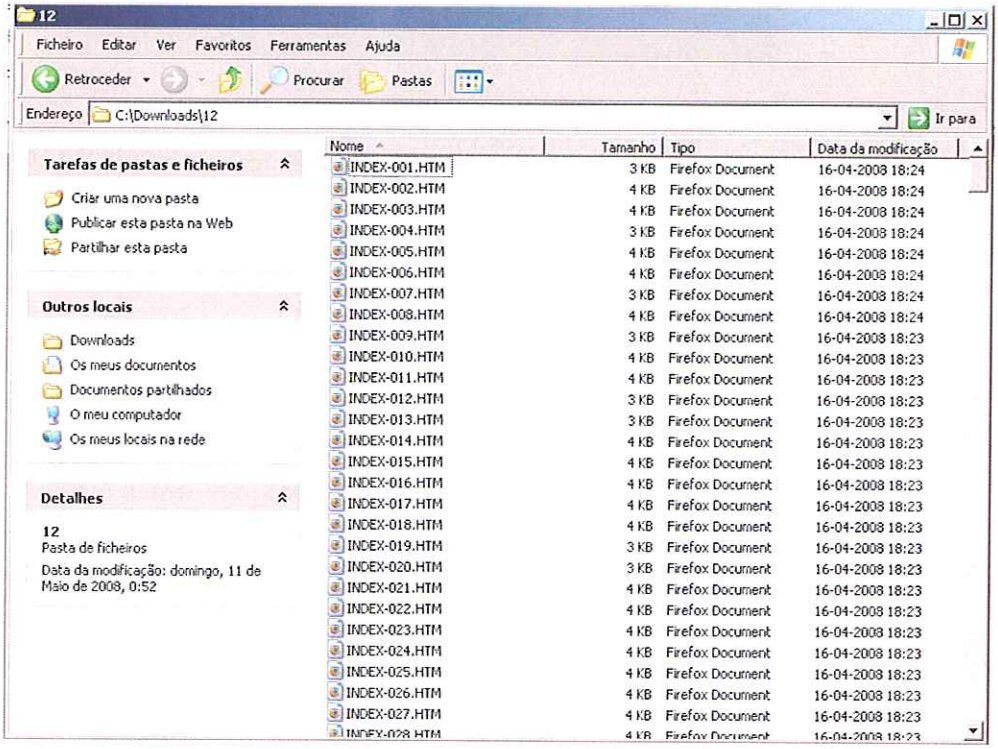


Fig 17: Folder with html files of the table history

After this step 3 will be complete.

4.3 - Joining the Hands

To join all the html files in one the software File Fuse [File Fuse, 2008] was used. This program enables to join a group of files in one single file. The program interface can be seen in figure 18.

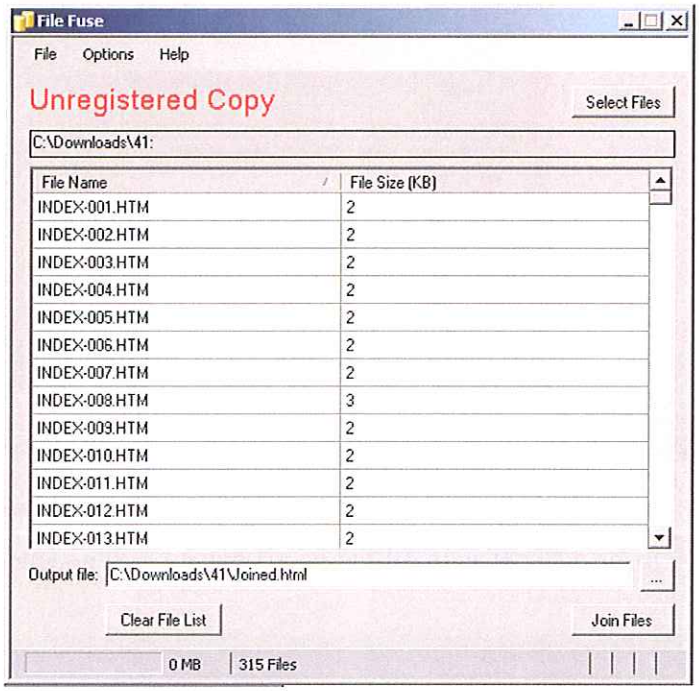


Fig 18: Joining the html files

After all the files are joined it is necessary to edit with windows notepad (or any other editor) the resulting file and remove all </html>. Otherwise any html editor would consider </html> as the end of the file. After this process, step 4 will be complete.

4.4 - Converting from Html to Text

It is necessary to convert from html to txt in order to remove all html coding. To do this Microsoft Word 2003 was used. There is a bug in the program which makes it necessary to do the mapping of the pages before converting, otherwise the output will be scrambled, to do this after opening the program it is needed to scroll down and wait a few seconds. To convert the file the process consists in saving the file using the "save as a txt file" option. The procedure and the options selected can be seen in figure 19.

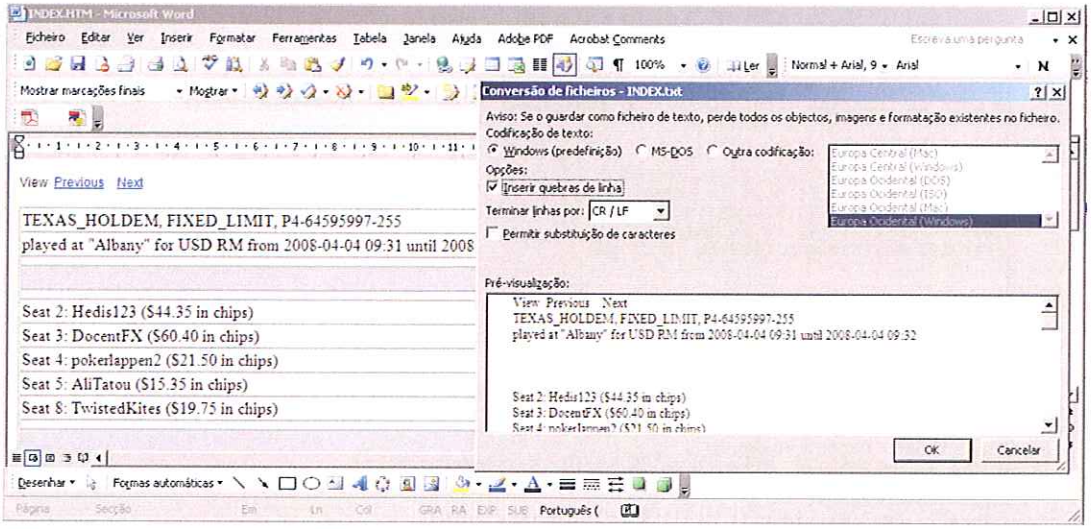
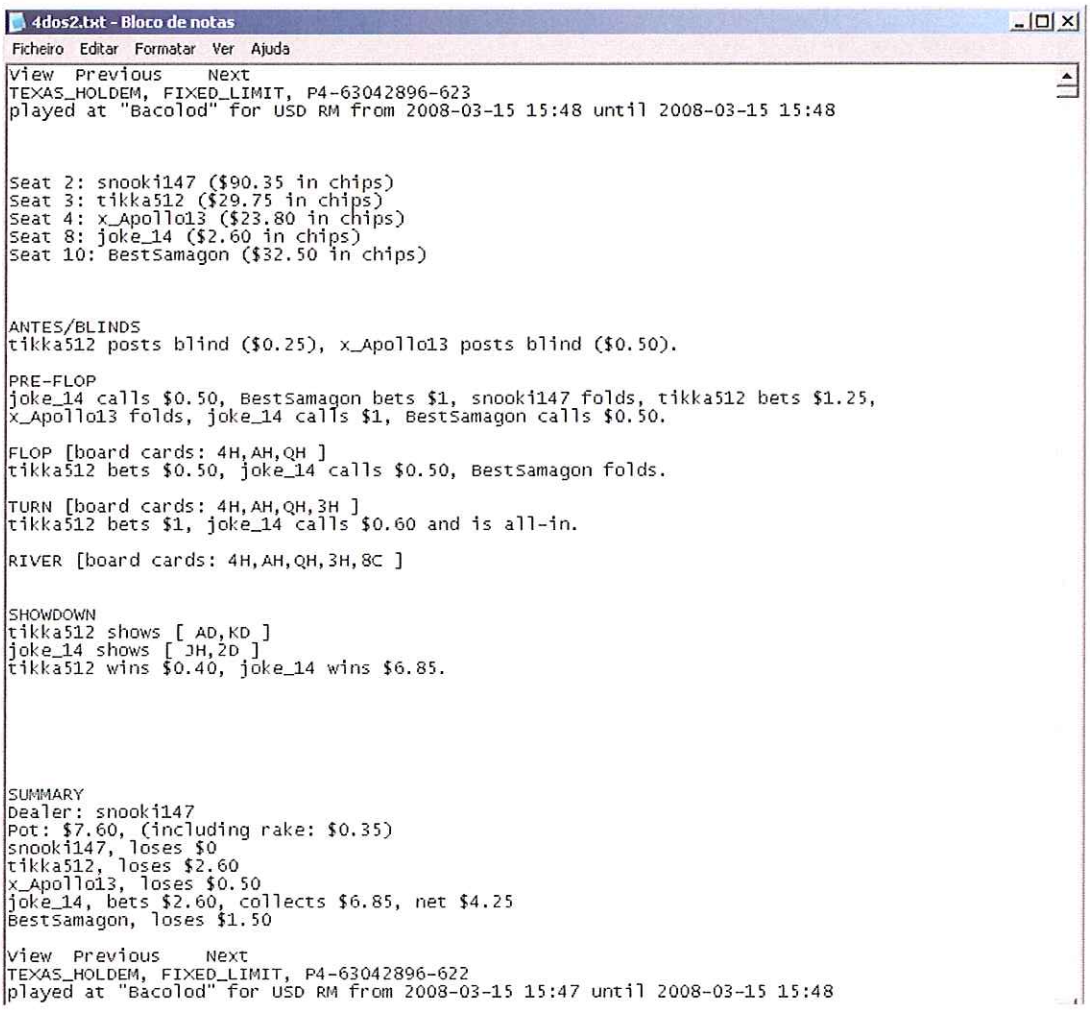


Fig 19: Converting html to txt

The result will be a txt file with all the hands, each hand starts with "View". The result can be seen in figure 20.



4dos2.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
View  Previous     Next
TEXAS_HOLDEM, FIXED_LIMIT, P4-63042896-623
played at "Bacolod" for USD RM from 2008-03-15 15:48 until 2008-03-15 15:48



Seat 2: snooki147 ($90.35 in chips)
Seat 3: tikka512 ($29.75 in chips)
Seat 4: x_Apollo13 ($23.80 in chips)
Seat 8: joke_14 ($2.60 in chips)
Seat 10: BestSamagon ($32.50 in chips)



ANTES/BLINDS
tikka512 posts blind ($0.25), x_Apollo13 posts blind ($0.50).

PRE-FLOP
joke_14 calls $0.50, BestSamagon bets $1, snooki147 folds, tikka512 bets $1.25,
x_Apollo13 folds, joke_14 calls $1, BestSamagon calls $0.50.

FLOP [board cards: 4H,AH,QH ]
tikka512 bets $0.50, joke_14 calls $0.50, BestSamagon folds.

TURN [board cards: 4H,AH,QH,3H ]
tikka512 bets $1, joke_14 calls $0.60 and is all-in.

RIVER [board cards: 4H,AH,QH,3H,8C ]


SHOWDOWN
tikka512 shows [ AD,KD ]
joke_14 shows [ JH,2D ]
tikka512 wins $0.40, joke_14 wins $6.85.




SUMMARY
Dealer: snooki147
Pot: $7.60, (including rake: $0.35)
snooki147, loses $0
tikka512, loses $2.60
x_Apollo13, loses $0.50
joke_14, bets $2.60, collects $6.85, net $4.25
BestSamagon, loses $1.50

View  Previous     Next
TEXAS_HOLDEM, FIXED_LIMIT, P4-63042896-622
played at "Bacolod" for USD RM from 2008-03-15 15:47 until 2008-03-15 15:48
```

Fig 20 –Final Result - converted BetAndWin game into a Txt file

After this step 5 will be complete. The txt file contains all the games that were played at the selected table.

4.5 - Converting from Text to a Database

The structure created for storing the information of a hand is quite simple but contains all the information of the hand. Using BNF – Bakus Naur Form, it is possible to represent the format of the hand as:

<HAND>::= <ID>
<PLAYERS>
<PLAYERS_MONEY>
<PLAYERS_CARD>
<PLAYERS_BETS>
<PRE-FLOP>

<FLOP>
<TURN>
<RIVER
<SHOWDOWN>
<WINER>
<POTMONEY>

<ID>::= [String] Unique for each game

<PLAYERS>::=[String [5] Players names, by order of position, being player 0 playing the small blind and player 1 the big blind and so on.

<PLAYERS_CARD>::=[String [5] Players Cards

<PLAYERS_MONEY>::=[Float [5] Players money before the hand

<PLAYERS_BETS>::=[Float [5]] Players wins\losses of the hand

<PRE-FLOP>::=[short int [5][5]] Actions of the players in the pre-flop being 1 fold, 2 call, 3 check, 4 raise. The first indices corresponded to the player, the second to the order of the action being 0 the first, 1 de second. It's considered that the max of actions are 5.

<FLOP>::=[short int [5][5]]

<TURN>::=[short int [5][5]]

<RIVER>::=[short int [5][5]]

<SHOWDOWN>::=[bool] if the game ended with showdown or not

<WINER>::=[short int] corresponds to the number of the player you won

<POTMONEY>::=[float] the money in the pot

In order to extract the information from the txt to the database, a small program in C++ was designed. With no special requirement any C++ Compiler would be good enough. For this project Borland C++ was used. The reason Borland C++ was chosen is because it is free and has a powerful debug utility included. After this all the downloading and converting steps will be complete.

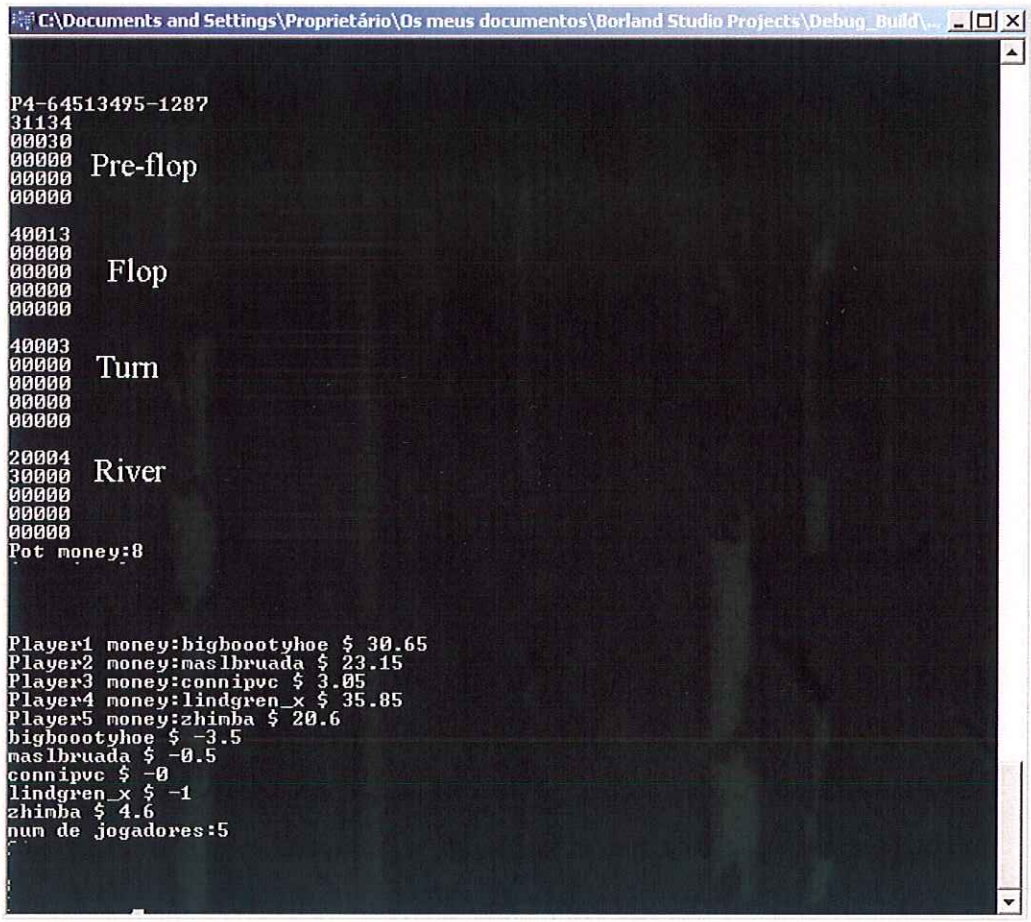


Fig 21: Game in Database

Figure 21 shows the result of a printscreen of a hand in the database. The first player is the small blind the second the big blind. For the first actions 31134 should be read 13431 this means 1 players folds, second calls, third raise, small blind calls, and then second player calls. In the flop it's only possible to see the actions of the remaining players. In the flop the small blind(sb) raises, the second player folds and the third calls and so on. Its possible to see that zhimba won 4.6$. It appears like player five because player 1 is sb and player 2 is bb.

4.6 - Conclusions

The procedure to collect data adopted for this project is simple but demands 1 hour per 3.000 hands, which makes it quite time consuming. The procedure worked and after 6 weeks 40.000 hands were collected.

Repeating this procedure in other thesis would not be wise, as it consumes a lot of time. One possible option would be contact other Universities, namely Alberta and ask for their poker database. This would be a great help to any poker research done in FEUP, as the researchers would have more time to concentrate in the thesis itself.

Chapter 5

5 Game Models

After the extraction of the Hands from the internet it is necessary to classify them. The history of past games is commonly used to build a counterstrategy in order to explore the opponents to use all Hands as equals and produce the best counterstrategy against them would not be wise. The best counterstrategy against the general of the players/table, is not necessarily the best counterstrategy against a particular player/table. As there is no best static strategy in poker, it is widely accepted that the best strategy is one that adjust better to a particular situation of the game. Because of that is needed to classify the hands joining the hands which theoretical the counterstrategy would be similar. In head's up poker, past programs like Vexbot usually discard old hands from the record in order to adjust to opponents changing strategies. The opponents choices were kept in a tree shaped database were each nod is influenced by the player's cards and probability's of choice.

This thesis analyse, for multiplayer tables, if different situations of the game influence the player's tactics. This way it will be possible to provide different counterstrategies against different situations. The player's actions are kept in the same format that was used for the head's up poker.

5.1 - Theoretical Game Model

The theoretical Game Model is an abstraction that considers all background and presents situations of an opponent is relevant to his strategy. So it is assumed that everything will influence the opponent, and the opponent will have a different strategy for each situation. This means that the opponent is affected by the duration of the game, opponents, money, and current table status. This model will go into details assuming that a player will have a different strategy if a particular player has less or more money and even his recent losses\wins.

<GAME>::= //A Game will consist of the table information, the player's information and the history
<TABLE>
<PLAYERS>
<HISTORY>

<TABLE>::= //The table information will consist in the number of players the blind the fold ratio the average pot and the time is been playing
<NUM_PLAYERS>
<BLIND>
<FOLD_RATIO>
<AVERAGE POT>
<TIME>

<NUM_PLAYERS>::= [integer]

<BIND>::= [integer]

<FOLD_RATIO>::= [PERCENTAGE]

<AVERAGE POT>::= [integer]

<TIME>::= [integer] //periods of 5 minutes

<PLAYERS>::= {<PLAYER>} // Players is a set of N individual Players

<PLAYER> // Each player is defined by his name, current money situation and recent money losses/wins
<NAME>
<M>
<MONEY_FLOTATION>

<NAME>::=[STRING]

<M>::= Green Zone | Yellow Zone | Red Zone

<MONEY_FLOTATION>::= [float]

<HISTORY>::= // the history will be defined by the game information and by the actions that occur
<GAME>
<ACTIONS>

<GAME>::= // for the game I is necessary to know the starting player, the players and board cards
<STARTING_PLAYER>
<PLAYER_CARD_STRENGHTS>
<BOARD_CARD_STRENGHT>

<STARTING_PLAYER>::= [int]

<PLAYER_CARD_STRENGHTS>::= // using the same process as Alberta in each phase instead of keeping the value of the cards it is measure the strength
{ <PLAYER_CARD_STRENGHT> }

<PLAYER_CARD_STRENGHT>::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<ACTIONS>::= // here all the actions of the game are kept
<PRE-FLOP>
<FLOP>
<TURN>
<RIVER>

<PRE-FLOP>::=[MATRIX INT SIZE [3][3][3][3][3][3][3][3][3][3][3][3][3][3][3]]

<FLOP>::=[MATRIX INT SIZE [3][3][3][3][3][3][3][3][3][3][3][3][3][3][3]]

<TURN>::=[MATRIX INT SIZE [3][3][3][3][3][3][3][3][3][3][3][3][3][3][3]]

<RIVER>::=[MATRIX BOOL SIZE [3][3][3][3][3][3][3][3][3][3][3][3][3][3][3]]

Although assuming that players can be affected by all environments is logical the model generated by that thought becomes unpractical. There is not enough data to build or support so detailed model. Also there is the problem of the exponential effect of the game tree. In multiplayer poker games the number of decision nods becomes too large to be suited in such structure. While in heads up a normal game goes until 3^15 decision nods, in multiplayer the number can go up to 3^50.

5.2 - Simplified Model

In order to produce measurable results some simplifications to the theoretical model were made. The simplified model is a subset of the general model, keeping the main structure of the theoretical model, so any results and conclusions gathered for the subset will be also true in the general model.

For this thesis it was decided to analyse the player's strategy in the pre-flop. As the pre-flop is the basis of any strategy a change in the pre-flop strategy translates in a change of the strategy. It was analysed quantity the changes of a player strategy due to:

- *Position;*
- *Number of players;*
- *Other players actions;*
- *Money:*

The actions of the players in the pre-flop are kept using the same tree structure that was used in head's up poker. As only the pre-flop is considered it is avoided the exponential effect of the choice nods.

<GAME>::= <TABLE> <PLAYERS> <HISTORY>

<TABLE>::= <NUM_PLAYERS> <BLIND> <FOLD_RATIO> <AVERAGE POT>

<NUM_PLAYERS>::= [integer]

<BIND>::= [integer]

<FOLD_RATIO>::= [PERCENTAGE]

<AVERAGE POT>::= [integer]

<PLAYERS>::= {<PLAYER>}

<PLAYER>::= <NAME> <M>

<NAME>::=[STRING]

<M>::= Green Zone | Yellow Zone | Red Zone

<HISTORY>::= <GAME> <ACTIONS>

<GAME>::= <STARTING_PLAYER> <PLAYER_CARD_STRENGHTS>

<STARTING_PLAYER>::= [int]

<PLAYER_CARD_STRENGHTS>::= {<PLAYER_CARD_STRENGHT>}

<PLAYER_CARD_STRENGHT>::= 1 | 2 | 3 | 4 | 5

<ACTIONS>::= <PRE-FLOP> // only the actions of the pre-flop is considered

<PRE-FLOP>::=[short int [3][3][3][3][3][3][3][3][3][3][3][3][3][3][3]]

5.3 - Model Validation

As was stated before, it is generally accepted that players are influenced by a large number of factors. In the simplified model it was considered that players are affected by:

- *Position;*
- *Number of players;*
- *Other players actions;*
- *Money.*

To validate this theory it is necessary to prove that none of these parameters is redundant. For that purpose each factor will be isolated and measure the each one individual impact on the player's pre-flop strategies.

5.3.1 - Search Tree algorithm

In order to perform some tests it was needed to develop the search tree algorithm in C++. The compiler Borland C++ was used due to powerful debug tools and free license. This algorithm will allow analysing the game the following aspects:

1) Average Behaviour – the average behaviour for a determine position.
2) Change of Behaviour – will measure the behaviour of the player after a specific action.
3) Impact Measure – will measure the impact of a specific action.

In order to keep the results free of noise, all user actions were isolated from the rest of the actions of the other users. So it was compared exact same situations changing only the action of one user.

Figure 22 represents the beginning of the tree, in this representation it was ignored the first nod that represents which player started. So instead of the tree being tree[5][3][3]... this tree is tree[3][3][3]. The first player has 3 possible actions, fold call or raise. All possible outcomes of the pre-fold actions are covered. Every time a sequence occurs it is incremented the number of times it happened. So let's imagine that there is sequence fold folds fold fold call, this would be tree[1][1][1][1][2]++. So for example if we want to measure the player one fold ratio, it is necessary to visit the entire tree till the end and count the outcomes were player one folded and the ones he didn't fold.

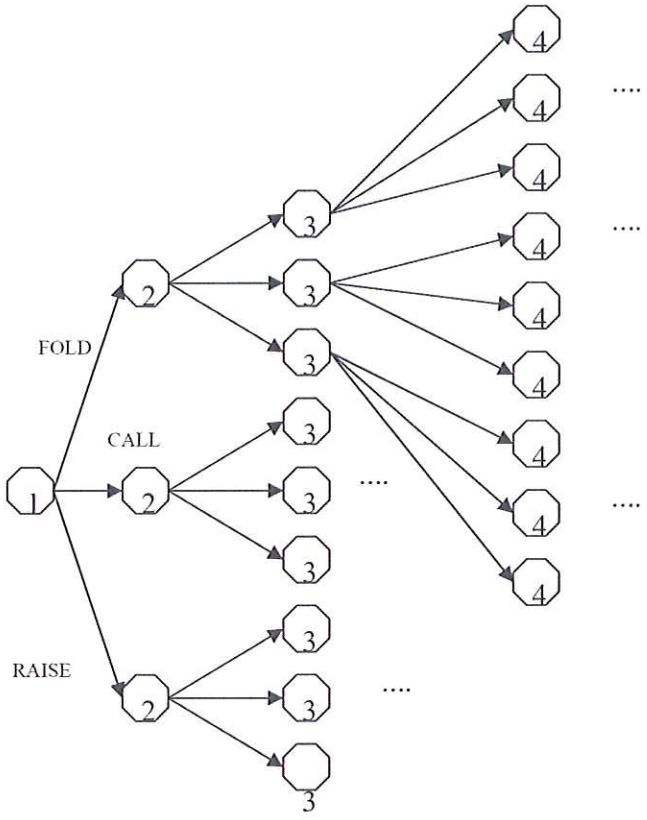


Fig 22: Tree structure

The search tree algorithm is a recursive function that visits all nodes and retrieves information from all the end nodes to the first one.

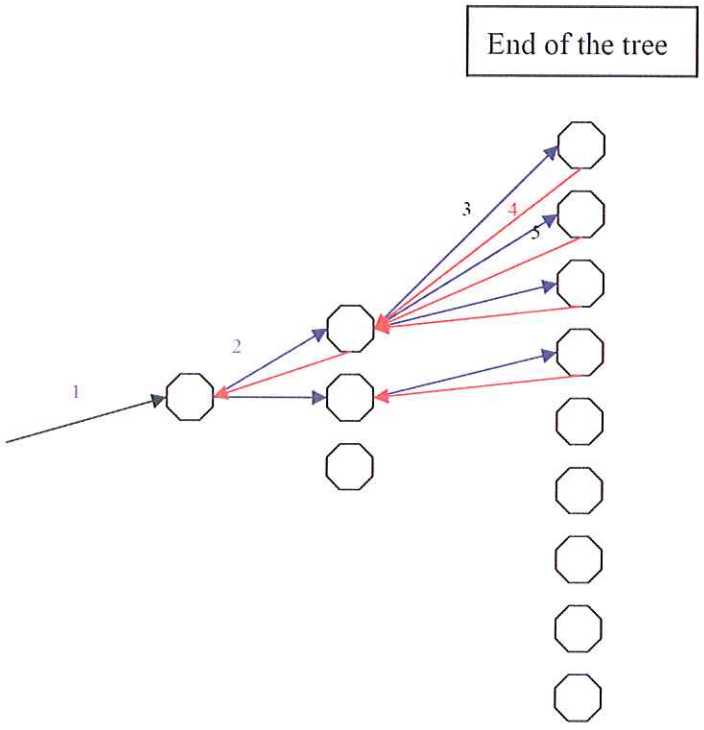


Fig 23: Search tree algorithm reaching the last branch

Figure 23 shows the algorithm reaching the last branch of the tree, analysing the information and retrieving it to the previous branch. When the branch has all the information regarding the forwarding branches retrieves its own information to the previous branch. This procedure repeats until the first branch is reached.

The algorithm does not treat every branch equally; as it was designed to retrieve data from the tree it is necessary to make different actions in different levels. For that purpose the search tree algorithm has 5 levels.

1) The first level corresponds to selecting the position of the players. So, for example, we can select the player "Paul" as the big blind.
2) The second level corresponds to the places before the actions that are being analysed. For example if it is requested to measure the effect of player 3 raising. Level 2 would be the actions of player 1 and 2.
3) The third level corresponds to the branch that corresponds to the action we want to analyse. If we want to analgise the actions of player 3 it would be the third branch.

4) The fourth level would correspond to the branches after the action in which we have to measure some facts. It was decided only to analyse the first 5 actions so it would correspond to the left branch from the action to 5.

5) Level five corresponds to the branches after the first 5 branches. In these branches we only collect data and no math is done.

In the algorithm it's possible to see clearly the 5 levels. The small sample of the code shows it.

```
1 level selecting the player
if(place.branchposition==0)
   { …
      newplace.branch[place.branchposition]=newplace.player;
                                                  //selecting the player
   }
2 level before the action
if(place.branchposition < min(place.posic1,place.posic2))
   { …
    }
3 level at the position of the action
if(place.branchposition==place.posic1)
   { …
    }
4 level after the position until branch 5
for(int i= actionpossition;i<5;i++)
   { …
    }
5 level after branch 5
if(place.branchposition > 5 && place.branchposition < 11)
   { …
    }
```

Level 1

In the first level the algorithm is initiated. The search tree algorithm is an algorithm that calls itself and sends information for one branch to another. As many variables were needed to be send two structures were created. The structures to send and to receive were created. In the code below it's possible to see both of them with a breath explanation.

```
struct send
{
   int branch[12];
   int branchposition;
   int posic1;
   int posic2;
   int value1;
   int value2;
   int player;
};
/* The first and second itens branch and branchposition carry the information
of the position of the nod in the tree.
The third and fourd are the deeth of the branch we want to analise for example
player 2 and player 3.
The values are the action so like for example player 2 folds and player 1
raises.
The player is the position of the game like paul at big blind */

struct receive
{
  double numberofplaces;
  double newfold[5];
  double newcall[5];
  double newraise[5];
  double diferencefold[5];
  double diferencecall[5];
  double diferenceraise[5];
  double auxiliarwatched[8];
};
/* The number of places it the number the node has been visited for example we
are the nod 1 player fold and second raise and we have the value 400. This
means that in all the games objserved 400 times the 1 player folded and the
second raised. If the observed 800 games this would mean 50% of the times.
The new fold call and raise are the new averages after the action of the
player. For example we want to analise the effect of player 2 folding, we are
going to analise the tree in the case of player 2 folding. New fold call and
raise will contain the new values.
The diference valoues instead of the averages will measure the impact after
the action.*/
```

In level 1 the algorithm will select the branch with the position desired. It corresponds to the place were the Big Blind is seated. The procedure is quite simple. And can be seen in the pseudo code.

```
    if(place.branchposition==0)
    {
        newplace.branchposition++;
        newplace.branch[ place.branchposition ]=newplace.player; // we put the
//information of the player
        receive playerx; // reserve memory to receive the results
        playerx = searchtreealgorithm(newplace); // call for the next branch
        return playerx; // return the results
    }
```

Level 2

Level 2 is the level before the actions of players we selected. We can divide level 2 in two parts. The part that we call the next branches and the part where we receive the information from the branches.

The part were we call the new branches can be seen in the pseudo code below.

```
    {
        receive toreturn;
        newplace.branchposition++;
        newplace.branch[ place.branchposition ]=0;
        receive fold;
        fold = searchtreealgorithm(newplace); /* call next branch that
corresponds to this player folding */
        newplace.branch[ place.branchposition ]=1;
        receive call;
        call = searchtreealgorithm(newplace); /* call next branch that
corresponds to this player call*/
        newplace.branch[ place.branchposition ]=2;
        receive raise;
        raise = searchtreealgorithm(newplace); /* call next branch that
corresponds to this player raising */
}
```

The part were the information is received is basically joining the information from the branches. The pseudo code can be seen below.

```
toreturn.numberofplaces = raise.numberofplaces +call.numberofplaces +
fold.numberofplaces; // the sum all places of this nods will be the sum of all
the branches.
        if(newplace.branchposition<6 && toreturn.numberofplaces != 0 )
        {
            toreturn.newfold[i] = (raise.newfold[i]*raise.numberofplaces +
call.newfold[i]*call.numberofplaces +  fold.newfold[i]*fold.numberofplaces) /
toreturn.numberofplaces; //joins the results of the branchs
```

```
        //the same for
        toreturn.newcall[i]
        toreturn.newraise[i]
        toreturn.diferencefold[i]
        toreturn.diferencecall[i]
    }
Return toreturn //return the results of this branch
```

Level 3

In level three it is selected the branches that correspond to the actions we desire. The procedure can be seen in pseudo code below.

```
if(place.branchposition==place.posic2)
{
    newplace.branchposition++;
    newplace.branch[ place.branchposition ]=place.value2; /*selected the
action*/
    receive toreturn;
    toreturn = searchtreealgorithm(newplace); /* calls for all branchs that
derive from this action*/
    return toreturn;
}
```

Level 4

The actions on level 4 are quite similar to the ones in level 2 the only difference is in level 2 when the results are received it is necessary to measure for the first time the folding ratios of the position. So for example if we are in branch 3 we calculate for the first time the actions of player 3 but also we combine the results gathered so far of player 4,5. The following pseudo code shows how the procedure works.

```
    for(int i= place.branchposition +1; i<5; i++)
        {
            ..
        Like in level 2
            ..
        }
         // FOR I = place.branchposition - The previous calculation were the
update of caltulation of the information provided by the branchs of call, fold
and raise. This calculation is the first one and is the information of the
action of the player at position I, so for exemple if I = 3 we are going to
measure the statitcs of player 3, in branches i=4.5.6… this was not possible
        toreturn.newfold[place.branchposition-1] =  fold.numberofplaces /
toreturn.numberofplaces;
        toreturn.newcall[place.branchposition-1] =  call.numberofplaces /
toreturn.numberofplaces;
```

```
        toreturn.newraise[place.branchposition-1] =  raise.numberofplaces /
toreturn.numberofplaces;
      }
```

Level 5

In level five because that to measure the impact of one player action it is needed to measure the impact if the player had taken another action, it is necessary to call the branches were he took different actions and gather the information. So not only in level 5 we call the actions of the player of the nod, we also call all the other branches were the player took different actions. The procedure can be seen in pseudo code below.

```
if(place.branchposition==6)
    {
       int d=0;
       receive callauxiliar;
       receive foldauxiliar;
       receive raiseauxiliar;
       if( newplace.posic1 > 0 && newplace.posic2 < 0)
// this is the procedure if we just analising 1 action
       {
        for(int x=0;x<3;x++)
         {
          if(x != newplace.value1)
           {
           newplace.branch[ newplace.posic1]=x;
           newplace.branch[ place.branchposition ]=0;
           foldauxiliar = searchtreealgorithm(newplace);
           newplace.branch[ place.branchposition ]=1;
           callauxiliar = searchtreealgorithm(newplace);
           newplace.branch[ place.branchposition ]=2;
           raiseauxiliar = searchtreealgorithm(newplace);
           toreturn.auxiliarwatched[d] = raiseauxiliar.numberofplaces +
callauxiliar.numberofplaces + foldauxiliar.numberofplaces;
           }
           d++;
         }
       }
       if( place.posic1 > 0 && place.posic2 > 0)
// this is the procedure for analising 2 actions
       {
        for(int x=0;x<3;x++)
         {
```

```
            for(int y=0;y<3;y++)
              {
             if(x != newplace.value1 && y != newplace.value2 )
                  {
                     newplace.branch[ newplace.posic1]=x;
                     newplace.branch[ newplace.posic2]=y;
                     newplace.branch[ place.branchposition ]=0;
                     foldauxiliar = searchtreealgorithm(newplace);
                     newplace.branch[ place.branchposition ]=1;
                     callauxiliar = searchtreealgorithm(newplace);
                     newplace.branch[ place.branchposition ]=2;
                     raiseauxiliar = searchtreealgorithm(newplace);
                     toreturn.auxiliarwatched[d] =
raiseauxiliar.numberofplaces+callauxiliar.numberofplaces+foldauxiliar.numberof
places;
                  }
               d++;
               }
            }
         }
      }
```

We can see in the pseudo code for each branch there are in the maximum 8 comparable branches. Figure 24 illustrates the procedure.

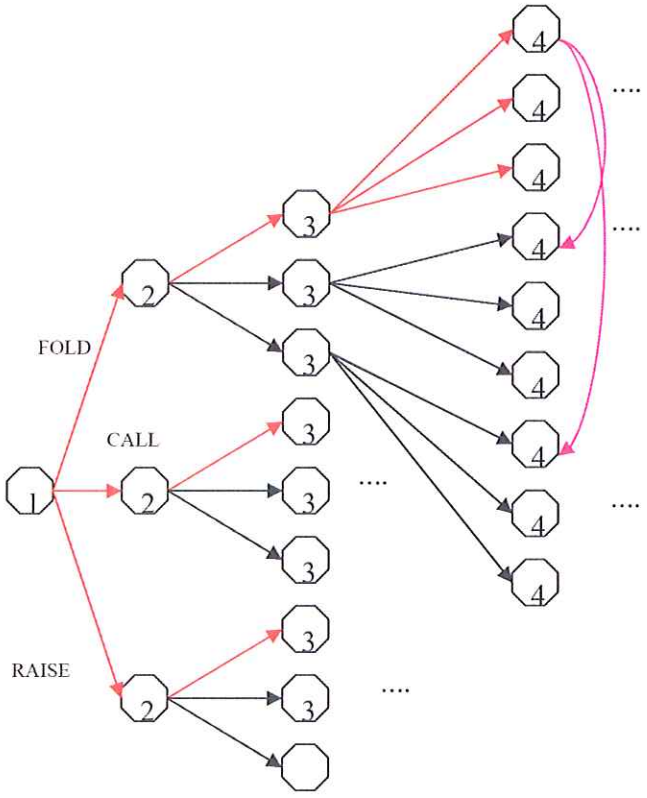


Fig 24: Search tree algorithm measuring the impact.

So we are analysing the action where player 2 folded. In the possible actions of player 4 we are going to compare the exact same situation but where player 2 raised or called, the actions of player 1 and 4 are intact. The results of the auxiliary branches are transmitted throw the algorithm, so we only need to call them once. The procedure for measuring the procedure of the impact can be seen below

```
for(int i=0; i<8; i++)
  tem[i] = toreturn.auxiliarwatched[i] //makes a copy so if less than 50
  occurances can be ignored in this branch but sent to the next
      {
        if( toreturn.auxiliarwatched[i] > 55) /* only compare if there is at
least 50 occurences, the next procedure convest the results to frequences */
        {
          raise.auxiliarwatched[i] = raise.auxiliarwatched[i] /
toreturn.auxiliarwatched[i];
          call.auxiliarwatched[i] …
          fold.auxiliarwatched[i]…
          flag=1;
        }
        else // so less than 50 occurences are ignored
```

```
        {
          temp[i]=0;
          fold.auxiliarwatched[i]=0;
          call.auxiliarwatched[i]=0;
          raise.auxiliarwatched[i]=0;
         }
        if(i==7 && flag==1)
        {
           toreturn.diferenceraise[place.branchposition-1]=
(fabs(toreturn.newraise[place.branchposition-1] -
raise.auxiliarwatched[0])*temp[0]+ ..
raise.auxiliarwatched[6])*temp[6]+fabs(toreturn.newraise[place.branchposition-
1] - raise.auxiliarwatched[7])*temp[7])/( temp[7]+…temp[0]);
/* this means for each diferente occurrence the percentage of the diference is
multiplied by the number of times it happened (more than 50) then the sum of
all diferente is divided by the number of occurences */
          toreturn.diferencecall[place.branchposition-1] …
          toreturn.diferencefold[place.branchposition-1] …
        }
        if(i==7 && flag==0)
        {
          toreturn.diferenceraise[place.branchposition-1]=0;
          toreturn.diferencecall[place.branchposition-1]= 0;
          toreturn.diferencefold[place.branchposition-1]=0 ;
        }
      }
```

Last branch

The procedure for the last branch is quite simple. The number of occurrences is read of the search tree and returned. Some variables of the return are also set to 0.

```
if(place.branchposition==11)
   {
      receive toreturn;
      for (int g=0; g < 5; g++) {
          toreturn.newfold[g]=0;
          toreturn.newcall[g]=0;
          toreturn.newraise[g]=0;
          toreturn.diferencefold[g]=0;
          toreturn.diferencecall[g]=0;
          toreturn.diferenceraise[g]=0;
       }
      toreturn.numberofplaces =
modelofthegame.history.ACTIONSPREFLOP[newplace.branch[0]][newplace.branch[1]][
newplace.branch[2]][newplace.branch[3]][newplace.branch[4]][newplace.branch[5]
```

```
][newplace.branch[6]][newplace.branch[7]][newplace.branch[8]][newplace.branch[
9]][newplace.branch[10]];

      return toreturn;

   }
```

5.3.2 - Testing

In order to determine the pre-flop strategies of the players and how they are influenced, a group of tests were conducted. Each tests isolates of factor from the rest of the game. This procedure guarantees reliable results.

5.3.2.1 - Number of players

To determine if the number of player was relevant, it was observed the general pre-flop behaviour in the tables of 3,4 and 5 players. Also the differences between the tables were considered.

In the image 25 is possible to see the general results for three players.

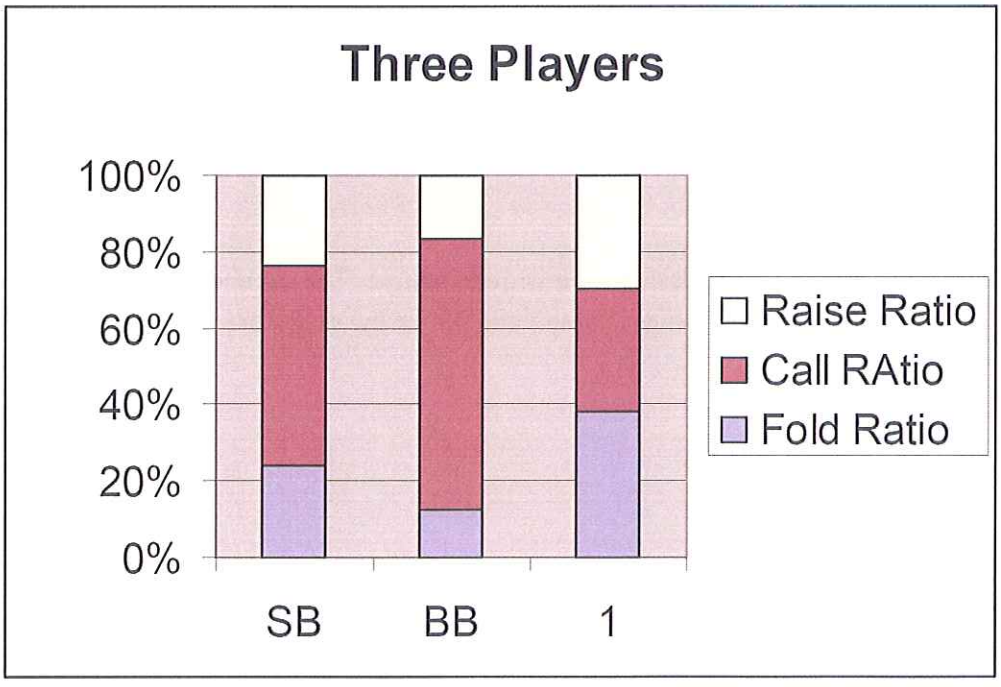


Fig 25: Three players' Pre-flop strategy

Considering the fold ratio the BB is 17,5% the SB is 22,0% and the player 1 38,5%. The player 1 is more aggressive than the rest of the players

In image 26 the results for a table with four players is shown. The fold ratio increases compared with the three player table. The raise ratio decreases in all players. The behaviour of the SB and BB does not change, they continue to have smaller fold ratios and smaller raise ratios.

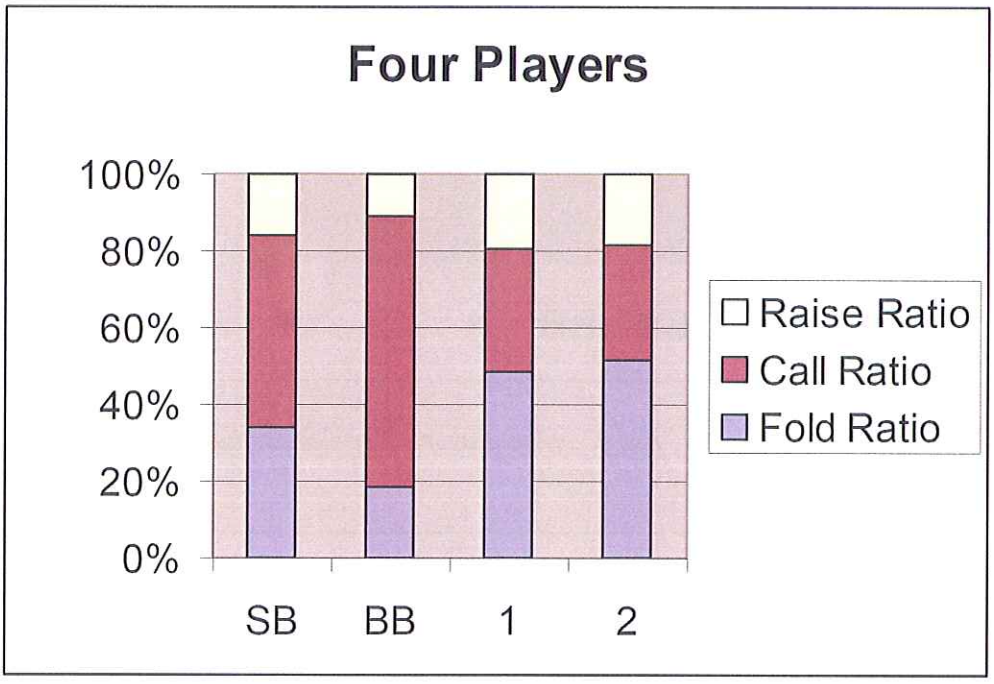


Fig 26: Four player's Pre-flop strategy

In image 27 the results for a five player table is shown. The fold ratios are higher than the three and four table. The behaviour of the SB and BB is kept similar

Fig 27: Five player's Pre-flop strategy

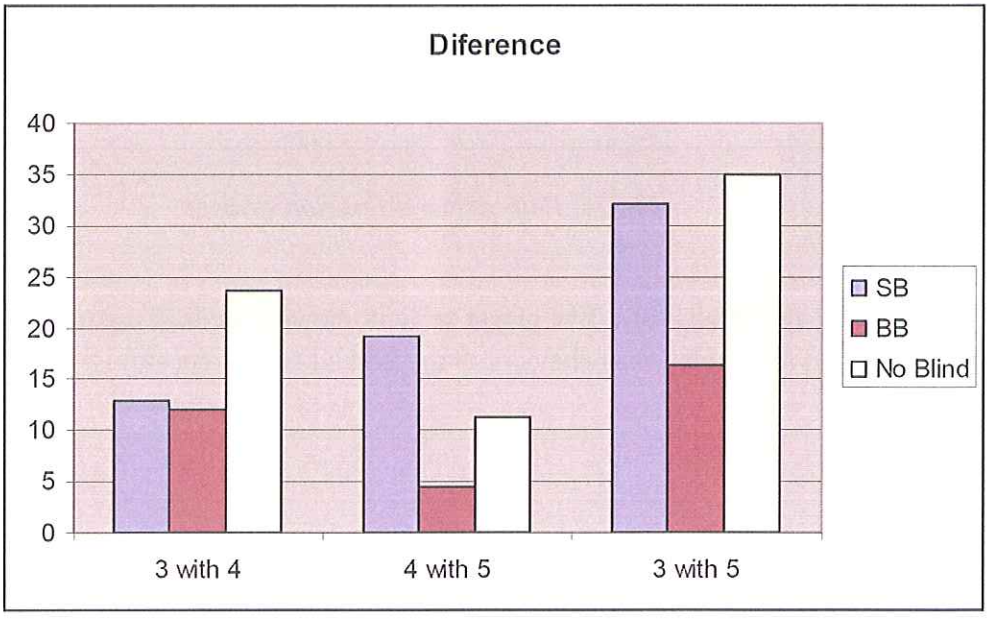


Fig 28: Differences between the pre-flop strategies

Figure 28 illustrates the difference between the strategies caused by the number of players. It is possible to see that the number of player do not affect the same positions equally. The big blind position is the less affected in all the tests with very little difference between four and five player tables. The biggest difference is clearly in the 3

to 5 player table, where very significant changes occur. The sum of the difference of the strategy regarding fold, call and raise reach the order of 35%. The difference between adjacent levels is smaller reaching in the highest 24%

5.3.2.2 - Position

The second tests aimed to determine if the position of the players is relevant for their strategy. That fact can be easily studied in the five player pre-flop diagram.

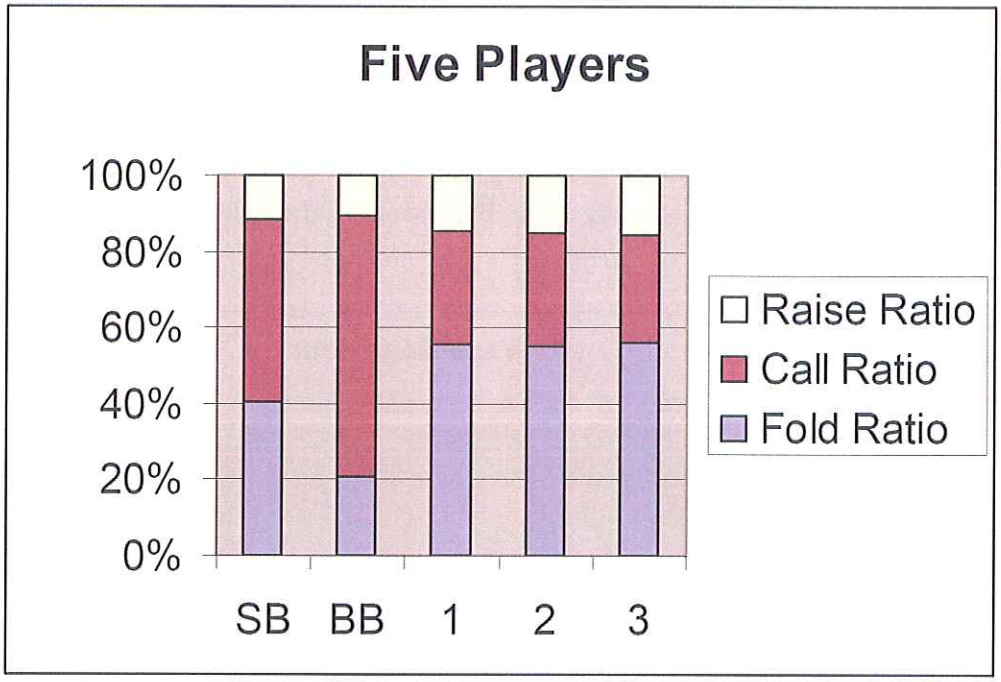


Fig 29: Five players pre-flop strategy

It is clear in figure 29 that big blind and the small blind have different strategies than the rest of the players. The average strategy in the first, second and third player is similar but it is possible to see an increase of aggressively (Raise Ratio) from the first to the third. The differences between positions can reach the order of 35% in the fold ratio

5.3.2.3 - Actions

The third set of tests will determine the importance of the actions of the players. This means that it will be measure that if the actions of player 1 and 2 will be important for the player in the third position. Seven tests using the search tree algorithm are going to be made. It is going to be tested:

- First player Raises

- First player Calls
- First player Folds
- Second player Folds
- Second players raises
- First and second players Fold
- First and Second players Raise

The figure 30 and 31 show the results in the table of the first player raise. The fold ratio increases significantly as the average fold ratio is proximally 50% for player 1 and 2 and 40% and 20% for the SB and BB (figure 29). The impact shows that the must affected positions are the BB and SB, being the fold ratio the must affected. The impact level shows changes of strategy of the BB and SB of the order of 35%

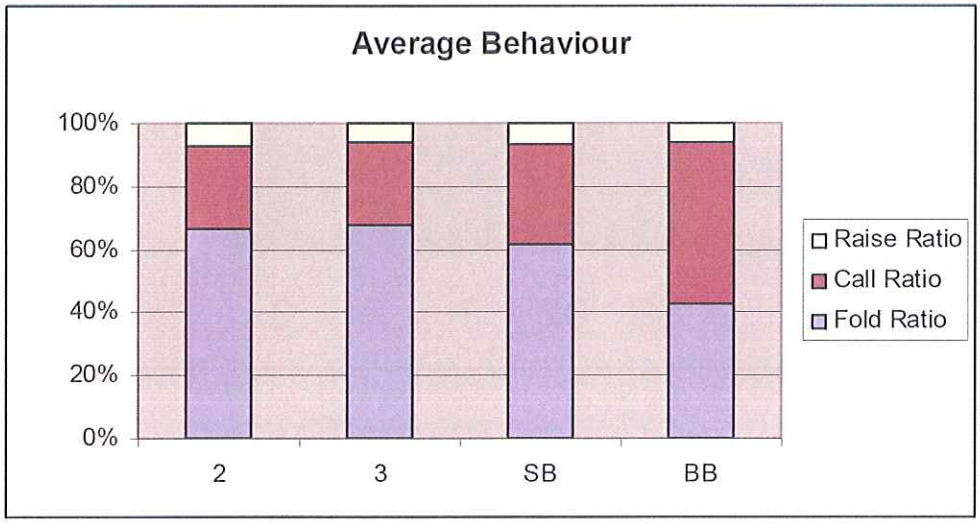


Fig 30: Strategies adopted by the players when first player raises

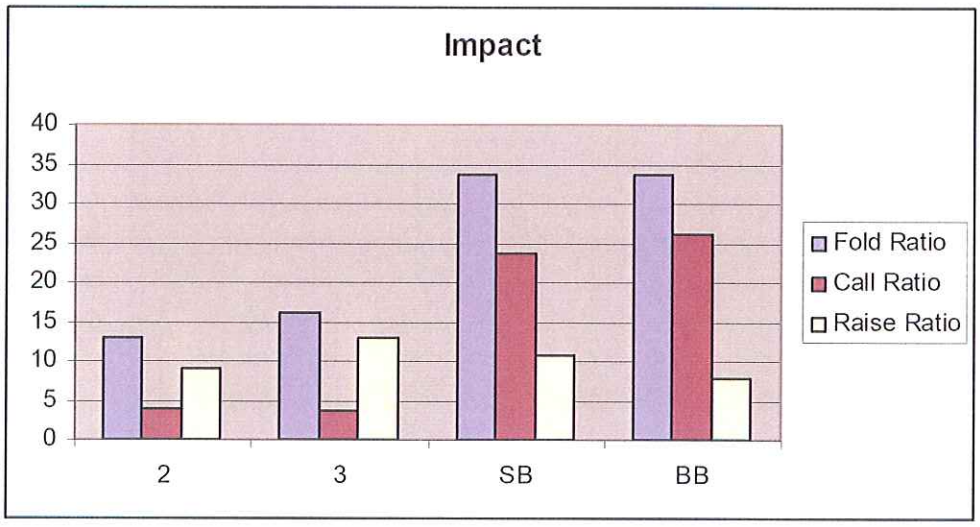


Fig 31: Impact of the first player raise

The figure 32 and 33 show the results in the table of the first player call. The impact of the action call is lower than the action raise. There is a little increase of the fold ratio compared with the average situation. The impact ratio is in average 8-10% for all positions.

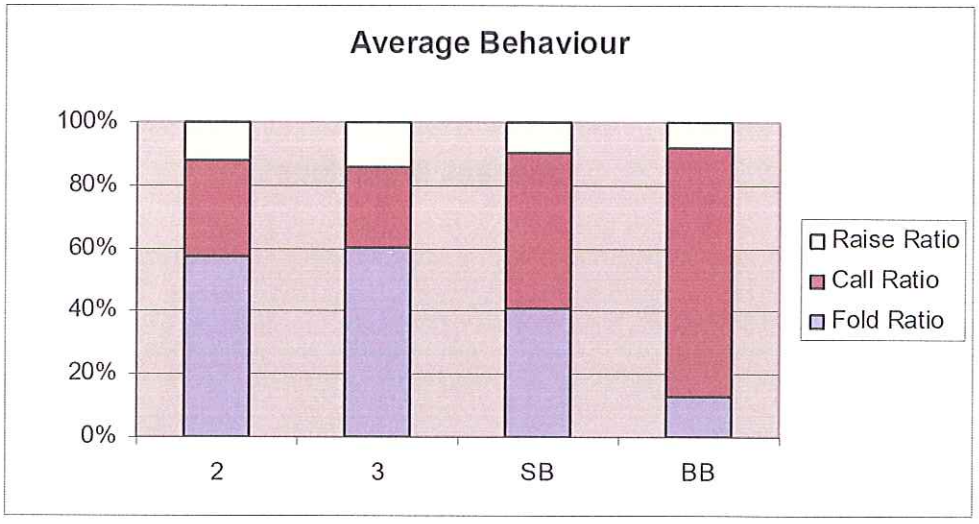


Fig 32: Strategies adopted by the players when first player calls

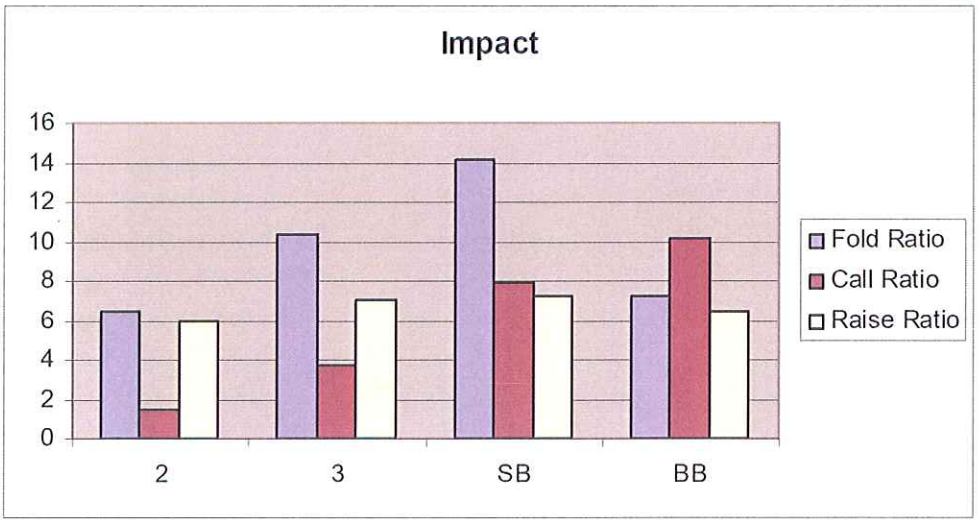


Fig 33: Impact of the first player calls

Figures 34 and 35 show the results in the table of the first player fold. The impact of the action fold is lower than the action raise. The players average behaviour after the action fold, is very similar with the general situation. The actions of the players do in average do not change when the player one folds. While there is no significant change in the average values, the impact figure shows that fold action produces differences in the player's actions. This is very significant as it means although the average results did not change the players change strategies.

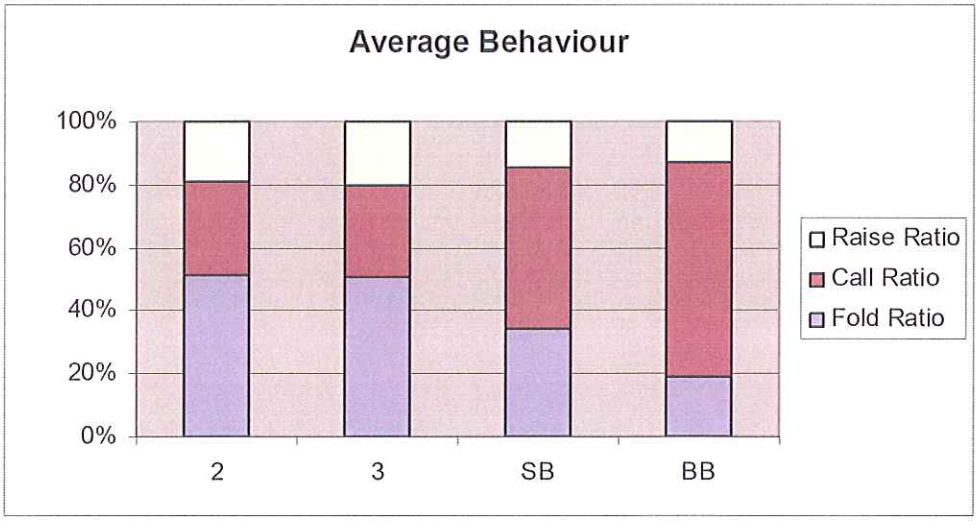


Fig 34: Strategies adopted by the players when first player folds

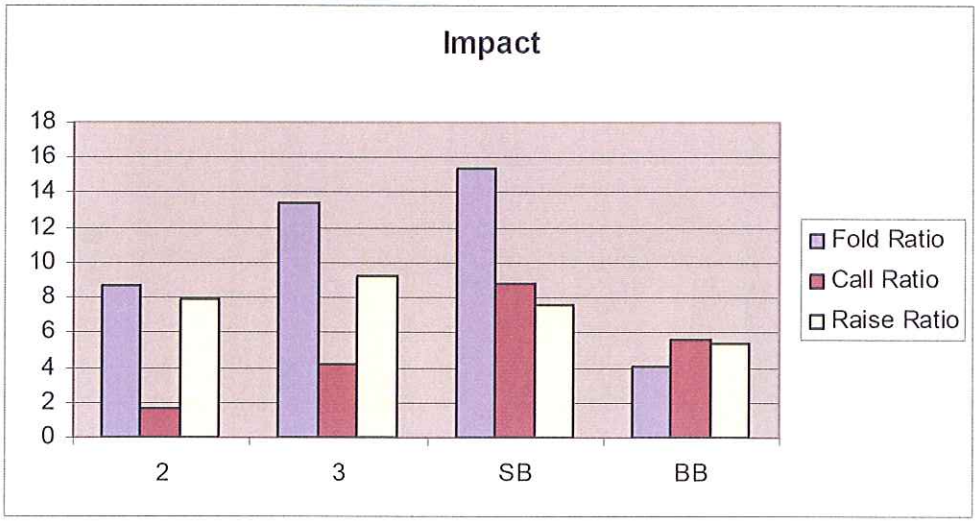


Fig 35: Impact of the first player folds

The figure 36 and 37 show the results in the table of the second player folding independently of the action of player 1. The results of player 2 folding are very similar to the ones of player 1 folding.

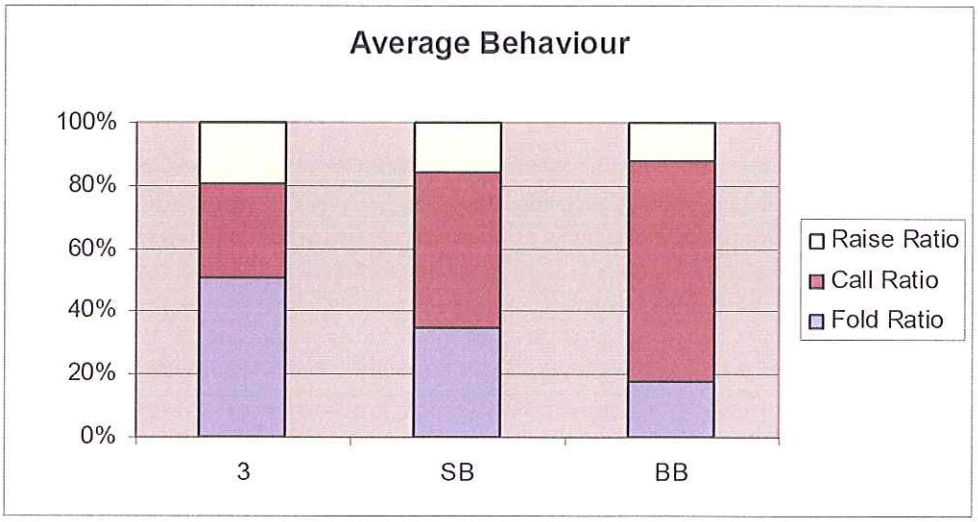


Fig 36: Strategies adopted by the players when second player folds

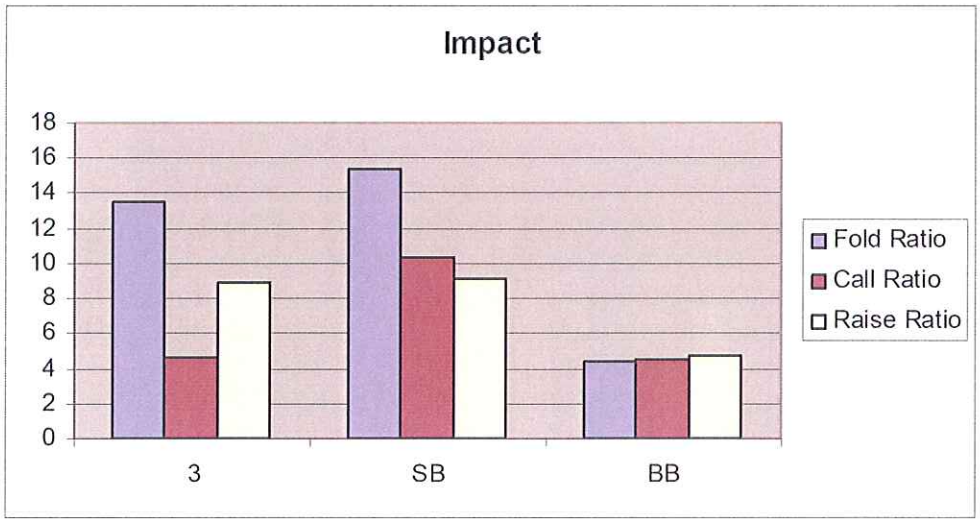


Fig 37: Impact of the second player folds

The figure 38 and 39 show the results in the table of the second player raising independently of the action of player 1. The results of player 2 folding are very similar to the ones of player 1 folding. This demonstrates that impact of actions of the player 1 and 2 are similar.

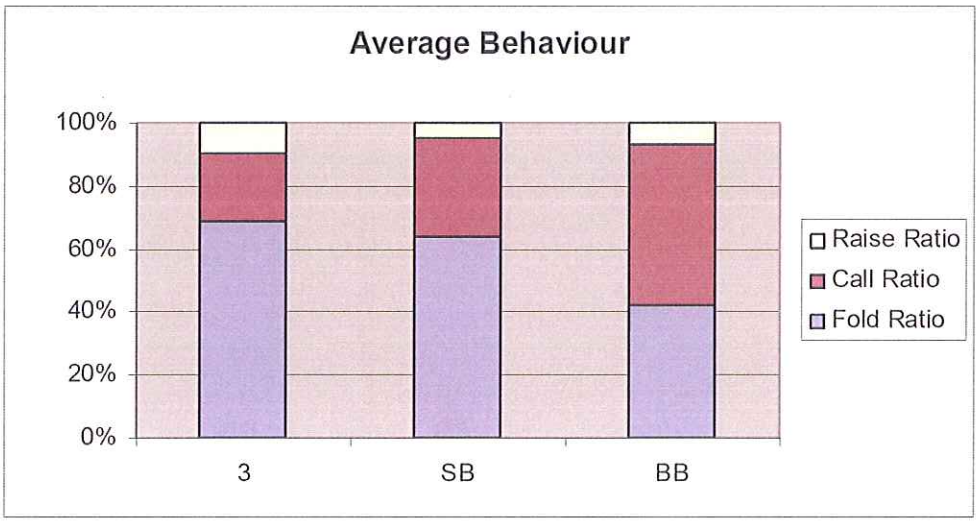


Fig 38: Strategies adopted by the players when second player raises

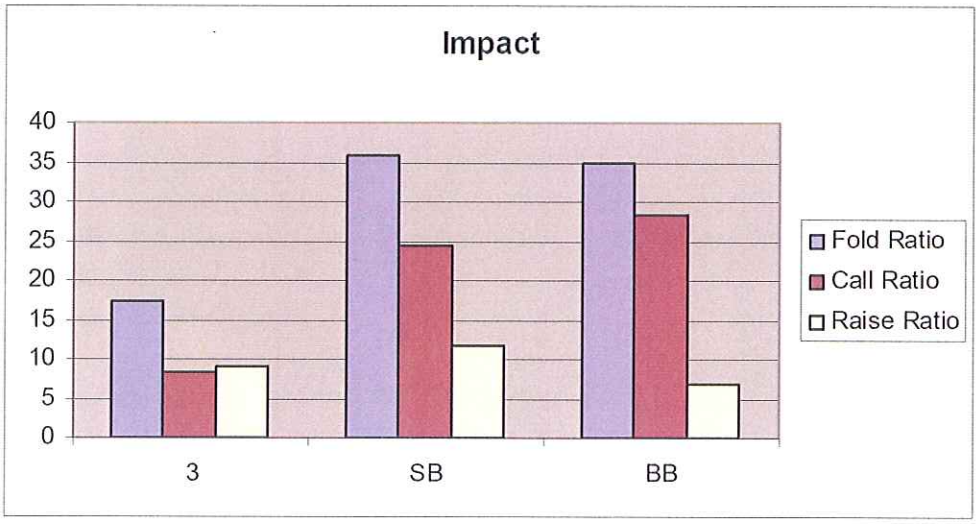


Fig 39: Impact of the second player raises

Now we will initiate so see the impact of two consecutive actions. Figures 41 and 42 show the results of both player 1 and player 2 folding. The fold ratio will decrease. The raise ratio of player three will increase. This is a logical move for player 3 as it forces the SB and BB to pay to see the flop. The impact of player 1 and 2 folding is higher than just one folding, but is still lower than a raise.

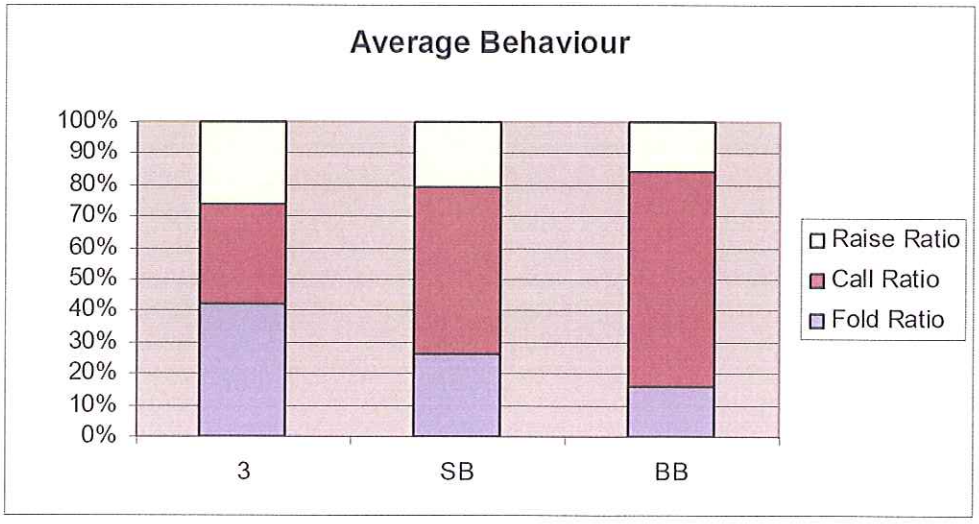


Fig 40: Strategies adopted by the players when first and second player fold

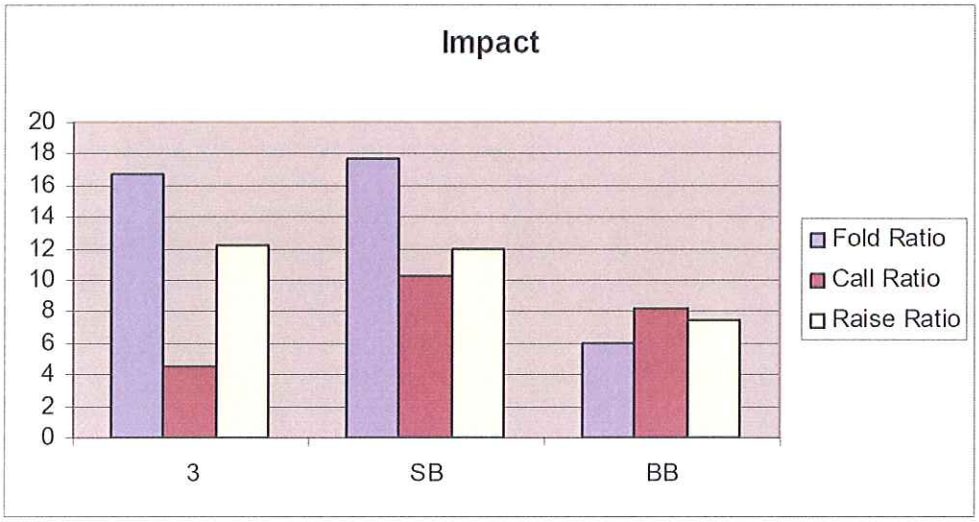


Fig 41: Impact of the first and second player folding

Figure 42 and 43 show the results of both player 1 and player 2 raising. The fold ratio will increase. The raise ratios of player three will disappear. The impact of these consecutive actions is very high and it is higher than a raise from player 1 or player 2.

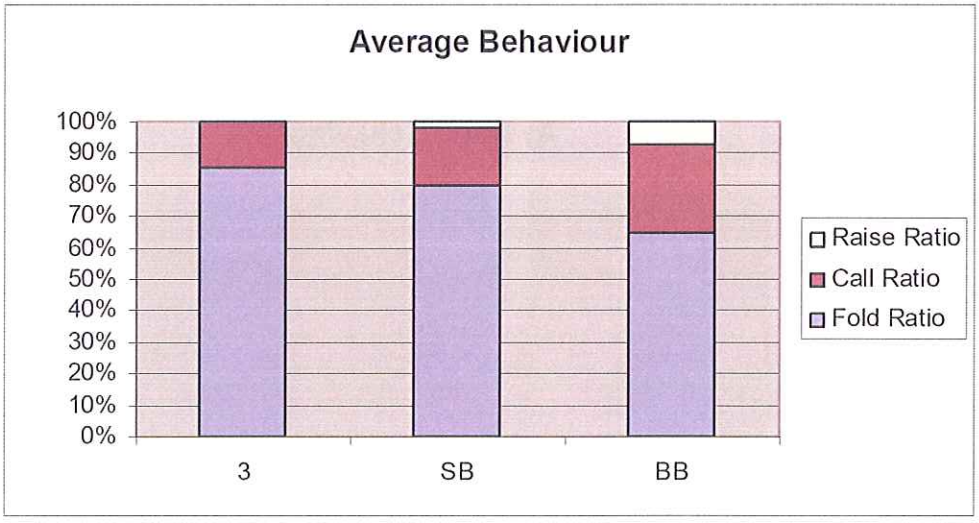


Fig 42: Strategies adopted by the players when first and second player raise

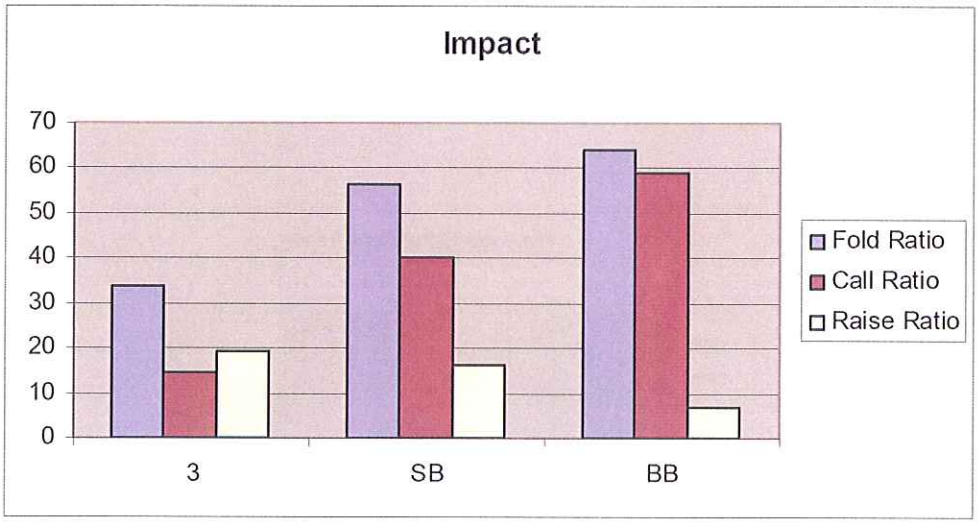


Fig 43: Impact of the first and second player raise

5.3.2.4 - Money

To analyse the impact of the money it was selected three random players and their pre-flop strategy was compared based only on the money.

The first player will be the player John (fig 44). John chooses not to play in the RedZone. In online poker tables many players prefer always to play with a comfortable bankroll. Nevertheless when he is playing in the yellowzone there is a slit change of strategy. He folds less and becomes more aggressive

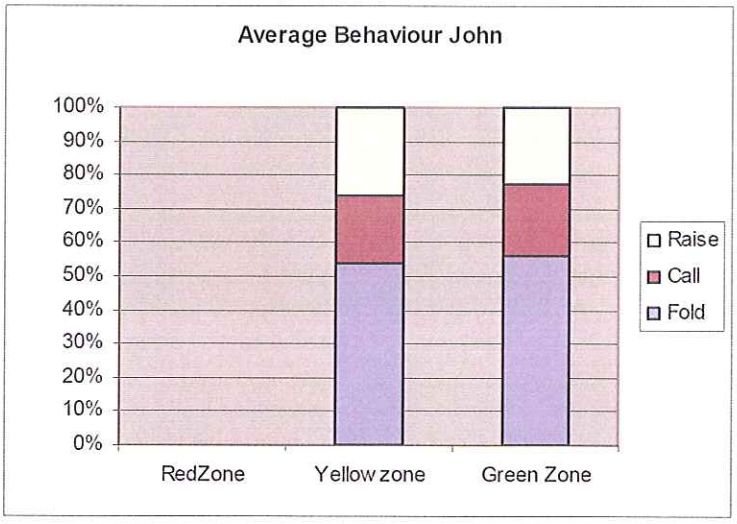


Fig 44: John pre-flop strategy based on Money

The second player will be Peter. Peter becomes tight when he has less money. In the redzone he always folded, this result is not accurate as he only played 10 games in the redzone, not enough games to provide a statistic correct result.

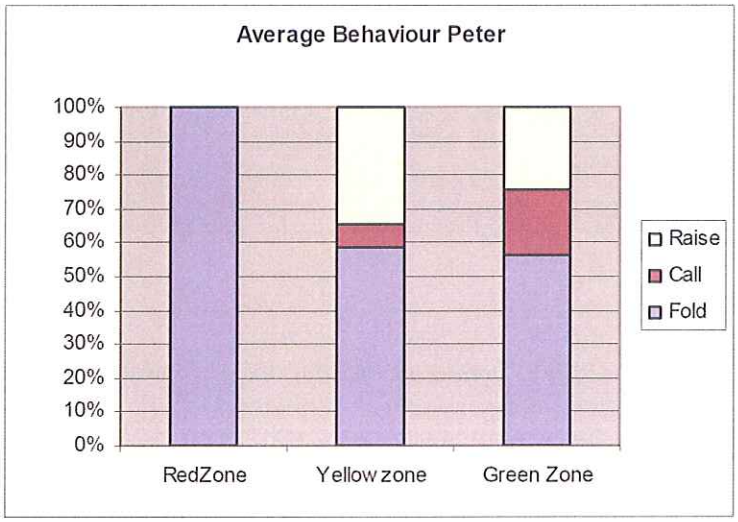


Fig 45: Peter pre-flop strategy based on Money

The last player Fred is a player that plays in all zones. He becomes less tight (fig 46) when he has less money and becomes clearly more aggressive.

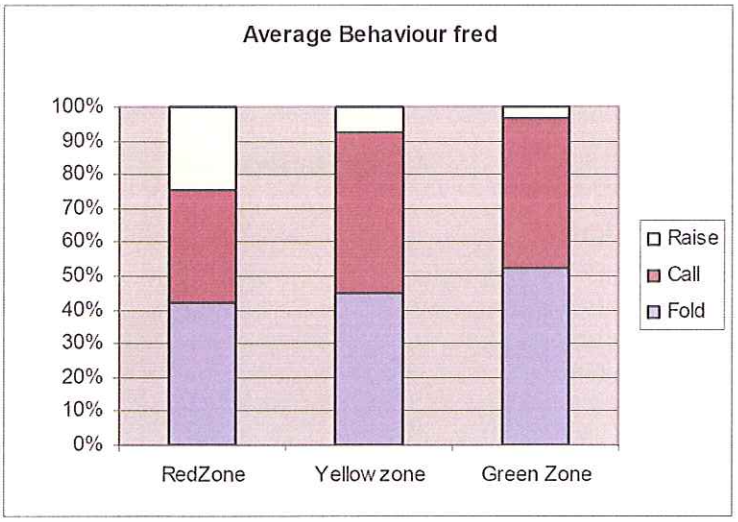


Fig 46: Fred pre-flop strategy based on Money

5.4 - Conclusions

The results obtained demonstrated that all the aspects of classification idealized by the simplified model are indeed relevant for the pre-flop player strategy. This proves that players change strategy considering the number of players in the table, the player's actions and their position.

The tests of the players actions impact also allows to conclude that aggressive actions have more impact than passive ones. It is also clear all actions of the previous players are relevant for the player strategy. The third player will have a different strategy if player 1 raises or if player 1 and 2 raise. This will be a problem a full betting (Pre-flop, flop, turn, river) model due to the fact that tough it is possible to keep all players actions in the pre-flop this will not be possible for all the betting rounds as the size of decision nodes will grow to much and will make it impossible to extract useful information as two exact betting sequence are unlikely to occur. This will make necessary to develop an algorithm that measures the impact of certain betting sequences and group the similar ones. For example the effect of player 1 raise or player 2 raise are very similar in player 3, therefore the error of joining these two situations would be small.

If one factor had to be removed it would be chosen the money, as it has clearly the factor that produces less difference between the players. The result only applies to limit Texas Hold'em games, and cannot be extended to tournaments. In tournaments the money factor is considered a key element, but outside of the tournaments it depends from player to player. In the three players selected although some changes were measured none of them compared with the impact of the other factores were changes up to 35% were measured.

Chapter 6

6 Opponent Modelling

In this chapter we will combine the results given by the Game model with statistic information of the opponent.

6.1 - Player Statistical Behaviour

The statistical model of the player represents the player's general actions. This feature is very common in commercial software. For this project the objectives were step higher that the common software so additional features were added. Besides the common features of Fold Ratio, Call Ratio, and Raise Ratio there is also possible to see the Bluff Ratio, Trap Ratio and money earned distribution. All this features can be studied according to the player money.

<PLAYER>:= // Each player will be defined by the name, M money, Statistic information and money earnings distribution
<NAME>
<M>
<STATISTIC>
<MONEY>

<NAME>:= [String]

<M>:= General | Green Zone | Yellow Zone | Red Zone //General will not considered the money he has

Rezone 0-10 Bets, Yellow Zone 10-20 Bets, RedZone 0-10 Bets

<STATISTIC>:= // average fold, call, raise ratios per pre-flop flop and so on
<FOLD>
<CALL_CHECK>
<RAISE>
<BLUFF>
<TRAP>

<FOLD>:= <GENERAL> <PRE-FLOP> <FLOP> <TURN> <RIVER>

<CALL_CHECK>::= <GENERAL> <PRE-FLOP> <FLOP> <TURN> <RIVER>

<CALL_CHECK>::= <GENERAL> <PRE-FLOP> <FLOP> <TURN> <RIVER>

<CALL_CHECK>::= <GENERAL> <PRE-FLOP> <FLOP> <TURN> <RIVER>

<GENERAL>::=[float]

<PRE-FLOP>::=[float]

<FLOP>::=[float]

<TURN>::=[float]

<RIVER>::=[float]

<BLUFF>::= <TEORICAL>

<TEORICAL>::=[float]

<MONEY>:= <POSITION> <GENERAL> <PRE-FLOP> <FLOP> <TURN> <RIVER>

<POSITION>:= <1> | <2> | <3> | <4> |<5>

<1>:= [float] (small blind)

<2>:= [float] (big blind)

<3>:= [float]

<4>:= [float]

<5>:= [float]

6.1.1 - Algorithms

To measure the parameters of the model, it was used a dynamic average calculator and a probability of win.

```
Dynamicaverage(Numberofseen,observed, probabilyvalue)
{
/* probabilyvalue a value from 0 to 1 of the probability of something to
happen.
   Numberofseen number of previous opportunity
       Observeded ….*\
Probabilituvalue = (probabilityvalue*numberofseen + observed)/(numberofseen+1)
}
// this algortims will automaticaly update the average values.
```

The dynamic average allows updating an average value. So for example the fold ratio is 25% in 4 games, in the 5 game the player did not fold, the fold ratio will decrease to 20%.

```
Probabilityofwin((int player, int numberofplayers)
{
      double sumavailable=0;
      for(int i=1;i<=numberofplayers;i++)
```

```
        {
                if( Player is in the game)
                {
                 Sumedprobability  = Sumedprobability * (
(playermodel[i].general.gamesplayed /
playermodel[i].general.gamesinsituation)*100 ) ;
                }
        }
        if( sumavailable != 0)
        {
        return  (Combination(off all the possible outcomes how many he would
woon C(player/all)) /sumavailable  ;
        }
        else return 0;
        }
```

The probability to win is used to measure the bluff value. The probability to win assumes that when a player folds he folds the worse hands. So for example if a player that never folds at the pre-flop and a second player that folds 50% of the time, the second player would selected 50% of the best hands, and discard the other 50%. With this tactics if both players did not fold at the pre-flop which is the probability of the second player wining? The probability to win algorithm measures that probability. In this case the probability of the second player wining would be 75% against the 25% of player 1.

6.1.2 - Example

Table 2 and table 3 represent an example of a possible result of the statistic information of a player.

Table 3: Player betting strategy

Player snoki	Action	Average	Green Zone	Yellow Zone	Red Zone
PRE FLOP	Fold	64,6	65,4	55,5	0
	Check	19,4	19	25	0
	Raise	15,9	15,5	19,4	100
	Trap	0	0	0	0
	Bluff	-6,8			
FLOP	Fold	4,7	4	12,5	0
	Check	41,3	42,6	31,2	0
	Raise	53,8	53,3	56,2	100
	Trap	0,5	0,6	0	0
	Bluff	-15,4			
TURN	Fold	10,3	9,9	15,3	0
	Check	40	38,8	46,1	100
	Raise	49,6	51,2	38,4	0
	Trap	0,7	0	7,6	0
	Bluff	-15,8			
RIVER	Fold	3,9	4,5	0	0
	Check	56,9	55,6	72,7	0
	Raise	39	39,7	27,2	100
	Trap	1,9	2,2	0	0
	Bluff	-19			

Table 4: Player money distribution

	Average	Green	Yellow	red	n of games	impact	per game
Average	-33	-83	463	5050	500	100%	-0,033$
PRE-FLOP	-76	-75	-87	0	500	67%	-0,051$
FLOP	-542	-512	-833	0	167	6%	-0,035$
Turn	-442	-409	-1000	0	135	7%	-0,031$
River	414	152	2086	5050	100	20%	0,083$
Position	Average	Green	Yellow	Red			
Small Blind	-238	-315	637	0			
Big Blind	55	-69	1764	0			
1	-108	-157	-250	5050			
2	7	-22	392	0			
3	105	135	-250	0			

6.2 - Player Analysis

In this section we analyse a single real player with the tools developed in this thesis. This will serve as an example of the potential of the tools, both as commercial or for agent developing. To protect the identity of the player the player will be called Jimmy.

Jimmy is a regular player in bet and win. He usually plays at multiplayer tables of five players, limited Texas Hold'em Poker. The games observed were with a blind of 0.5$ a small blind of 0,25$. Jimmy was followed over 500 real games.

6.2.1 - Statistical Model

Using the algorithms developed we analysed Jimmy and the results can be seen in table 5.

Table 5: Jimmy results

Player Jimmy	Action	Average	Green Zone	Yellow Zone	Red Zone
PRE FLOP	Fold	64,6%	65,4%	55,5%	0,0%
	Check	19,4%	19,0%	25,0%	0,0%
	Raise	15,9%	15,5%	19,4%	100,0%
	Trap	0,0%	0,0%	0,0%	0,0%
	Bluff	-6,8%			
FLOP	Fold	4,7%	4,0%	12,5%	0,0%
	Check	41,3%	42,6%	31,2%	0,0%
	Raise	53,8%	53,3%	56,2%	100,0%
	Trap	0,5%	0,6%	0,0%	0,0%
	Bluff	-15,4%			
TURN	Fold	10,3%	9,9%	15,3%	0,0%
	Check	40,0%	38,8%	46,1%	100,0%
	Raise	49,6%	51,2%	38,4%	0,0%
	Trap	0,7%	0,0%	7,6%	0,0%
	Bluff	-15,8%			
RIVER	Fold	3,9%	4,5%	0,0%	0,0%
	Check	56,9%	55,6%	72,7%	0,0%
	Raise	39,0%	39,7%	27,2%	100,0%
	Trap	1,9%	2,2%	0,0%	0,0%
	Bluff	-19,0%			
		500	1	36	463

The strategy of Jimmy is quite clear consulting the statistics (table5). He is a very tight aggressive player. His fold ratio is much higher than the usual 50% and after the pre-flop he is very aggressive high 50% of the times raising and almost never folding. The Pre-flop bluff value is negative which means that from the 500 games he won less than 100. The same happens throw the flop turn and river. This bluff value being negative means that he given his selective throw the game he should have statistically won more games. Statistically speaking the player is making one obvious mistake, he is only being selective in the pre-flop. While other players use the pre-flop, the flop and the turn to select the best Hand for the river, Jimmy selects his hands only in the pre-flop. While statistically using the tree moves will provide a better hand, it can also be defended that Jimmy is very selective in the pre-flop to intimidate players throw the flop to the river, making them fold and collect the pot.

Table 6: Jimmy money distribution

	Average	Green	yellow	red	n of games	impact	Dollars per game
Average	-33	-83	463	5050	500	100%	-0,033
PRE-FLOP	-76	-75	-87	0	500	67%	-0,051
FLOP	-542	-512	-833	0	167	6%	-0,035
Turn	-442	-409	-1000	0	135	7%	-0,031
River	414	152	2086	5050	100	20%	0,083
Position	Average	Green	yellow	red			
Small Blind	-238	-315	637	0			
Big Blind	55	-69	1764	0			
1	-108	-157	-250	5050			
2	7	-22	392	0			
3	105	135	-250	0			

In table 6 it is possible to see that the strategy of Jimmy is not working correctly. The key element of his strategy was to intimidate players after the flop, but his results show most of the games after the flop end up in the river. This turns the odds against Jimmy. Being selective after the flop would be expensive. The money distribution is a little strange between the small blind and the big blind. Most of the players lose more money in the big blind than in the small blind. Out of the blinds the results were the expected. The player 3 is the one that makes more money as they have observed the moves of the players in the position 1 and 2, being seat 3 a superior position,.

The strategy of Jimmy is clearly not working for this type of table and as a result he is losing 0.033 dollars per game. The strategy of tight aggressive might not be working because he is playing against skilled players or because in this type of table with just a couple of dollars for pot players do not get easily intimidated and will call.

6.2.2 - Game Model

Using the game model described in chapter 4, it was selected games were Jimmy played with no restrictions of money. The next tables present the fold, call and raise ratio of the table per position.

Table 7: Jimmy pre-flop strategy

	1	2	3	SB	BB
FOLD	45,9%	48,1%	62,7%	**59,8%**	14,5%
CALL	37,9%	37,9%	28,4%	**28,4%**	70,0%
RAISE	16,0%	13,8%	8,7%	**11,6%**	15,3%
FOLD	46,5%	42,4%	47,9%	38,3%	**34,2%**
CALL	39,0%	39,7%	36,3%	50,0%	**59,5%**
RAISE	14,3%	17,8%	15,7%	11,6%	**6,1%**
FOLD	**75,1%**	45,8%	46,6%	30,0%	21,0%
CALL	**0,7%**	40,6%	35,3%	54,8%	67,6%
RAISE	**24,0%**	13,5%	18,0%	15,0%	11,2%
FOLD	60,2%	**75,7%**	51,4%	31,6%	19,1%
CALL	27,9%	**2,2%**	36,0%	49,2%	71,3%
RAISE	11,7%	**22,0%**	12,5%	19,1%	9,5%
FOLD	44,5%	54,7%	**81,7%**	35,1%	20.2%
CALL	36,4%	30,4%	**2,7%**	55,4%	64,1%
RAISE	18,9%	15,5%	**15,5%**	9,4%	15,5%

The results in Blue are the results when Jimmy is playing at that position. Comparing Jimmy with the other players at his table he has a higher fold ratio in all positions.

Table 8: Jimmy strategy after 1 player fold

	Impact of 1 player Folding					
	FOLD		CALL		RAISE	
	Jimmy	Average	Jimmy	Average	Jimmy	Average
FOLD	**5,7%**	8,5%	**5,7%**	6,4%	**6,2%**	8,7%
CALL	**2,6%**	1,9%	**2,6%**	1,5%	**11,6%**	1,7%
RAISE	**8,4%**	7,9%	**8,4%**	6,0%	**17,9%**	7,9%

In table 8 Jimmy strategy after the 1 player folded is shown. The actions of player 1 do not have high impact on the Jimmy actions. The action RAISE of the first player is the one that results in a bigger response especially between the decision between call and raise.

In order to better analyse the effect of other player's actions on Jimmy it will be analysed the impact of the first 2 players folding.

Table 9: Jimmy strategy after first and second player folded

	2 players Fold	
	FOLD	
	Jimmy	Average
FOLD	**9,3%**	16,4%
CALL	**3,0%**	4,7%
RAISE	**9,8%**	12,4%

In the scenario of the players folding Jimmy (table 8 and 9) is clearly less influenced than the Average of players.

With the results presented is clear that Jimmy is a player that has a Pre-Flop strategy that is not very influenced by the other players actions. This makes him a predicable player in the Pre-flop.

The game model besides the impact also gives the distribution of the player after a specific scenario.

Let's imagine a specific scenario. You are the third player, Jimmy is the second, and a general player is in first. In this game the first player folds and then Jimmy raises. I am the third player and I want to have a better idea of Jimmy hand. Using the program I get that in this situation Jimmy never calls, fold 73,1% and raises 28,1%. These values will provide any poker agent or human player with a more accurate knowledge of the scenario, and make a statistical better choice. Just as another example now, let's imagine the first player raised. Jimmy will fold 81,2%, call 12,5% and raise 6,2%. Just out of curiosity the probability of the general player raise in that position is 10%.

6.3 - Conclusion

The combination of results obtained from the game model search tree and the statistical results, allowed a deep and highly mathematical analyse of the player in question. The procedure can be repeated for all players at the table and the result will provide a significant advantage in the game. With more work a proper tool that provides an agent with the best statically pre-flop strategy could be achieved.

The tool that provides an agent with the pre-flop strategy would have to consider two aspects. The search tree results of the game model and the other player's strategy after the pre-flop. For example in the case of Jimmy most of the games after the pre-flop end in the river, so the strategy would have to focus in a strategy that aims for the best possible hand in the river.

Chapter 7

7 Conclusion and Future work

This work intended the study of opponent modelling in multiplayer tables and to implement a tool that enabled to perform real opponent model using data gathered from online poker playing sites.

Initially the history of Artificial Intelligence in games and previous poker research were explained, given a clear idea of what has been developed and the future of the poker research. This thesis comes in the line of future poker research in multiplayer tables that is still unexplored in the current poker research literature.

The main objective was to study opponent modelling in multiplayer tables. This objective was accomplished by developing tools which analyse the tactics of the players in the pre-flop. The results obtained have value both as research and economic in the poker programs business.

The developed tools allow a better analysis of player's behaviours and tendencies. Thus, the tools may be easily used for developing agent-based applications with opponent modelling capabilities for playing in real poker tables.

The model presented to save the information in multiplayer's gathered key elements of the game, as it was proven in the tests shown in chapters 5 and 6.

The small period of time in which the thesis was developed compromised the extent of the experimental procedure and the quality of the results. The main obstacle was to have enough data to ensure a proper hand distribution and guarantee that the average values are correct and not translate the stochasticity of the game.

The main objective of future work will be to improve the quality of the results. This quality may be guaranteed mathematically, or by designing a program that uses the results in real games to obtain advantage over the other players. Designing such a program would be a key breakthrough in poker research as it typically focuses more in experimental results than mathematical statistics.

Bibliography

[Billings et al., 2002] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron. The challenge of poker. Artificial Intelligence, 134(1.2):201.240, January 2002. [online] available at: http://poker.cs.ualberta.ca/papers/AIJ02.html [consulted in June 2008]

[Billings et al., 2004] Darse Billings, Michael Bowling, Neil Burch, Aaron Davidson, Rob Holte, Jonathan Schaeffer, and Terence Schauenberg. Game tree search with adaptation in stochastic imperfect information games. In International Conference on Computers and Games (CG), pages 21–34, 2004.

[Billings, 2006] Darse Billings. Algorithms and Assessment in Computer Poker. 2006. [online] available at http://poker.cs.ualberta.ca/papers/Papers/billings.phd.pdf [consulted on June 2008]

[DeDonno , 2008] Michael A. DeDonno, Douglas K. Detterman. Gaming Law Review. February 1, 2008, 12(1): 31-36. doi:10.1089/glr.2008.12105, 2008.

[Félix et al., 2008] Dinis Felix, Luis Paulo Reis. An Experimental Approach to Online Opponent Modeling in Texas Hold'em Poker, 19th Brazilian Symposium on Artificial Intelligence (SBIA), Salvador, Bahia, Brazil, October 26-30, 2008 (to appear)

[Félix et al., 2008] Dinis Felix, Luis Paulo Reis. Opponent Modelling in Texas Holdem Poker as the Key for Success, 18th European Conference on Artificial Intelligence, University of Patras, Patras, Greece, July 21-25, 2008 (to appear)

[Gilpin et al., 2006] Andrew Gilpin and Tuomas Sandholm. A competitive texas Hold'em poker player via automated abstraction and real-time equilibrium computation. In Proceedings of the Twenty-First Conference on Artificial Intelligence (AAAI-06) Pages: 1453 - 1454, 2006.

[Gilpin et al., 2007] Andrew Gilpin and Tuomas Sandholm. Better automated abstraction techniques for imperfect information games, with application to texas Hold'em poker. In Autonomous Agents and Multi-Agent Systems (AAMAS'07) Article No. 192 2007 .

[Hauk et al.,2004] T. Hauk, M.Buro, and J. Schaeffer. *-minimax performance in backgammon. In H.J. van den Herik, Y. Bjornsson. and N. Netanyahu, editors, Computers and Games: 4th international Conference, CG'04 Ramat-Gan, Israel, July 5-7, 2004, Revised Papers, volume 3846 of Lecture Notes in Computer Science, pages 51-66. Springer-Verlag GmbH, 2004.

[Irc poker database, 2008] IRC Poker Database, http://games.cs.ualberta.ca/poker/IRC/ [consulted on june 2008]

[Johanson, 2007] Michael Bradley Johanson, Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player, Master of Science, 2007. [online] available at http://poker.cs.ualberta.ca/papers/Papers/johanson.msc.pdf [consulted on June 2008]

[John Nash, 1951] J. F. Nash. Non-cooperative games. Annals of Mathematics, 54:286-295, 1951.

[Luger et al,2004] Luger, George & Stubblefield, William (2004), Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th ed.), The Benjamin/Cummings Publishing Company, Inc., pp. 720, 2004

[McCarthy, 2007] John McCarthy , What is artificial inteligence?, [online] http://www-formal.stanford.edu/jmc/whatisai/whatisai.html [Consulted on june 2008] .

[McKenna, 2005] James A. McKenna. Beyond Tells: Power Poker Psychology, Lyle Stuart,2005

[Neumann, 1941] J. von Neumann and O. Morgenstern. The Theory of Games and Economic Behavieur. Princeton University Press, 2nd edition, 1947.

[Nolan Bard, 2008] Nolan Deans Carlisle Bard, Using State Estimation for Dynamic Agent Modelling, Master of Science, 2008 [online] available at http://poker.cs.ualberta.ca/papers/Papers/bard.msc.pdf [consulted on June 2008]

[Papp, 1998] D. Papp. Dealing with imperfect information in poker. Master's thesis, Department of Computing Science, University of Alberta, 1998. [online] available at: http://www.cs.ualberta.ca/~jonathan/Grad/papp/thesis.html [consulted in June 2008]

[Poker Academy, 2008] Poker Academy. Your Source for great poker sofware

http://www.poker-academy.com [consulted on June 2008]

[Poker Tracker, 2008] Poker Tracker, Online Poker Tracking & Analysis Software Tool, http://www.pokertracker.com/ [Consulted on June 2008]

[Russel et al.,2003] Russell, Stuart J. & Norvig, Peter, Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, NJ: Prentice Hall, 2003

[Schauenberg, 2006] Terence Schauenberg. Opponent modeling and search in poker. Master's thesis, University of Alberta, 2006.

[Schwab, 2004] Brain Swab, AI Game Engine Programming Charles River Media; 1 edition, 2004.

[Shannon,1950] Claude Shannon. "Programming a Computer for Playing Chess". Philosophical Magazine 41 (314), 1950.

[Sun Tzu, 6th century BC] Sun tzu, The Art of War, El Paso Norte Press; Special edition (March 21, 2005) 6th century BC.

[Tesauro, 2002] G. Tesauro. Programming backgammon using self-teaching neural nets. Artificial Intelligence, Elsevier Science Publishers Ltd.volume 134(1-2):181-199 January 2002.

[University of Alberta Games, 2008] The University of Alberta GAMES Group. World Wide Web, 2008. http://www.cs.ualberta.ca/.games/ [Consulted on June 2008]

[Wikipedia, 2008] Wikipedia – Poker Probability. Wikipedia: The Free Online Encyclopedia, http://en.wikipedia.org/wiki/Poker_probability [consulted in June 2008]

[Wikimedia, 2008] Wikimedia Commomns - Texas Hold'em Table. http://commons.wikimedia.org/wiki/Image:Holdem_Table.png [consulted in June 2008]

[Wikimedia, 2008] Wikipedia – Online poker . http://en.wikipedia.org/wiki/Online_poker [consulted in June 2008]

Appendix A – Glossary of Poker Terms

- **All-in**. To have one's entire stake committed to the current pot. Action continues toward a side pot, with the all-in player being eligible to win only the main pot.
- **All-in Equity**. The expected value income of a hand assuming the game will proceed to the showdown with no further betting (i.e., a fraction of the current pot, based on all possible future outcomes).
- **Bad Beat.** An unlucky loss. In particular, losing a game where the opponent probably should have folded, but instead got extremely lucky to win.
- **Bet**. To make the first wager of a betting round (compare raise).
- **Bet for Value.** To bet with the expectation of winning if called (compare bluff).
- **Big Bet**. The largest bet size in Limit poker (e.g., $20 in $10-$20 Hold'em).
- **Big Blind** (sometimes called the Large Blind). A forced bet made before the deal of the cards (e.g., $10 in $10-$20 Hold'em, posted by the second player to the left of the button).
- **Blind**. A forced bet made before the deal of the cards (see small blind and big blind).
- **Bluff.** To play a weak hand as though it were strong, with the expectation of losing if called (see also semi-bluff and pure bluff, compare bet for value).
- **Board** (or Board Cards). The community cards shared by all players.
- **Board Texture**. Classification of the type of board, such as having lots of high cards, or not having many draws (see dry).
- **Button**. The last player to act in each betting round in Texas Hold'em. Also called the dealer button, representing the person who would be the dealer in a home game.
- **Call**. To match the current level of betting. If the current level of betting is zero, the term check is preferred.

- **Cap**. (a) The maximum number of raises permitted in any single round of betting (typically four in Limit Hold'em, but occasionally unlimited). (b) (vt) To make the last permitted raise in the current betting round (e.g., after a bet, raise, and re-raise, a player caps the betting).
- **Check.** To decline to make the 1rst wager of a betting round (compare call).
- **Check-Raise.** To check on the 1rst action, with the intention of raising in the same betting round after an opponent bets.
- **Community Cards.** The public cards shared by all players.
- **Connectors.** Two cards differing by one in rank, such as 7-6. More likely to make a straight than other combinations.
- **Dominated**. A Hold'em hand that has a greatly reduced chance of winning against another because one or both cards cannot make a useful pair (e.g., KQ is dominated by AK, AQ, AA, KK, and QQ, but not by AJ or JJ).
- **Draw**. A holding with high potential to make a strong hand, such as a straight draw or a flush draw (compare made hand).
- **Draw Potential**. The relative likelihood of a hand improving to be the best if it is currently behind.
- **Drawing Dead**. Playing a draw to a hand that will only lose, such as drawing to a flush when the opponent already holds a full house.
- **Drawing Hand**. A hand that has a good draw (compare made hand).
- **Dry**. Lacking possible draws or betting action, as in a dry board or a dry game.
- **Equity (or Pot Equity)**. An estimate of the expected value income from a hand that accounts for future chance outcomes, and may or may not account for the effects of future betting (e.g., all-in equity).
- **Expected Value (EV)** (also called mathematical expectation). The average amount one expects to win in a given game situation, based on the payoffs for each possible random outcome.
- **Flop.** The first three community cards dealt in Hold'em, followed by the second betting round (compare board).
- **Fold**. To discard a hand instead of matching the outstanding bet, thereby losing any chance of winning the pot.
- **Fold Equity.** The equity gained by a player when an opponent folds. In particular, the positive equity gained despite the fact that the opponent's fold was entirely correct.

- **Forward Blinds**. The logical extension of blinds for heads-up (two-player) games, where the first player posts the small blind and the second player (button) posts the big blind (compare reverse blinds). (Both rules are seen in practice, with various casinos and online card rooms having different policies for multi-player games that have only two active players).
- **Free-Card Danger**. The risk associated with allowing an opponent to improve and win the pot without having to call a bet (in particular, when they would have folded).
- **Free-Card Raise**. To raise on the flop intending to check on the turn.
- **Game.** (a) A competitive activity in which players contend with each other according to a set of rules (in poker, a contest with two or more players). (b) A single instance of such an activity (in poker, from the initial dealing of the cards to the showdown, or until one player wins uncontested).
- **Game Theory**. Among serious poker players, game theory normally pertains to the optimal calling frequency (in response to a possible bluff), or the optimal bluffing frequency. Both depend only on the size of the bet in relation to the size of the pot.
- **Hand.** (a) A player's private cards (e.g., two hole cards in Hold'em). (b) One complete game of poker (see game (b)).
- **Heads-up.** A two-player (head-to-head) poker game.
- **Hole Card.** A private card in poker (Texas Hold'em, Omaha, 7-Stud, etc.).
- **Implied Odds**. (a) The pot odds based on the probable future size of the pot instead of the current size of the pot (positive or negative adjustments). (b) The extra money a strong hand stands to win in future betting rounds (compare reverse implied odds).
- **Kicker.** A side card, often deciding the winner when two hands are otherwise tied (e.g., a player holding Q-J when the board is Q-7-4 has top pair with a Jack kicker).
- **Large Blind** (usually called the Big Blind). A forced bet made before the deal of the cards (e.g., $10 in $10-$20 Hold'em, posted by the second player to the left of the button).
- **Loose Game.** A game having several loose players.
- **Loose Player.** A player who does not fold often (e.g., one who plays most hands at least to the flop in Hold'em).

- **Made Hand**. A hand with a good chance of currently being the best, such as top pair on the flop in Hold'em (compare draw).
- **Mixed Strategy.** Handling a particular type of situation in more than one way, such as to sometimes call, and sometimes raise.
- **Offsuit**. Two cards of different suits (also called unsuited, compare suited).
- **Open-Ended Draw.** A draw to a straight with eight cards to make the straight, such as 6-5 with a board of Q-7-4 in Hold'em.
- **Outs**. Cards that will improve a hand to a probable winner (compare draw).
- **Pocket Pair.** Two cards of the same rank, such as 6-6. More likely to make three of a kind than other combinations (see set).
- **Post-flop**. The actions after the flop in Texas Hold'em, including the turn and river cards interleaved with the three betting rounds, and ending with the showdown.
- **Pot**. The common pool of all collected wagers during a game.
- **Pot Equity** (or simply Equity). An estimate of the expected value income from a hand that accounts for future chance outcomes, and may or may not account for the effects of future betting (e.g., all-in equity).
- **Pot Odds**. The ratio of the size of the pot to the size of the outstanding bet, used to determine if a draw will have a positive expected value.
- **Pre-flop**. The first round of betting in Texas Hold'em before the flop, beginning with the posting of the blinds and the dealing of the private hole cards.
- **Pure bluff.** A bluff with a hand that can only win if the opponent folds (compare semi-bluff).
- **Pure Drawing Hand**. A weak hand that can only win by completing a draw, or by a successful bluff.
- **Raise**. To increase the current level of betting. If the current level of betting is zero, the term bet is preferred.
- **Raising for a Free-card.** To raise on the top intending to check on the turn.
- **Rake.** A portion of the pot withheld by the casino or host of a poker game, typically a percentage of the pot up to some maximum, such as 5% up to $3.
- **Re-raise.** To increase to the third level of betting after a bet and a raise.
- **Reverse Blinds.** A special rule sometimes used for heads-up (two-player)

games, where the second player (button) posts the small blind and the first player posts the big blind (compare forward blinds). (Both rules are seen in practice, with various casinos and online card rooms having different policies for multi-player games that have only two active players).

- **Reverse Implied Odds.** The unaccounted (negative) money a mediocre hand stands to lose in future betting rounds (compare implied odds (b)).
- **River.** The fifth community card dealt in Hold'em, followed by the fourth (and final) betting round.
- **Semi-bluff**. A bluff when there are still cards to be dealt, with a hand that might be the best, or that has a reasonable chance of improving to the best if it is called (compare pure bluff).
- **Second pair.** Matching the second highest community card in Hold'em, such as having 7-6 with a board of Q-7-4.
- **Session.** A series of games, typically lasting several hours in length.
- **Set.** Three of a kind, formed with a pocket pair and one card of matching rank on the board. A very powerful and well-disguised hand (compare trips).
- **Short-handed Game.** A game with less than the full complement of players, such as a Texas Hold'em game with five or fewer players.
- **Showdown.** The revealing of cards at the end of a game to determine the winner.
- **Side pot**. A second pot for the remaining active players after another player is all-in.
- **Slow-play.** To check or call a strong hand as though it were weak, with the intention of raising in a later betting round (compare smooth-call and checkraise).
- **Small Bet**. The smallest bet size in Limit poker (e.g., $10 in $10-$20 Hold'em).
- **Small Blind.** A forced bet made before the deal of the cards (e.g., $5 in $10-$20 Hold'em, posted by the first player to the left of the button).
- **Smooth-call**. To only call a bet instead of raising with a strong hand, for purposes of deception (as in a slow-play).
- **Suited.** Two cards of the same suit, such as both Hearts. More likely to make a flush than other combinations (compare offsuit or unsuited).
- **Table Image**. The general perception other players have of one's play.

- **Table Stakes.** A poker rule allowing a player who cannot match the outstanding bet to go all-in with his remaining money, and proceed to the showdown (also see side pot).
- **Texture of the Board.** Classification of the type of board, such as having lots of high cards, or not having many draws (see dry).
- **Tight Player**. A player who usually folds unless the situation is clearly profitable (e.g., one who folds most hands before the flop in Hold'em).
- **Time Charge.** A fee charged to the players in a poker game by a casino or other host of the game, typically collected once every 30 minutes.
- **Top Pair**. Matching the highest community card in Hold'em, such as having Q-J with a board of Q-7-4.
- **Trap**. To play a strong hand as though it were weak, hoping to lure a weaker hand into betting. Usually a check-raise or a slow-play.
- **Trips.** Three of a kind, formed with one hole card and two cards of matching rank on the board. A strong hand, but not well-disguised (compare set).
- **Turn**. The fourth community card dealt in Hold'em, followed by the third betting round.
- **Unsuited.** Two cards of different suits (also called offsuit, compare suited).
- **Value Bet**. To bet with the expectation of winning if called (compare bluff).
- **Wild Game.** A game with a lot of raising and re-raising. Also called an action game.