An Investigation into Tournament Poker Strategy using Evolutionary Algorithms

Richard G. Carter

Doctor of Philosophy

Centre for Intelligent Systems and their Applications

School of Informatics

University of Edinburgh

2007

Abstract

Poker has become the subject of an increasing amount of study in the computational intelligence community. The element of imperfect information presents new and greater challenges than those previously posed by games such as checkers and chess. Advances in computer poker have great potential, since reasoning under conditions of uncertainty is typical of many real world problems.

To date the focus of computer poker research has centred on the development of ring game players for limit Texas hold'em. For a computer to compete in the most prestigious poker events, however, it will be required to play in a tournament setting with a no-limit betting structure. This thesis is the first academic attempt to investigate the underlying dynamics of successful no-limit tournament poker play. Professional players have proffered advice in the non-academic poker literature on correct strategies for tournament poker play. This study seeks to empirically validate their suggestions on a simplified no-limit Texas hold'em tournament framework.

Starting by using exhaustive simulations, we first assess the hypothesis that a strategy including information related to game-specific factors performs better than one founded on hand strength knowledge alone. Specifically, we demonstrate that the use of information pertaining to one's seating position, the opponents' prior actions, the stage of the tournament, and one's chip stack size all contribute towards a statistically significant improvement in the number of tournaments won.

In extending the research to combine all factors we explain the limitations of the exhaustive simulation approach, and introduce evolutionary algorithms as a method of searching the strategy space. We then test the hypothesis that a strategy which combines information from all the aforementioned factors performs better than one which employs only a single factor. We show that an evolutionary algorithm is successfully able to resolve conflicting signals from the specified factors, and that the resulting strategies are statistically stronger than those previously discovered.

Our research continues with an analysis of the results, as we interpret them in the context of poker strategy. We compare our findings to poker authors' recommendations, and conclude with a discussion on the many possible extensions to this work.

Acknowledgements

First and foremost I would like to thank my supervisor, Dr John Levine, for his contributions and support in my research efforts. I feel very fortunate that I found a supervisor who shared my passion for the subject over the last three and a half years. Additionally, I wish to thank Dr Dave Robertson for providing much needed guidance at several junctures.

For making the PhD experience enjoyable and memorable, I would like to give my appreciation to all the staff and fellow students within CISA. Without them the journey would have been far lonelier.

Finally I owe my biggest debt of gratitude to my partner, soulmate, best friend, and financial sponsor, Ruth Mitchell. Without her I would not have had the opportunity to undertake this research, nor the kicks up the backside necessary to complete it.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Richard G. Carter*)

To Ruth

Table of Contents

Chapter 1

Introduction

1.1 Overview

This thesis presents an investigation into the nature and discovery of robust tournament poker strategies through the application of evolutionary computation. Whilst research into games is commonplace in the artificial intelligence (AI) community, work on poker is nascent. Impressive results have already been achieved through academic research into the ring game format of poker, but tournament poker play has so far been neglected. This research aims to address this gap in the academic literature.

1.2 Motivation

Games provide a common test bed for researchers into computer science due to their well defined problem domains. The goal of such research is typically to develop the strongest possible programs, and the rules of the game provide fixed bounds within which this goal can be achieved. Poker is of particular interest since it provides many different challenges for machine intelligence which are not found in the more commonly researched games such as chess and draughts. A strong computer poker program needs to display a range of intelligent behaviours to play the game successfully, such as handling imperfect and incomplete information, competing against multiple adversaries, opponent modelling and risk management [11]. Many of these competencies are found in other problem domains. Results from the study of poker may prove beneficial in a wide variety of other areas such as bargaining problems, financial trading and forecasting, where reasoning under conditions of uncertainty is present.

To properly define a domain for research into the application of AI techniques to poker we are required to specify the:

- Poker variant,
- Betting structure,
- Game format (i.e. ring game or tournament), and
- Number of players.

If we consider the ultimate aim for research into computer poker as the development of a player which is able to successfully compete for the world championship, then we can select choices which align with this goal. By common agreement the poker world champion is deemed to be the winner of the $10,000 buy-in no-limit Texas hold'em tournament, played during the World Series of Poker in Las Vegas every summer [59]. Therefore a poker player that is capable of winning the world championship needs to be able to play:

- Texas hold'em poker,
- With a no-limit betting structure,
- In a tournament setting,
- Against any number of opponents.

When we investigate the research that has previously been undertaken in computer poker we see that the last three of these targets have yet to be properly addressed. The present state of the art tackles Texas hold'em as played with a limit betting structure in a ring game format, and typically against only one other opponent. This study marks the first academic attempt to research poker as played in a tournament setting, with no-limit betting, and against multiple opponents.

The non-academic literature, comprising books and articles written predominantly by professional poker players, contains suggestions for strong tournament poker play. The strategies which they espouse are the result of the authors' own experiences, and as such are usually more anecdotal than scientific in nature. In performing this study we shall compare our results with such authors' recommendations, to see whether the professionals' advice can be demonstrated empirically.

1.3 Aims and Scope

This central intent of this thesis is to better understand tournament poker strategy. The goal is not, therefore, to produce the strongest possible players; rather we wish to comprehend some of the necessary elements of a strong player. It is then expected that these strategic elements can be incorporated into the poker programs of future researchers who do seek to develop highly competitive computer poker strategies.

Since this research does not directly build upon any previous studies, we must first create a framework for the experiments. Texas hold'em is an extremely involved game, and therefore certain simplifications are introduced to make the problem of understanding tournament strategy as practical as possible without losing all of the complexity.

First of all we limit the number of competitors to ten. In a real tournament it is possible to have literally thousands of entrants. All such competitions are eventually reduced to a final table of around ten players, though, and this forms the starting point of our research.

To simplify the betting actions available to the players in our no-limit tournaments we restrict their options to either all in or fold. In real no-limit poker a player may decide how much they wish to stake in a bet. Replacing this possibility with a choice of all or nothing shifts the focus to the more general and important strategic question of when to bet, rather than how much. An additional benefit of this simplification is that we only need concern ourselves with one round of betting, rather than the four found in authentic Texas hold'em. This greatly reduces the game's strategy space to a more tractable size, thus facilitating its investigation.

Lastly we choose to neglect opponent modelling. This is one of the most crucial elements of a real player's poker strategy: all strong players decide upon a course of action by incorporating knowledge of their opponents' previous betting patterns. However, whilst it is extremely important, opponent modelling is a secondary strategic element to the more fundamental aspects of tournament poker play which we seek to address. Opponent modelling can be thought of as an additional layer built upon a foundation of correct tournament strategy. This kernel of understanding is not only required when competing against unknown opponents, but is perpetually essential in guiding a tournament player's general strategic considerations.

Having built a framework upon which to investigate tournament poker strategy, we must finally decide which technique from the AI library to employ. To find strategies which maximize the number of tournaments won we start our enquiries with small problems for which the testing of all permissible strategies is feasible. For this method we use the term exhaustive simulation. As we expand the problems we find that this procedure is no longer practical, and instead seek to use another appropriate technique. The tackling of an optimization problem in which potential solutions can be thought of as candidates within a population leads us naturally to the use of evolutionary algorithms (EAs). We recognize that this choice was not mandatory, and that other techniques such as simulated annealing or hill climbing [102] might alternatively have been employed. As we shall see, though, EAs have been used extensively in similar studies into computer games and appear an appropriate tool for the task.

1.4 The Organization of the Thesis

Chapter 2 gives an overview of the game of poker, focusing in particular on the Texas hold'em variant. We discuss previous academic research into games generally, and poker specifically. The chapter concludes by highlighting the relevant non-academic literature on poker strategy, stressing the importance of seating position and hand strength knowledge to a competitive player.

Following this Chapter 3 provides a more detailed discussion of tournament Texas hold'em. Acknowledging the non-academic literature we discuss the significance of information related to the stage of the tournament and one's stack size in a player's decision making. We then talk about all in or fold systems for playing in a Texas hold'em tournament, and show how such a method fared in an organized computer challenge.

Our initial tournament poker investigations using exhaustive simulations are presented in Chapter 4. We show how the four game factors highlighted in the two previous chapters can be used in conjunction with hand strength knowledge to build strong tournament poker programs. We demonstrate statistically that players who are able to utilize this additional information fare better than those who base decisions on their hand strength alone.

Given the limitations of the exhaustive simulation approach, we turn to evolutionary algorithms in Chapter 5. We detail the different classes of EA, and show how they have been used in previous academic enquiries. We conclude the chapter with an example of how a genetic algorithm can be used to encode players for a toy poker problem.

Chapter 6 returns to empirical methods, by using the EA techniques of the preceding chapter. After an explanation of the experimental framework we employ evolutionary computation to find poker strategies which utilize an amalgamation of the game factors assessed individually in the exhaustive simulations. Using statistical hypothesis testing we show that players who employ the complete information set score a significantly larger number of tournament wins than the strategies previously discovered in Chapter 4.

Further analysis of the results of this research is presented in Chapter 7. We compare the strategies found by evolutionary computation with those from the exhaustive simulations, then interpret the numerical data from our experiments in terms of poker strategy. We conclude with a comparison of our findings to the strategic recommendations in the non-academic poker literature.

Chapter 8 contains many suggestions for ways in which this work could be extended by future researchers in the field, towards the goal of creating a world championship winning computer poker player.

The thesis concludes with Chapter 9, which summarizes the study and discusses the contributions of this thesis to computer poker research.

Chapter 2

Poker

This chapter gives a general introduction to the game of poker, before focusing specifically on the variant known as Texas hold'em. We introduce much of the jargon that is used in discussing the game, and give an example of how a hand of Texas hold'em proceeds. Next we examine recent methods of building computer game playing programs, looking at some of the different artificial intelligence techniques that have been used. Following this we discuss academic investigations into poker. Then we see how an approach combining game theory and opponent modelling is producing the strongest computer poker players at this time. Finally we survey the non-academic poker literature, and discuss some of the strategic considerations employed by professional players to guide their decision making.

2.1 Introduction to Poker

Poker is a generic term covering hundreds of different variations of card games, in which players with fully or partially concealed cards bet sequentially against each other to win a communal pot of money. Poker is a highly skilled gambling game in which strategies and techniques can be learnt and applied to defeat one's opponents. As with most popular card games, however, an element of chance is present in poker due to the random deal of the cards.

The history of poker is not accurately known, but it is most commonly believed to be derived from the gambling card game of primero [122]. This game was popular in Italy in the 16^{th} century, before moving across western Europe to England. Poker also shares similarities to the Persian game of as nas and the English game brag, and it is possible that these pastimes and others have influenced the development of the present games.

Whilst there are many forms of poker, most share several common features. A single poker game may contain any number of players, typically from two to ten. A standard 52-card deck of playing cards is most often used, with a prescribed dealer responsible for shuffling between hands and overseeing the game. The right to deal is passed clockwise with each hand, and play proceeds with the player to the dealer's immediate left.

On the commencement of a hand of poker, one or more players are required to post a forced bet into the pot, known either as a *blind* or *ante* depending on the specific form of the game. The dealer then gives each player the appropriate number of cards one at a time, and these may either be *private* (dealt face down and seen only by their recipient), or *public* (dealt face up for all players to see).

After the deal, a betting round occurs in which players sequentially wager money against each other. Typically there are three options available to each player, with different terminology depending upon whether money has already been entered into the pot in that betting round. Assuming no money has been posted (i.e. the pot is *unopened*) a player may choose to *bet* by placing money into the pot, *check* (a bet of zero chips), or *fold*. A player who folds takes no further active part in the hand and gives their cards back to the dealer. If the pot has been opened each player may either *raise* by increasing the size of the bet, *call* by matching it, or again fold. A betting round ends when all active players (i.e. those that have not folded) have wagered exactly the same amount.

After the initial betting round, additional cards are dealt - in some poker games as a replacement for existing cards - again followed by a round of betting. Most poker games contain from two to four such stages of dealing and betting. If in any round a player's bet or raise is not met by at least one caller, that player wins the pot outright and is under no obligation to reveal his hand. If two or more players remain active on completion of the final betting round, they must declare their hands in a *showdown*. The holder of the best hand, determined by a relative ranking of all possible hands, wins all monies posted into the pot.

2.2 Texas Hold'em Poker

At present the most widely played poker variant is Texas hold'em (also known as hold'em or holdem). It is this form which is used as the main event in the annual

World Series of Poker [59], and is played in many casinos and home games across the world. Hold'em is believed to have been invented in Texas in the early 1900s, and its popularity has spread due to it being one of the most strategically complex forms of poker. A game of Texas hold'em can be played with up to 22 players, although it is more usual to see from two to ten players at a single table.

The objective in a standard game of hold'em is to win the pot. There are two ways that a player can succeed in this aim. Firstly, a player may have the best hand in a showdown. Alternatively, a player wins the pot if all other players have folded their cards.

Play commences with the two players to the dealer's immediate left posting forced bets, known sequentially as the *small blind* and *big blind*. The size of the big blind is typically twice that of the small blind. This blind money initiates the pot for which all players compete.

Two cards are then dealt face down to each player, and comprise that player's *hole cards*. These cards are private, since only the player to whom they are dealt knows their identity. Hole cards can be classified as a *pair* (two cards of the same rank), *suited* (two cards of the same suit), or *offsuit* (two non-paired cards of different suits). There are 1,326 unique two-card starting hands in Texas hold'em, but due to the equivalence of suits this number can be reduced to 169. A fuller discussion on Texas hold'em starting hands can be found in Appendix A.

Once all cards are dealt the first betting round, called the *pre-flop* round, takes place. Here the player to the immediate left of the big blind acts first, with play continuing clockwise. Each player in turn may either call the previous bet, raise it, or fold and release his cards.

After the pre-flop betting, three cards are dealt face up in the middle of the table to form the *flop*. Cards dealt publicly in this way are known as *community cards* and form the *board*, and may be used by all the players in constructing their final hands. The flop is followed by another betting round, starting this time with the first active player to the left of the dealer. Once complete, another single card (the *turn*) is placed on the table, and a third betting round takes place. A final community card (the *river*) follows next, then a last round of betting.

During any of the four betting rounds it is possible for one or more players to become *all in*. This term denotes the situation where a competitor has placed all of their chips into the pot. An all in player takes no part in the residual betting rounds, but otherwise play continues as normal.

If two or more players remain - all in or not - at the conclusion of the final betting round, a showdown determines the winner. In a showdown each player selects his best five-card poker hand from the seven available: his two hole cards plus the five community cards. There is no stipulation on the number of cards that must be used from either one's hand or the board. The pot is passed to the holder of the best hand, as determined by the ranking of all possible poker hands. In exceptional circumstances two or more players may have equally ranked hands, in which case the pot is shared out equally.

Texas hold'em typically employs one of three different forms of betting structure: *limit*, *pot-limit*, or *no-limit*. In limit hold'em, the size of the bets are fixed amounts. In the first two betting rounds each bet or raise is equal to the size of the big blind. In the final two rounds the fixed bet size doubles. In pot-limit hold'em a player may wager any amount between the size of the big blind and the size of the pot. No-limit hold'em allows each player to bet any amount from the size of the big blind up to the amount that they have in their stack. Pot-limit and no-limit poker are commonly referred to as *big bet* poker.

As with all poker games, hold'em can either be played as a *ring game* (also known as *cash game*) or in a *tournament*. In a ring game the players contest pots with real money and no predetermined end time. A poker tournament, on the other hand, is played with tournament chips and ends once the game has been reduced to a preset number of players (usually one). In a ring game players may continually enter and exit the table, and players who lose all of their chips are able to purchase more to bring onto the table. By contrast, in a tournament the players usually buy a set number of tournament chips before the game, and are eliminated from the competition if their stack size reaches zero.

Appendix B gives an example of the play of a hand of no-limit Texas hold'em, and contains the ranking of all possible hands used in a showdown.

2.3 Approaches to Computer Game Playing

Having introduced the game which forms the basis of our research, we now step back and look at the categorization of games and some of the techniques that have proved useful in their investigation.

A game can typically be assigned to one of four different categories, depending on whether it is deterministic or not, and on whether or not it contains perfect information [17]. By definition, deterministic games contain no chance element. Such games include checkers and chess. The outcome of a deterministic game depends solely on the skills and abilities of the competitors. By contrast, a non-deterministic game admits a chance element, usually through the roll of dice or the shuffle of cards. Backgammon is an example of a non-deterministic game.

Games can also be of either perfect or imperfect information. This distinction depends upon whether the competitors have all the information relating to the present game state[1]. Checkers and chess are perfect information games, since both players can see the complete state of the game on the board. Contrariwise, the players of an imperfect information game, such as bridge or battleships, must act under conditions of uncertainty.

As with all poker variants Texas hold'em is an example of a non-deterministic game of imperfect information. The randomization brought about by shuffling the cards introduces non-determinism, whilst the concealed nature of the players' hole cards leads to the state of imperfect information.

The classification of several well known games is given in Table 2.1.

	Perfect Information	**Imperfect Information**
Deterministic	Chess, Checkers, Go	Battleships
Non-deterministic	Backgammon, Ludo	Bridge, Poker

Table 2.1: The classification of several games.

Researchers have had tremendous successes in recent years with investigations into games of perfect information. The first major triumph for computer game programs was Jonathan Schaeffer's *Chinook* [103], designed to play checkers. The then-world champion Marion Tinsley competed against *Chinook* in 1994, but was forced to withdraw before completion of the match due to ill health. Subsequent to this *Chinook* inherited the title of "Man versus Machine Champion of the World".

Some of the most notable successes of computer game programs have been those developed for chess. The well documented defeat of Garry Kasparov by IBM's Deep Blue in 1995 [25] was the first time that a reigning chess world champion had been beaten by a computer under regular time controls. Kasparov himself predicted in the

[1] Formally, a game of perfect information is one in which all the information sets of the game tree contain a single node.

mid-1990s [29] that a computer would be capable of beating human opposition to the world championship title by 2010.

In the cases of checkers and chess, the complete game state is known to both participants at all times during the game. No potentially important decision relevant information remains hidden. With this presence of perfect information a minimax search coupled to a position evaluation function is able to select stronger moves than a typical human opponent by looking deeper into the possible consequences of each action. Such brute-force search continually yields stronger programs as increases in computational power allow the machine to examine a larger number of potential positions. Deep Blue was reported to be able to analyse up to a maximum of 330 million positions per second [25].

Brute-force search is not yet able to defeat human opposition at all games of perfect information. The ancient Chinese game of Go is still extremely challenging for researchers into computer game playing due to the size of the search space [86, 87, 88]. A comparison of the game space size of Go to the more familiar board games of checkers and chess can be made by approximating each game's tree size. This can be done by raising a game's average branching factor to the number of individual moves (or plies) in an average game.

Estimates of the game space sizes for checkers, chess, and Go are shown in Table 2.2.

Game	Board	Pieces	Branching Factor	Game Space
Checkers	8-by-8	32	8 to 20	5x10^{20}
Chess	8-by-8	32	35	10^{120}
Go	19-by-19	381	250	10^{360}

Table 2.2: Game space sizes (taken from [43]).

The approach that has proved successful for checkers and chess is not practical for non-deterministic games of imperfect information such as poker and bridge. The game trees of these card games are also very large but include the additional complication of the random deal leading to multinodal information sets. This manifestation of imperfect information means that a player can never be entirely sure at which exact node in the game tree they are at during the play of a game. Computer bridge research [55] has a greater historical precedent than that of poker. Methods based on planning [50] have had some success thus far, but the advance of computer bridge programs is reported to be slow [54].

The non-determinism of a game is not necessarily a barrier to successfully creating a strong computer player. Indeed, the stochastic nature of backgammon was beneficial to Gerald Tesauro's development of TD-Gammon [118]. The neural network used trains itself to be an evaluation function through self-play. Temporal-difference (TD) learning seeks to reduce the error between the learner's current prediction and the next prediction at the next time step. The non-determinism of backgammon aids this process since the algorithm is able to visit many different nodes in the game tree and thereby produce more rounded estimates.

We can see, therefore, that the nature of the game can affect the decision of which AI techniques to use in its investigation. Before looking at the computational methods that have been applied to poker we next review the first scientific investigations into the game by the pioneers of game theory.

2.4 Early Game Theoretic Investigations into Poker

The earliest academic investigations into the game of poker were undertaken in the mid-twentieth century. Such luminaries as Kuhn [80], Nash and Shapley [89], and von Neumann and Morgenstern [120], employed greatly simplified poker variants to formulate a framework for strategy selection in non-cooperative environments. The toy pokers that were examined were typically only two- or three-player games, and used pared decks of cards to reduce the space of possible strategies. Much of this early work on poker formed the foundations of game theory, a discipline which has grown substantially over the last half-century. Applications of game theory are now commonplace in many diverse fields, including economics, sociology, and political science [52].

The motivation for the research of von Neumann *et al.* was not to derive winning strategies for poker *per se*; rather they had the wider aim of seeking to understand which strategies should be selected by rational, reward-maximizing individuals when the potential payoffs to their actions are dependent upon the choices of competing players.

One example of early poker research was that conducted by Kuhn [80] on a simplified two-player poker. The game requires each player to ante one unit, before receiving a single card from a deck of three uniquely numbered 1, 2 and 3. In the single betting round the players must either bet one unit or pass. If a player passes after his opponent has bet the opponent wins the pot. Two successive bets or passes lead to a showdown, with the highest numbered card determining the winner.

The benefit of choosing such a reduced form of poker is that it makes the number of possible plays and strategies far smaller than that of any real variant. Using linear programming [119] the resulting game matrix is easily solved to produce optimal strategies for both players. The results take the form of mixed strategies, such that a player should randomly employ a specific weighting of his available pure strategies.

This simple form of poker is typical of the toy games that have formed the test bed for much poker research. Borel's poker [19] and von Neumann's poker [120], both of which pre-date Kuhn's work, use real numbers from the unit interval as "cards". As with Kuhn's poker, limitations are placed on the available betting actions to facilitate tractability. Recent work co-authored by a world championship winning poker player [44, 45] extends the Borel model to solve games with larger bet sizes, multiple betting rounds, and more elaborate betting possibilities.

Research which examines extensions to such toy pokers is common in the literature. Additional game features are routinely incorporated to make the variant more complex and often more realistic. Cutler's poker [34] allows for pot-limit betting, Karlin and Restrepo [70] introduce a model incorporating multiple betting rounds, and Nash and Shapley [89] investigate a three-player game. Gillies *et al.* [53] calculate solutions for poker variants mentioned in the work of von Neumann and Morgenstern [120].

Interestingly, some of the results from the studies on toy pokers have highlighted strategies which are applicable to real poker variants such as Texas hold'em. For example, the optimal strategy for the first player in von Neumann's poker reveals that he should bet with both his very best and very worst hands. Betting with a hand which has no possibility of winning in a showdown is referred to as *bluffing*. This action creates greater uncertainty of a player's cards in his opponent's mind, and can often win pots with an inferior hand through the opponent folding. Bluffing is seen to be used in all regularly played forms of poker.

The major problem in applying game theory to real poker variants has been due to the size of their game trees. Given the combinatorial number of possible deals and betting actions over multiple rounds, it is calculated that the number of nodes in two-player Texas hold'em is over 10^{18} [8]. For the multi-player game this number will be palpably far larger.

More than fifty years after the initial mathematical investigations, much of the present advances in poker research employ game theory on larger and more realistic forms of the game. This work has been made possible by increases in computing power. Prior to looking at this we discuss some of the other computational techniques that have been employed by poker researchers.

2.5 Computer Poker Research

Several different methods have been used in the attempt to develop strong computer poker players. The first to use computers in poker research was Nicholas Findler [46, 47]. A simplified form of draw poker was used for testing the play of humans against intelligent machine systems that played according to heuristics and simple statistical features. The primary purpose of Findler's study was to understand how humans make decisions under conditions of uncertainty, for which poker was simply the test bed. The computer players he designed were overly simplistic to be practically competitive, and their betting behaviours too revealing.

A Bayesian network-based approach [77] has been used to model five-card stud poker. This methodology enabled the computer player to learn the game and its single opponent's strategy by continual updating. The results of this research were mixed, and whilst the program had some success playing against simple probabilistic and rule-based computer competitors, it was found to be deficient against human opposition. It appears that the hand classifications used were too broad, and that the updating was unable to adapt to deceptive opponents.

Following the previously mentioned success of Tesauro in applying a reinforcement algorithm to backgammon, Dahl [35] investigated the use of these techniques to a simplified two-player Texas hold'em game. In this work he explains that a value-based approach such as TD or Q-learning is not applicable to an imperfect information game such as poker, since the value of the game cannot be estimated with sufficient accuracy during play. Instead he applied a gradient search-based reinforcement learning algorithm to a much simplified poker variant. This method created a player which was seen to learn and improve against its opponents, but applying this technique to the far larger search space of a full-scale poker will require much greater research efforts.

Daphne Koller and Avi Pfeffer [75, 76] created a new algorithm for finding optimal randomized strategies in two-player imperfect information games, such as poker. Their *Gala* system represents games in sequential form, an alternative to the commonly employed matrix (normal form) and graphical (extensive form) representations. *Gala* is able to specify and solve a larger class of problems than was previously achievable. They report in their research, unfortunately, that the size of the game tree in full-scale poker is still beyond the capabilities of this system.

The most advanced and successful work on computer poker play to date has been produced by the Game-playing, Analytical methods, Minimax search, and Empirical Studies (GAMES) Group at the University of Alberta. Jonathan Schaeffer's group have been responsible for some of the most important advances in computer games player development generally, and are now focusing much of their efforts towards poker.

The GAMES Groups' poker research started with the development of *Loki* [13, 14, 15, 16, 37, 94]. This program is designed to play ten-player limit Texas hold'em in a ring game, and employs a combination of statistical measures and expert rules to effect decision making. The *Loki* system was re-written in 1999, and christened *Poki*. In this version the opponent modelling mechanism was changed from using statistical updating to a neural network [38, 39]. *Poki* also incorporates a more advanced decision making procedure, with selective-sampling providing the ability to search the game tree in real time. *Poki* still determines its betting actions with reference to an expert-defined rule base, but it selects its action randomly from a distribution over each possible action. This means that *Poki* is less predictable than its predecessor.

At the time of their development, *Loki* and *Poki* were arguably the most advanced computer poker players. Their reliance on expert rules, however, could be said to lessen the achievement of the GAMES Group in these systems. It may be that some preconceived strategies are actually incorrect, and thus handicap the computer player's abilities. Techniques which assume no prior knowledge have also been applied to computer poker, and will be examined thoroughly when we review the application of evolutionary computation to poker in Chapter 5.

2.6 Combining Game Theory with Learning

The end of the last century saw a return to game theoretic investigations of poker, due to the advances in computer technology. Alex Selby [106] was the first to successfully apply game theory to a variant of Texas hold'em. In his work, Selby solved the two-player pre-flop limit version of the game. This restricts the game to a single betting round, after which the remaining community cards are dealt out and the winner determined. The size of the resulting game tree is small enough to be mathematically tractable. The solution takes the form of a table of mixed strategies for each player depending on their starting cards and how their opponent bets.

Shi and Littman [107] built upon the work of Selby to produce the optimal strategy for a multi-round variant of Texas hold'em, which they christened Rhode Island hold'em. Their approach reduced the size of the two-player game tree by utilizing several abstraction and bucketing techniques. The resulting models were then small enough to be solved by linear programming, with the solutions interpreted in the context of the original game.

The success of this work led to the most significant recent innovations in deriving optimal strategies for a real poker variant by Billings *et al* [8] at the GAMES Group. Billings applied Shi and Littman's reduction techniques to two-player ring game limit Texas hold'em, to derive what they term "pseudo-optimal" strategies within the program *PSOpti*. This program was tested against a world-class poker player with remarkably good results. Eventually the human player was able to find weaknesses in *PSOpti*, and adapted to beat it. Although unable to compete at the very highest level, tests against weaker human opposition have confirmed the general strength of *PSOpti*'s play.

Whilst this effort of the GAMES Group is highly noteworthy, the benefits to learning the optimal strategy for any given game are limited. In game theory, an optimal strategy in a two-player zero-sum game is one for which neither player can increase his reward by unilaterally deviating. Knowing the optimal strategy in a game only provides the best worst-case across all the opponent's possible strategies. If the opponent also plays his optimal strategy, the game rests in equilibrium. If the opponent deviates, though, there may be a chance to increase the expected payoff by also playing non-optimally. By considering the opponent's original deviation to be a strategic mistake, a sub-optimal reply which increases the expected payoff is an exploitation of this mistake. Hence to play poker profitably it is not necessarily an optimal strategy that is required, but a maximal one. This is a strategy which maximizes its payoff against all the opponents' strategies. Determining a maximal strategy requires the use of opponent modelling.

The need for opponent modelling in games of imperfect information is highlighted in the creation of computer players for RoShamBo (commonly referred to as "Rock, Paper, Scissors"). In this game, two players must simultaneously select from the choice of the three objects, and an intransitive relation amongst the possibilities decides the winner: rock beats scissors, scissors beats paper, and paper beats rock. The optimal solution for this game is to select all three actions randomly and with equal probability, and no counter-strategy is able to defeat this play in the long run. The drawback to this approach is that in a pairwise league competition between many players, the optimal strategy will only produce middling results. It will score averagely well against all opposition, whereas the programs at the top of the league will have been able to exploit most of their opponents' weaknesses.

An optimal strategy is sub-maximal if one's opponent is playing sub-optimally. To illustrate this, suppose that a particular RoShamBo player selects paper with certainty. Then the optimal strategy will equally win, lose and draw one third of the time. An alternative program which always selects scissors against this "always paper" strategy will ensure the maximal result. The winner of the First International RoShamBo Programming Competition [9], *Iocaine Powder* [42], was designed to exploit its opponents where possible, only reverting to the optimal strategy as a default.

The rationale behind knowing the optimal strategy for a game is that it first creates a "no lose" strategy. For the program to produce better results it needs to be able to model its opponents and exploit their mistakes. So as Billings correctly points out [8], knowing the optimal strategy for a given poker variant forms a strong foundation upon which to build. The creation of a maximal player from an optimal one centres wholly on opponent modelling. By learning and understanding the opponent's strategy, the program will then be able to adapt its play and improve its results.

With this in mind, the GAMES Group followed the creation of *PSOpti* with the program *VexBot* [12]. This newer model is also designed to play two-player ring game limit Texas hold'em. The methodology behind *Vexbot* improves upon the solely game theoretic *PSOpti* by incorporating real time opponent modelling into its game tree search. The superiority of this approach was demonstrated in *Vexbot*'s defeat of *PSOpti* in the 2003 Computer Olympiad [10]. Since *PSOpti*'s strategy is not precisely optimal, *Vexbot* was able to identify and exploit weaknesses in its play. The results from this methodology are extremely encouraging, and it would seem likely that further advances in this direction (for example, by allowing for multiple opponents) would produce even more impressive results.

Recent efforts by the GAMES Group specifically target the area of opponent modelling [65, 116]. There are many difficulties in achieving adequate learning during the play of poker. The critical issue is how to model an opponent after only a few hands of play, when the space of all possible hands is vastly larger. Another challenge is in maintaining an up-to-date model when the opponent could be switching strategies mid-game.

If a game of poker can be divided into the components of a fundamental mathematical understanding plus opponent modelling, then the latter is undoubtedly a great obstacle to creating a world class program. However, the benefits of a highly competent opponent modelling function are lessened without a solid comprehension of the underlying physics of the game. Given that two-player limit Texas hold'em is now almost solved, attention for this specific variant has rightly turned towards opponent modelling. For other forms of poker, such as no-limit betting and tournament play, the foundational groundwork has yet to be established academically.

2.7 Non-Academic Poker Literature

Several books have been written on correct strategy for play in ring game limit Texas hold'em [22, 26, 31, 63, 64, 68, 83, 99, 121]. Due to the complexity of the game, all poker texts contain overgeneralizations in the advice offered. The guidance put forward is almost entirely based on anecdotal evidence, with the possible exception of one author. David Sklansky's books [108, 109, 111], are generally the most scientific in their approach, and often include brief sections on the application of game theory to idealized poker situations.

Much of the non-academic poker literature sets out the criteria that a hold'em player should use in deciding upon his strategy once the hole cards have been dealt. Hilger [64], for example, lists amongst other factors:

- Strength of starting cards
- Position
- Number of callers
- Raised/unraised pot

The last two factors can be grouped together into knowledge of the opponents' prior betting actions. Hand strength, seating position, and the opponents' prior actions are the three factors which are most commonly heralded as being important for basing betting decisions on in Texas hold'em. In the following sections we examine each of these in more detail.

2.7.1 Hand Strength

In the long run all players receive exactly the same distribution of starting hands in Texas hold'em. Whilst it is often observed that any two cards can win an individual hand, some of these figure to be profitable if played regularly, and others not. The skill in poker is in playing the right hands for a particular situation, and folding the rest. As Hilger [64] puts it:

> Winning players play mostly strong hands while losing players play both strong and weak hands.

In a hold'em showdown the highest ranking hand wins. Therefore the potential for success with two given hole cards must be assessed. Outside of the context of a particular situation, all starting hands have a certain intrinsic value. Many poker authors have sought to simplify this evaluation by grouping together hands of similar strength [26, 64, 68]. In formulating his hand strength groupings, for example, Sklansky [108] used these six interrelated criteria:

- What are the chances of making the best hand?
- What are the chances of making a flush? (Are the cards suited?)
- What are the chances of making a straight? (Are the cards close to each other and in the middle ranges?)
- What are the chances of flopping top pair[2] (or in the case of a pair in the hole, what are the chances that no overcard[3] will fall)?
- What are the chances of making a hand that figures to win a big pot (because the players will tend to make second best hands)?
- What are the chances of making a hand that might well just lose money since it will be second best?

He assigns 72 of the 169 different starting hands to one of eight groups, with the remainder deemed unplayable.

Sklansky then explains that, in the context of a specific situation, a hand's true value is heavily dependent upon other game factors. The two principal considerations

[2] Flopping top pair means that the highest ranking card on the flop matches one of the player's hole cards, thus making a pair.

[3] An overcard is a card higher in rank than that paired in a player's hole cards.

that he and all other authors cite are one's seating position and the opponents' prior actions.

2.7.2 Seating Position

The seating position of a player relative to the dealer is said to be a highly significant factor in the decision making process, according to poker professionals. Indeed, Miller's [83] opinion is that:

> Position is the second most important component of hand value (your own and your opponents' cards are the most important).

The seating position of players in a hand of Texas hold'em takes on increased noteworthiness since the order in which players act remains invariant, unlike some other poker games. The players to the immediate left of the dealer are said to be in *early* position. The first two such players must post the small and big blinds respectively, with the active betting in the pre-flop round commencing with the third around from the dealer. In the three later betting rounds the betting again starts with the first active player to the dealer's left. The dealer himself and the players to his right are classed as being in *late* position.

Poker authors contend that there is a benefit to being seated in late position. The more betting rounds there are, the greater the advantage. Hence with four betting rounds, seating position in Texas hold'em is exceptionally important. Late position is said to confer an informational advantage, since a player seated there is able to act based upon knowledge of how those seated in front of him have played. An early position player suffers from the opposite disadvantage.

2.7.3 Opponents' Prior Actions

The non-academic poker literature regularly states that a hand's worth is highly dependent upon the betting actions that have already occurred. For example, low pairs are often said to be worth playing after several callers, but not just one or two. Other hands, like high offsuit cards, are believed to lose value with every additional opponent. A player who observes his opponents' actions is able to use this information in evaluating his hand's strength and suitability for that particular situation.

The most common message from the professional players relates to the reduced number of hands that should be played once an opponent has entered the pot, particularly with a raise. According to Sklansky [111], after an opponent has raised:

> You should usually reraise or fold, with folding being much more prevalent.

His sentiment is echoed by Jones [68], who simply states that:

> You need a stronger hand to play in a raised pot.

According to the literature, the rationale behind this is simple. The most likely reason for an opponent to bet in poker is because he has a strong hand. Therefore one must always be more wary of playing a hand in an opened pot. It is safer to release marginal starting cards than play them after a raise, particularly for non-expert hold'em players. The expected value of a fold is zero, and this is frequently the maximum of all the available betting options when acting behind a raiser.

Note, that the consideration of one's opponents' prior actions is closely tied to the preceding factor of seating position. The later the seating position of a player, the more of his opponents' prior actions he can observe.

2.8 Summary

Research into correct play for ring game poker has advanced substantially over the last half-century, benefiting from increases in computational power. The early investigations by the founders of game theory on toy pokers have now been updated to address real poker variants. Different computational methods have been applied with the aim of developing strong poker programs. Many of the resulting programs now contain highly accurate game theoretic strategies, and work is advancing on opponent modelling techniques to improve their play even further.

Almost all academic research on Texas hold'em has centred on limit betting in a ring game environment. None of the studies published to date have been directed towards the tournament form of poker, and none have addressed the no-limit form of the game. These are striking omissions, given that the most prestigious events in poker are played within a no-limit tournament setting. Clearly any researcher wishing to develop a player which is able to compete for the world championship will need to frame their program within the no-limit tournament context.

The most palpable reason for the lack of progress into no-limit tournament poker are the additional complexities to those needed to produce a limit ring game player. Firstly, in limit betting each player need only decide whether to fold, call or raise at each turn. If a player raises in big bet poker, he must also decide by how much.

Also, the dual objective within tournament poker of both winning pots and survival poses a harder challenge to researchers than creating a ring game computer program. Since tackling the real life complexities of a no-limit Texas hold'em tournament is non-trivial, we seek to simplify the task whilst keeping sufficient features of the game intact.

In the next chapter we address the strategic issues surrounding tournament Texas hold'em more closely, and look at game simplifications which make experimental work feasible.

Chapter 3

All in or Fold Texas Hold'em Tournament Strategy

In this chapter we elaborate on tournament poker, explaining the rules and structure of such a competition. Following this we consider the differences between tournament poker and the ring game, and consider how these distinctions impact on a player's betting actions. Next we give an example of a complete strategy for no-limit Texas hold'em tournaments, as set out by poker professional and theorist David Sklansky. We then show how we used his all in or fold system to compete in the ICCM 2004 PokerBot World Series event against more complex computer poker programs.

3.1 Texas Hold'em Tournament Rules and Structures

In a Texas hold'em ring game players bring real money to the table in the form of poker chips. The betting structure is fixed, meaning the blind amounts do not alter throughout the duration of the game. It is common to see a spread of ring games offered both in live play and online, with the price level often denoted by the sizes of the small and big blinds. At a typical internet poker site Texas hold'em ring games can vary in price level from $0.05/$0.10 up to $1,000/$2,000 [1]. This spread allows players to find a game which is in keeping with their bankroll and, since better players typically play for higher stakes, their ability.

The competition in a ring game is limited to the number of people who can be seated around the table: typically ten. A ring game player is free to exit at any time, whether in profit or loss. If he loses all his chips he must either leave the table or purchase more chips to continue playing. A ring game has no fixed end point and players frequently drop out of the game to be replaced as seats become available. The profit or loss of a ring game player is simply the aggregate result of wins and losses on every hand played.

Poker tournaments employ the same underlying game rules to determine the play of each hand, but differ in their overall structure. Each competitor buys-in for a given starting stack amount, but unlike in a ring game the chips are not a direct replacement for cash and have no value outside of the tournament. So for example, a competitor entering a tournament with a $50 entry fee may actually receive $5,000 in tournament chips. Note that such a player stands only to lose their original $50.

Tournament entrants must continue playing until they have either lost their stack or have won everybody else's[1]. Tournaments can be played with any number of players, from two up to several thousand. Tournaments with more than ten entrants involve the use of many tables. The number of seats around each is limited, but as individuals get knocked out continuing players are taken from other tables to fill the gaps. In this way the number of active tables in a tournament is eventually reduced to a *final table*.

To ensure an expedient finish, tournaments employ a rising blind schedule. This means that after either a specified number of hands or a given time interval the size of the blinds increases according to a pre-specified programme.

The payoff in a tournament is determined by the reverse order in which players lose all of their chips. Hence those players who survive longest finish highest. Normally a player must finish in the top 10-20% of the competitors to receive a prize. Most tournaments use the progressive payback method in which the higher placed finishers receive an increasingly larger share of the prize money.

All of the above discussion on the structure of a poker tournament, and the differences with a ring game, are summarized in Table 3.1.

Following this we present an example of a typical low-stakes limit hold'em tournament in Table 3.2. In this tournament it is supposed that 47 competitors each pay $20 in real money and start with $1,000 of tournament chips. A progressive payback applied to this same tournament shows the potential payoffs to the entrants in Table 3.3.

The structural differences between tournament and ring game poker lead to distinct objectives for players in each event. Paramount amongst these differences are the contrasting ways in which players win or lose money. In a ring game money is collected

[1] The exception to this is a rebuy tournament, which allows players to purchase additional chips during a prescribed opening period of play. For the remainder of this thesis we assume a tournament which does not allow rebuys.

Game Feature	**Tournament**	**Ring Game**
Entry Fee	Fixed tournament cost	Variable
Chips	Game tokens	Real money replacement
Blinds	Rising schedule	Fixed
Number of Players	Unlimited	Limited by table seats
Game exit	No more chips remaining	Player discretion
Profit and Loss	Based on finishing position	Determined on each hand

Table 3.1: The structural differences between tournament and ring game poker.

Duration	**Blind Sizes**
15 minutes	$15/$30
15 minutes	$30/$60
15 minutes	$50/$100
15 minutes	$100/$200
15 minutes	$200/$400
Interval	*Interval*
15 minutes	$300/$600
15 minutes	$500/$1,000
15 minutes	$1,000/$2,000
15 minutes	$2,000/$4,000
15 minutes	$3,000/$6,000

Table 3.2: Typical limit tournament poker structure (from [117]).

Place	Prize Money	Percentage
1^{st}	$376	40%
2^{nd}	$216	23%
3^{rd}	$113	12%
4^{th}	$85	9%
5^{th}	$56	6%
6^{th}	$47	5%
7^{th}	$28	3%
8^{th}	$19	2%

Table 3.3: Typical limit tournament poker payout structure (from [117]).

or forfeited with every hand played. This means that it is almost always correct[2] to choose the betting decision which has the highest expected value (EV) amongst the alternatives in ring game play. In this way it is often said that a ring game player's long term winnings are simply the accumulation of the EV for each decision he makes.

Whereas the basic structural unit of a ring game is an individual hand, a tournament is the unit in itself. Money is only collected for finishing in the prize places in a tournament. In this format each player must weigh the EV of their decisions in terms of tournament chips with their EV in terms of prize money.

3.2 Strategic Factors in Tournament Poker

Players are continually forced to win chips to survive in a tournament. This is because as a tournament progresses the players are regularly required to post blinds, which rise according to a schedule. The process of losing chips through the continual posting of blinds is known as being *anted-away*. It is not usually enough, however, to simply aim to survive in a tournament. Players also need to accumulate chips to ensure a greater chance of finishing in the prize money. The dual objectives of survival and chip accumulation make tournament play strategically more complex than that required in a ring game.

There are many ways in which these objectives manifest themselves in a tournament player's actions. To ensure survival a player might eschew positive EV on a hand

[2] The exception to this is due to meta-game considerations, i.e. one may make a sub-maximal play to deceive one's opponent in the expectation that it will lead to greater winnings in a future hand.

if he believes that it maximizes his expected prize. Contrariwise, a player with only a few chips may choose to bet an inferior hand and accept a negative EV if he believes it improves his chance of receiving prize money over folding. Generally speaking, competitors wishing to protect their chips and survive will play more conservatively than in the identical ring game situation. Likewise, a player wishing to accumulate chips will pursue a more aggressive and potentially riskier course of action.

Texas hold'em literature for ring game play abounds, as discussed in Section 2.7. However, a lesser amount has been specifically written for tournament strategy [24, 60, 61, 81, 110, 114, 117]. The reasons for this may be due to the fact that tournament poker is only a relatively recent phenomenon in the United States, from where the vast majority of books are authored, or it may simply be due to the increased complexity of tournament play.

Since an individual hand within a tournament uses exactly the same rules as those of a ring game there are very clear similarities between the nature of successful strategies in both formats. The factors of hand strength, seating position, and opponents' prior actions are still as important in the decision making process. However, the non-academic tournament poker literature contains many illustrations of how overall tournament strategy differs from that required for a ring game. Hilger [64] states that:

> You will need to make many adjustments to your ring game in tournaments to consistently end up in the money.

This advice is repeated by Reuben and Ciaffone [99], who claim:

> Playing in pot-limit and no-limit money games does not completely prepare a person for tournament play. A formal competition has somewhat different strategy, and vastly different psychology.

Within the literature there are three commonly cited factors believed to influence correct tournament strategy, in addition to those discussed for ring game play. These are the presence of rising blind levels, the importance of stack size, and a phenomenon known as the "Gap Concept". These are explained in turn in the following sections.

3.2.1 Tournament Level

The style of a poker player can be crudely evaluated with reference to two complementary scales. One measures how many hands a player plays, whilst the other gauges how often a player bets and raises.

A player who bets with a great many poker hands is said to be *loose*. At the other end of the scale a *tight* player is one who frequently folds. There are several ramifications depending on the looseness of players in a poker game. A competitor who is very loose is likely to be playing hands which are weaker on average than his opponents. Therefore he bears a greater risk of facing an opponent with a stronger hand. However, when a loose player himself holds a very strong hand his previous frequent betting action often proves to be a profitable disguise. Also, a loose player may pick up many pots through his constant betting.

A very tight player usually only bets with *premium* hands. These are the highest ranking two-card holdings. The deficiency with this strategy is that opponents are usually quick to notice when a player frequently folds, so that when such a player does bet they signal a strong hand. The clever tight player can then use this to their advantage by occasionally betting with weak hands, and picking up pots unopposed.

The second scale on which poker players are often measured relates to their propensity to bet the hands they play. An *aggressive* player will typically bet and raise, whereas a *passive* player prefers to check and call. Aggressive players like to take a lead in the betting action. They reason that betting gives them an extra way to win a hand that calling does not: their opponents may fold. Almost all professional poker authors advocate a tight-aggressive strategy for ring game poker. That is to say, a player should not play many hands, but when they do they should invariably bet and raise with them.

With specific reference to tournament poker, the non-academic literature also counsels the tight-aggressive approach, but with one significant adjunct. This rider is the necessity for increased looseness and aggression as a tournament progresses. Poker authors maintain that a player should be very tight and cautious in the early stages of a tournament, and gradually become less tight to take more risks as play proceeds. Since the initial blinds are very small as a percentage of the starting stacks, there is little incentive to try to win pots with all but the very best of starting hands. Once the blinds escalate, players must start to win pots more frequently to survive and hence their starting requirements should be lowered.

Likewise, players in the later stages of a tournament should be more willing to bet and raise their hands, rather than simply call their opponents' bets. The greater chance of causing opponents to fold and steal blinds by becoming both looser and more aggressive is believed to offset the increased chance of tournament elimination by coming up against a stronger hand.

Harrington [60] sums this up by saying:

> As long as the cut-off point for prize payouts is distant, play proceeds normally. As the number of players shrink and gets close to the prize cutoff, play changes dramatically. Good players become more aggressive.

3.2.2 Stack Size

According to the poker literature, chip stack sizes are important in both ring game and tournament play, particularly in big bet games. Stack sizes take on even greater significance in tournaments due to the fact that players cannot replenish their chips. Once a player loses all of their stack they are eliminated from the tournament.

In no-limit poker the correct betting decisions are often said to be determined by the amount of chips the active players hold. This effect is most noticeable towards the end of a tournament. For example, a player with a sizeable chip stack at the final table is often wise to avoid confrontations with the other large stacks. The reasoning is that, although the player may have a positive EV on the hand being played, the downside to losing the hand and missing out on larger prize money often makes it correct to fold.

In a comprehensive book written on limit Texas hold'em tournaments, Buntjer [24] details what he believes should be the overriding strategic concerns in a player's decision making at different stages of a tournament. These weightings are reproduced in Table 3.4.

Phase	**Starting Hand**	**Playing Position**	**Player Type**	**Your Image**	**Opp. Stack**	**Your Stack**
Early	60	20	10	10	0	0
Middle	20	15	30	10	10	15
Last 2-3 tables	15	15	15	15	15	25
Last table	20	20	15	5	20	20
3-4 players	5	10	10	5	30	40
Heads-up	5	0	5	10	40	40

Table 3.4: The relative importance of strategic factors in a limit Texas hold'em tournament (from [24]).

The conclusions to be drawn from this include:

- one's starting hand begins as the most important reason to play a hand in the early stages of a tournament, but then becomes almost irrelevant towards the end,
- chip level, both one's own and one's opponents', begins as irrelevant but gradually becomes the most important factor in the play of a hand

A player with a large stack benefits in at least two ways. Firstly, he is able to wait until he receives strong cards before playing a hand, since he does not risk being anted-away. Secondly, he may choose to habitually raise his opponents, even with weak hands. This tactic can prove to be profitable since an opponent may then prefer to fold a mediocre hand rather than play back, realizing that their tournament survival could be at stake. The big stacked player may thereby pick up many uncontested pots.

The situation facing a player with a relatively short stack is often critical. Such a scenario often dictates that the player should be willing to bet with any and all hands, especially if the pot is unopened. Although they may face elimination if called and beaten, the potential for success makes it preferable to the alternative of being anted-away. One of the many skills of a strong poker player is determining the point at which a bet with a weak hand should be attempted.

A series of three books detailing no-limit Texas hold'em tournament strategy has been written by Dan Harrington [60, 61, 62]. The author is one of the most successful poker players of all time, having won the World Series of Poker main event in 1995 and reaching the final table in both 2003 and 2004. Harrington introduces many new concepts within these books.

In the second of his trilogy [61] he comments that:

> The most important single number that governs your play towards the end of tournaments is M, which is simply the ratio of your stack to the current total of blinds and antes...As your M drops, your play needs to get more and more aggressive.

He then develops strategies which depend upon the different possible values of this number.

3.2.3 The Gap Concept

The need to preserve chips and avoid confrontations in a poker tournament contributes to the Gap Concept, a term coined by David Sklansky [110]. He describes his informal theory thus:

> You need a better hand to play against someone who has already opened the betting than you would need to open yourself. The difference between the hand you need to call an opener with, and that with which you would open yourself, I call the Gap... In a tournament this Gap is often extremely high... As important as the Gap Concept is anytime in a tournament, it becomes more important still with the Gap usually widening even more during the last stages of the tourney.

So for example, a player with a relatively weak holding of a pair of sevens seated halfway around the table may be correct in betting into an unopened pot, but they would require a stronger hand (say, a pair of jacks) if an opponent has already bet. The "gap" is the difference between these two hands. The Gap Concept also states that this gap typically increases as a tournament progresses. Hence as a tournament progresses weaker hands are sufficient to open a pot, but even stronger ones are required to call or raise a prior bet.

A corollary of the Gap Concept is that it is seldom correct to open the pot with a call (also known as *limping*) in a poker tournament. This is especially true in the later stages when the blinds have risen. The reason for this is that a raise will very often win the blinds, since the remaining players will fold many more hands to a raise than they would to a call.

3.3 Sklansky's *System*

In the same book that introduced the Gap Concept, David Sklansky reveals the basis for a simplistic strategy for no-limit tournament Texas hold'em. A casino owner had entered his daughter into the $10,000 World Series of Poker main event, but she had no previous experience of playing the game. The owner approached Sklansky one week before the start of the tournament and asked him to help his daughter learn how to play. Given her lack of proficiency and the short time available for training, Sklansky developed what he termed *The System*.

Sklansky's instructions were simply:

- If someone has raised in front of you, move all in with aces, kings, or ace-king suited. Otherwise fold.
- In no one else has raised in front of you, move all in with any pair, any ace-other

suited, ace-king (suited or offsuit), or two suited connected cards[3], except for four-three or three-two.

- Do not play any hands in the first few rounds of the tournament.

There were two main reasons behind the given strategy. Firstly, Sklansky wanted to keep the specifications as simple as possible since the lady had never played poker before. He felt that the more complexity he put into the decision rules, the bigger the chance that she would err. The second basis for playing in such a way is that it would make her extremely difficult to play against. By committing all of one's chips before the flop, it removes the necessity to act on the final three betting rounds. Experienced tournament poker players can use these rounds to make improved inferences about their opponent's likely holdings. Sklansky's *System* negates this advantage.

Since there are 1,326 possible two-card starting hands, the *System* advocates an all in move with 13.1% of these (see Table 3.5). With a ten-player table, an individual needs to win the blinds at least 10% of the time to avoid being anted-away. Assuming an average distribution of cards, a player who employs the *System* will therefore be competing in a sufficient number of hands.

Hand	**Combinations**
Any pair	78
Ace-king (suited or offsuit)	16
Any ace-other suited	44
Two suited connectors (except 43 and 32)	36
Total	174

Table 3.5: Playable hands in Sklansky's basic *System*.

The downside to employing such a strategy is that at some stage its user will be faced with an opponent who is prepared to call an all in bet. If the opponent has a larger stack then the *System* player must win the hand to ensure survival. Unfortunately for the casino owner's daughter she found herself in this position against an opponent holding a pair of aces, and she was eliminated towards the end of the first day.

[3] Suited connected cards, also referred to as *suited connectors* or *zero-gap* hands, are two cards of adjacent rank in the same suit. By extension, *one-gap* and *two-gap* hands are those whose cards are two and three ranks apart respectively.

Sklansky's book next includes a chapter entitled "Improving *The System*". The author points out the two major flaws in the simple strategy expounded:

- It did not take into account stack size versus blind size.
- It did not take into account how many players were yet to act.

The first of these points was behind Sklansky's instruction not to employ the strategy at the start of the tournament. His reasoning was that the daughter would be risking losing all her $10,000 stack for blinds totalling only $75. The risk/reward ratio in such a case is too unfavourable. The second flaw with the original *System* comes from realizing that it is far safer to make an all in bet at a poker table if there are less players to act behind oneself. The fewer the number of remaining players, the lower the chance that somebody has a playable hand sufficient to call a bet.

Sklansky seeks to ameliorate his strategy by addressing these two defects. The foundation of the improved *System* remains the same: the user will either go all in or fold in the pre-flop betting round. If an opponent has already raised the pot then the improved *System* again dictates that a player should re-raise all in with a pair of aces, a pair of kings, or ace-king suited. Otherwise one should fold. With no prior raise Sklansky bases his decisions around a "key number". To calculate this value, a player must first divide the total amount of the blinds into his stack amount. The resulting figure is then multiplied by the number of players yet to act (including the blinds). With no prior players in the pot, the calculated key number stands. With n preceding callers, the key number should be multiplied by $(n+1)$.

The rules determining the acceptable all in starting hands in the improved *System* after a prior raise are given in Table 3.6. All other hands should be folded.

As an example of determining play using the improved *System*, suppose that the blinds are $160 and $320, and we are dealt $K\heartsuit J\clubsuit$ with a stack of $8,400. One player has already called the big blind, and there are two players left to act after us. Here, the key number is calculated to be:

$$k = \tfrac{8400}{160+320} \times 2 \times (1+1) = 70$$

From Table 3.6 our hand requires a key number of less than 60, and so we should fold. Note, however, that without the limper an all in would be appropriate:

$$k = \tfrac{8400}{160+320} \times 2 \times (0+1) = 35$$

Key Number	Playable Hands
≥ 400	AA only
[200, 400)	As above plus KK
[150, 200)	As above plus QQ, AK
[100, 150)	As above plus JJ, TT, AQ, KQ
[80, 100)	As above plus any pair, any ace suited, zero-gap suited down to 54
[60, 80)	As above plus any ace, any King suited, any zero- or one-gap suited
[40, 60)	As above plus any king
[20, 40)	As above plus any two suited cards
< 20	Any two cards

Table 3.6: Sklansky's improved *System*.

Sklansky does not provide any assessment of the superiority of his improved *System* over its predecessor, but by the nomenclature employed he clearly assumes that the extra information incorporated into the decision rules increases its strength.

Wishing to gauge the strength of the improved *System* in a recent computer poker competition we took the opportunity to encode and enter it into the ICCM 2004 PokerBot Tournament.

3.4 ICCM 2004 PokerBot Tournament

In July 2004, the International Conference on Cognitive Modelling [93] held the "PokerBot World Series" in which entries were solicited to play in a series of no-limit Texas hold'em tournaments between computer players. The purpose of the competition was to study and evaluate cognitive models within an environment which is simple yet representative of many real world situations. The organisers listed the cognitive abilities required for playing strong poker as:

- Reasoning under uncertainty (one's opponents' cards)
- Dealing with probabilistic outcomes (the future cards)
- Decision making with multiple options (the chips used for bets)
- Individual differences (different styles of play)

- Inference of intent (from opponents' bets)
- Intentional deception (bluffing)
- Pattern recognition (detecting trends from the flow of the game)
- Social and emotional aspects (dealing with winning and losing)
- Economic behaviour (factoring the impact of amount of bets)

The tournament administrators specifically chose computer poker as their experimental domain because it provides a challenging problem covering a broad range of cognitive abilities, yet remains more tractable than an environment based on human interactions.

3.4.1 Tournament Rules

Entries were required to conform to a pre-specified protocol, and were accepted in the form of an executable program. Each competitor commenced a no-limit Texas hold'em tournament with $10,000 in chips. The blinds started at $10 and $20, and doubled every 100 hands. The response times of each entry were limited to an aggregate of 100 seconds per 100 actions to ensure fairness. The winner of a tournament was simply the player who won all of the chips. The results of the event were aggregated over 104 such tournaments.

3.4.2 Entries

Unfortunately the response to the tournament invitation was poor, and from 50 initial expressions of interest only five working programs were entered. The reason for the low numbers was believed to have been due to the restrictive time between the conference notification and the tournament date. The five computer players entered, and their programmers, were:

- *Ace_Gruber* - Stanislav Sokorac (University of Toronto)
- *Carleton* - Terrence C. Stewart and Robert West (Carleton University)
- *Dbot* - Dan Bothell (Carnegie Mellon University)
- *DumbBot* - Richard Carter (University of Edinburgh)
- *YesterdaySushi* - Maxim Makatchev (University of Pittsburgh)

3.4.3 *DumbBot*

Our entry into the PokerBot competition, *DumbBot*, encoded the improved *System* set out by David Sklansky explained previously. Since these decision rules were designed for full-table play against nine opponents, an adjustment was made for play against three or fewer opponents. This was done to accommodate the supposed requirement for playing more hands later in a tournament, discussed in Section 3.2.1.

When entering an unraised pot in such a situation, *DumbBot* would play as per Sklansky's instructions. However, if the pot had already been raised before its turn, *DumbBot* would perform a hand evaluation and only go all in if it possessed a sufficiently strong two-card starting hand.

To perform the evaluation, the hole cards were compared to a ranking of all 169 possible starting hands and its percentile placing noted. The naive listing used ranks all hands based on their expected win percentages against each other. The minimum rankings required for an all in move dependent upon the number of opponents were chosen to be:

- 3 opponents: 0.90
- 2 opponents: 0.85
- 1 opponent: 0.80

The selection of these values was arbitrary, but again made with the desire to play more hands against fewer opponents.

3.4.4 Results

The overall winner of the PokerBot tournament was Stanislav Sokorac's *Ace_Gruber*. It won 44 of the 104 tournaments, placing it comfortably above all other entries. The number of tournament wins within the competition are shown in Figure 3.1.

The strategy that *Ace_Gruber* implemented was the most complex of all the competitors. The program was based on the architecture of the University of Alberta's *Loki* and *Poki*. It enumerated over the possible future cards in a hand, and was able to evaluate its present holding in terms of how the hand might develop. A superior inclusion in *Ace_Gruber*'s encoding was its opponent modelling. The program maintained probability tables for each opponent, representing the likelihood that each competitor held

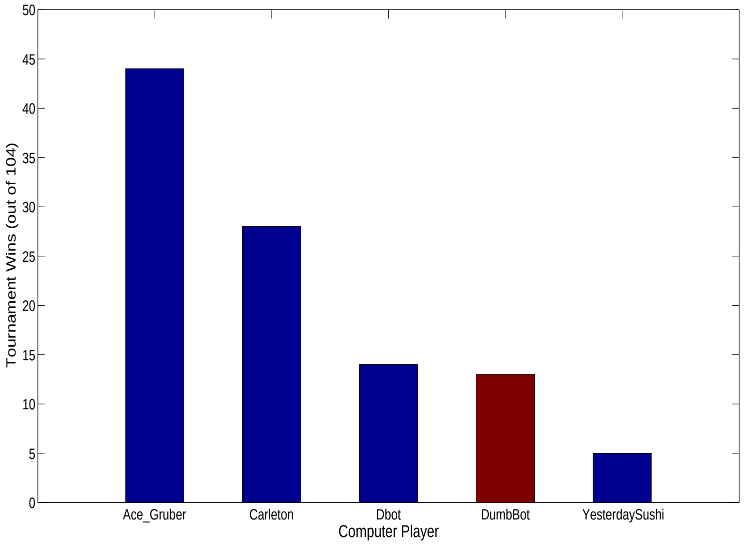


Figure 3.1: ICCM 2004 PokerBot Event #1.

any feasible hand. This allowed the program to learn its opponents' playing styles, and in this regard was unique amongst the entrants.

One significant difference between *Ace_Gruber* and its GAMES Group predecessors was its ability to play no-limit poker, since *Loki* and *Poki* were only designed for limit betting. Possible bet sizes were grouped into "small", "big", and "giant", and treated as distinct actions.

3.4.5 Extra Tournaments

In addition to the main competition, the organizers ran a supplementary event which included an extra player of their own design. This program, named *Mr-AI*, simply pushed all in every time it was its turn to act. The purpose behind including this program was to see how well such a simple strategy would perform, and also to notice whether the other players were able to adapt against it. The results of this extra event are shown in Figure 3.2.

The results from these 68 extra tournaments reinforce *Ace_Gruber*'s dominance over its competitors. Sokorac's program won 34% of the tournaments, again well ahead of the second placed program.

The results for *DumbBot* show a marked difference between the first and second events. Placing only fourth out of five for the first competition, the program improved to second place out of six in the extra event. One plausible reason for this is that

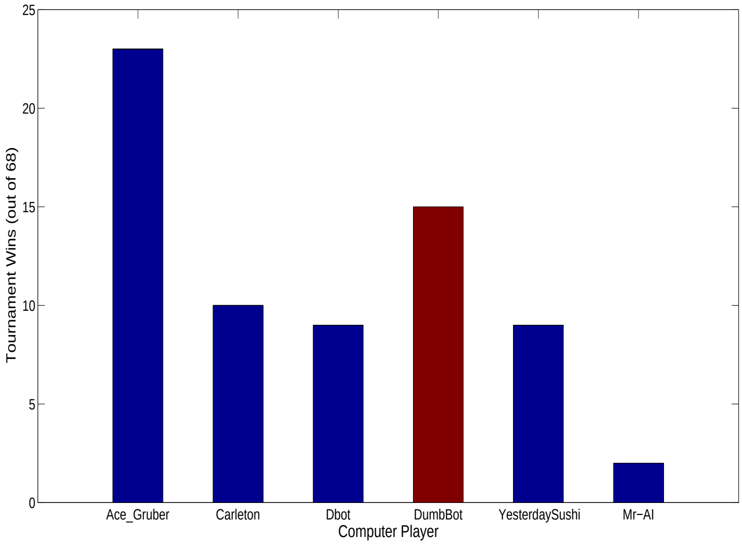


Figure 3.2: ICCM 2004 PokerBot Event #2.

the increased number of opponents suited *DumbBot*'s strategy better. As we have noted, Sklansky's *System* was only meant for full-table play and he expected that it would be inappropriate for fewer than nine opponents. The additional player made the tournament slightly more appropriate for the *System*'s intended setting.

Whilst it is unreasonable to expect that a simple pre-flop all in or fold strategy could outplay a sophisticated program such as *Ace_Gruber*, the results of this competition do support Sklansky's assertion that such an approach is one of the most effective simple ways of participating in a no-limit Texas hold'em tournament.

3.4.6 Comments

Using the Z-test for a proportion (binomial distribution) [123], it is possible to determine the number of tournament wins that are required to reject the null hypothesis that a program scored a naive average proportion of victories: one fifth in the first event, and one sixth in the second. The test statistic is given by

$$Z = \frac{(p - p_0)}{\left\{\frac{p_0(1-p_0)}{n}\right\}^{\frac{1}{2}}}$$

where p is the observed proportion, p_0 the assumed proportion, and n the number of observations.

Given n and p_0 we can find what values of p are required such that $Z \geq 1.65$, the 5% one-tailed significance level. Multiplying these proportions up by the number of tournaments in each event yields the values 28.0 and 16.9. In scoring 44 and 23, *Ace_Gruber* is statistically far better than average.

The results of these events bolster the argument that opponent modelling is the single most important facet of a strong computer poker player. It would have been interesting to note the difference in *Ace_Gruber*'s performance with its opponent modelling turned off.

It is not accurately known to what extent each program used the game factors previously discussed in their decision making procedures. At least two - *Dbot* and *DumbBot* - used some form of ratio of bet size to blind sizes in selecting their actions.

The second competition results are very encouraging in showing how many tournaments it is possible to win by using an all in or fold strategy. Although the strength of the competitors cannot be objectively measured, it appears that such a strategy can perform adequately against more realistic opposition.

3.5 Kill Phil Strategies

One of the most recently published books on tournament poker strategy is that co-authored by Blair Rodman and Lee Nelson [101]. This work builds upon Sklansky's *System* to make a collection of four no-limit Texas hold'em tournaments strategies. The different approaches are entitled "Kill Phil Rookie", "Kill Phil Beginner", "Kill Phil Basic Plus", and "Kill Phil Expert", and are ordered in increasing levels of sophistication. All share the common premise of being all in or fold strategies. The Kill Phil Rookie strategy is given in Table 3.7.

The authors argue that there are essentially two methods that can be utilized in playing a no-limit poker tournament. They call these "small ball" and "long ball". The first of these strategies is identified as being the most complex and difficult to master. It is symbolized by playing many hands and trying to win small pots through outplaying one's opponents in all betting rounds of a Texas hold'em event. It is the strategy that the top professional poker players employ, and it allows them to amass many chips in the early stages of a tournament without putting a significant proportion of their stack at risk.

Long ball poker is the opposite technique, being far simpler to realize and requiring much less subtlety. A player using the long ball strategy will play very few hands, but

Tournament Stage	No Prior Bet	Prior Bet
Early	AA, KK	AA, KK
Middle	Pairs, AK, any ace-suited, suited connectors 54s or higher	AA, KK, AKs
Final 4-6 players	Pairs, A7 or higher[a], KT or higher, any ace-suited, suited connectors 54s or one-gaps 64s or higher	Pairs 66 or higher; AT or higher
Final 3 players	Pairs, any ace or king, suited connectors 54s or one-gaps 64s or two-gaps 63s or higher	Pairs; A7 or higher, KJ or higher; any ace-suited
Final 2 players	Move in every hand	Move in every hand

[a] Note that the term "A7 or higher" means a hand with an ace and *at least* a 7.

Table 3.7: Kill Phil Rookie Strategy.

when he does he will often commit all of his chips to his decision before any of the board cards have been dealt. This approach is far more hazardous because it constantly puts the player at risk of elimination. However, successful use of the long ball game can be an effective way to maintain chips by stealing blinds, and also scores the occasional "double-up" when one is called and wins the pot.

Like Sklansky, Rodman and Nelson argue that the all in or fold method is an effective approach for a tournament poker player wishing to reduce the disparity between his abilities and those of more experienced opponents. They believe that their strategies, especially the more sophisticated ones, are sufficiently strong to give their users a good chance of positive returns from real Texas hold'em tournaments.

3.6 Summary

In this chapter we introduced tournament poker, and highlighted several of the ways in which it differs from its ring game equivalent. We discussed the ramifications of the tournament structure, and examined some of the strategic considerations that are deemed crucial for strong decision making.

The factors previously discussed for ring game play - hand strength, seating position, and opponents' prior actions - are still important in tournament play. In addition to these, knowledge of the tournament level and chip stack sizes should also be accounted for.

In seeking to understand tournament poker play, we have studied Sklansky's *Systems*, and shown through application in the ICCM 2004 PokerBot Tournament that all in or fold strategies do have some merit. We have seen that a strong tournament poker player requires opponent modelling to achieve top results, but that all in or fold systems which incorporate some of the factors mentioned above can perform adequately against more elaborate opposition.

Given the relative success of the above methodology, and the simplifications it makes to encoding a strategy, we continue our research by examining the importance of the aforementioned strategic factors in an all in or fold Texas hold'em tournament domain.

Chapter 4

Initial Tournament Poker Investigations using Exhaustive Simulations

We have seen how an all in or fold strategy can be used with reasonable success in no-limit Texas hold'em tournaments. In this chapter we further investigate this approach by performing simulations of different strategies against three static opponents from the non-academic literature, and gauging the impact of knowledge of certain game factors. Specifically we seek to determine whether the inclusion of knowledge of seating position, opponents' prior actions, tournament level, and stack size leads to improved tournament performance over strategies based on hand strength alone.

4.1 Exhaustive Simulations Experimental Framework

4.1.1 Tournament Structure

In our experimental work we wish to discover whether the strategic considerations noted by poker authors in Sections 2.7 and 3.2 can be demonstrated empirically. To do this on a full scale no-limit Texas hold'em tournament would be very difficult given the complexity of the game. Therefore we seek to make simplifications to full scale poker, but still retain a sufficient amount of realism.

One of the first restrictions we can impose to reduce the strategy space to a tractable size is to limit the players' betting actions to all in or fold. In real poker a player choosing to bet may decide the amount he wishes to stake. Replacing this possibility with the binary choice of all or nothing shifts the focus to the more general and important strategic question of when to bet, rather than how much. We commented in the previous chapter that pre-flop strategies have been proposed for use in real poker tournaments. We also noticed that such a system can score a significant number of tournament successes when we entered one into a computer poker competition.

The use of the all in or fold restriction for the competitors removes the necessity for the final three betting rounds in Texas hold'em. This is because all active players will have committed the whole of their stack to the pot in the pre-flop betting. Since the poker professionals' recommendations relate to the full game, they implicitly apply to the first, pre-flop round of betting. Concentrating on the pre-flop form of Texas hold'em should not therefore affect the applicability of the authors' advice.

All of the subsequent experiments employ all in or fold Texas hold'em between ten players with a winner-takes-all prize format. Whilst it is more usual for ten-player tournaments to employ a percentage payout structure, the *satellite* format is also sometimes employed. This design only credits a player for finishing first, and hence second place is equivalent to finishing last. Formulating the competitions in this way ensures that we assess each strategy's ability to win tournaments, and not just their capacity for tournament survival[1]. Also, by solely concentrating on whether a given strategy wins a tournament or not we are able to cease a simulation once our test player is eliminated. This condition saves a large amount of computation time.

In our experiments all players commence with $1,000 in tournament chips. The blind structure is based on the number of hands played. After every ten hands the blinds rise through a possible eleven levels according to the schedule given in Table 4.1. If reached, the blinds stay at the final level until the tournament's cessation. The structure chosen is very similar to those used in ten-player poker tournaments found online.

The deals in each tournament are seeded so that each experiment uses the same card ordering. However, the nature of tournament poker precludes precise comparison of two players in separate tournaments. As chips move between players and competitors are eliminated, the situations faced by two test players in simultaneous tournaments will start to differ. For example, once a player is eliminated from a tournament the first card that he would have been dealt had he survived automatically passes to the next remaining player with a predetermined deck. This offset then applies to all subsequent cards. Seeding the deals only keeps comparable tournaments similar until such an

[1] With a percentage payout format profitable strategies will exist which win few tournaments but score a sufficiently high number of 2^{nd} and 3^{rd} place finishes. The satellite format allows us to sharpen our definition of what constitutes a successful strategy.

Level	Small Blind	Big Blind
1	$5	$10
2	$10	$20
3	$25	$50
4	$50	$100
5	$100	$200
6	$200	$400
7	$300	$600
8	$500	$1,000
9	$1,000	$2,000
10	$2,000	$4,000
11	$5,000	$10,000

Table 4.1: Blind sizes at each tournament level.

inevitable divergence.

The computer code used in performing these experiments incorporated much of the GAMES Group's publicly available Texas hold'em poker resources [92]. Their framework was designed for limit Texas hold'em in a ring game setting, and so several modifications and new classes had to be written to allow for no-limit poker in a tournament format. The random number generation used throughout our experiments for the card shuffle is an implementation of the Mersenne Twister [6], which ensures unbiased sampling.

4.1.2 Hand Groupings

Appendix A details how the 1,326 possible two-card starting hands in Texas hold'em can be reduced to 169 using suit equivalence. Due to the length of time required to run a sufficient number of poker tournaments on all of these hands the list was further condensed. This reduction is non-trivial since starting hands in Texas hold'em do not readily conform to a total ordering, and are best conceptualized as a partially ordered set.

To illustrate this point, take the three starting hands:

$$2\heartsuit 2\clubsuit,\ J\spadesuit T\spadesuit,\ \text{and}\ A\clubsuit K\diamondsuit.$$

Calculating the pairwise expected win percentages of these hands yields:

Pairing	Winner	Loser
$2\heartsuit 2\clubsuit$ vs. $J\spadesuit T\spadesuit$	$J\spadesuit T\spadesuit$ (54.0%)	$2\heartsuit 2\clubsuit$ (46.0%)
$2\heartsuit 2\clubsuit$ vs. $A\clubsuit K\diamondsuit$	$2\heartsuit 2\clubsuit$ (52.6%)	$A\clubsuit K\diamondsuit$ (47.4%)
$J\spadesuit T\spadesuit$ vs. $A\clubsuit K\diamondsuit$	$A\clubsuit K\diamondsuit$ (58.8%)	$J\spadesuit T\spadesuit$ (41.2%)

Table 4.2: Pairwise expected win percentages of three Texas hold'em starting hands.

Hence we see that the hands possess an intransitive relation.

There is no doubt amongst poker players that some starting hands are superior to others. By far the most desired starting hand is a pair of aces, and similar high pairs are also strongly coveted. Several attempts have been made to rank the starting hands by strength value, although all such listings are inherently flawed due to intransitivities as noted above. One very recent book [112] contains a plausible ranking of all 169 possible starting hands. For the complete ranking of the hands, and an explanation of how this list was prepared, refer to Appendix B.

Given this total ordering we split the hands into thirteen groups, each containing thirteen elements. Group one contains the strongest hands, group two the next strongest, and so forth down to the weakest in group thirteen. These groupings are shown in Table 4.3.

Note that there are some general comments that we can make about the strength of hands within these rankings. Firstly, high cards and pairs are more likely to be found near the top of the listings. These hands have a high chance of winning in a showdown without any improvement from the board cards. Secondly, two suited cards are always stronger than the respective offsuit hand due to the increased chances of making a flush. Finally we see that zero-gap hands are generally more highly rated than similar one- and two-gap hands due to their straight potential.

It should also be noted that although these groupings each contain thirteen elements they are not all of equal size with regard to unique starting hands. This is due to the different frequencies of pairs, suited, and offsuit hands, as explained in Appendix A.

Although the classification employed is coarse, it helps to provide a distinction between the strengths of starting hands and reduce the strategy space to a more tractable size.

Group	Starting Hands	#
1	AA, KK, AKs, QQ, AKo, JJ, AQs, TT, AQo, 99, AJs, 88, ATs	82
2	AJo, 77, 66, ATo, A9s, 55, A8s, KQs, 44, A9o, A7s, KJs, A5s	84
3	A8o, A6s, A4s, 33, KTs, A7o, A3s, KQo, A2s, A5o, A6o, A4o, KJo	110
4	QJs, A3o, 22, K9s, A2o, KTo, QTs, K8s, K7s, JTs, K9o, K6s, QJo	94
5	Q9s, K5s, K8o, K4s, QTo, K7o, K3s, K2s, Q8s, K6o, J9s, K5o, Q9o	100
6	JTo, K4o, Q7s, T9s, Q6s, K3o, J8s, Q5s, K2o, Q8o, Q4s, J9o, Q3s	100
7	T8s, J7s, Q7o, Q2s, Q6o, 98s, Q5o, J8o, T9o, J6s, T7s, J5s, Q4o	100
8	J4s, J7o, Q3o, 97s, T8o, J3s, T6s, Q2o, J2s, 87s, J6o, 98o, T7o	108
9	96s, J5o, T5s, T4s, 86s, J4o, T6o, 97o, T3s, 76s, 95s, J3o, T2s	92
10	87o, 85s, 96o, T5o, J2o, 75s, 94s, T4o, 65s, 86o, 93s, 84s, 95o	108
11	T3o, 76o, 92s, 74s, 54s, T2o, 85o, 64s, 83s, 94o, 75o, 82s, 73s	100
12	93o, 65o, 53s, 63s, 84o, 92o, 43s, 74o, 72s, 54o, 64o, 52s, 62s	108
13	83o, 42s, 82o, 73o, 53o, 63o, 32s, 43o, 72o, 52o, 62o, 42o, 32o	140

Table 4.3: The 13 groups of the 169 possible starting hands used within the experiments. The suffices “s” and “o” denote suited and offsuit hands respectively. The final column shows the number of unique starting hands contained within each group.

4.1.3 Decision Making

All of the experiments performed in this chapter associate either an all in or fold betting action to a test player’s starting hand depending on a number which governs the playable groups. This threshold value has a minimum of zero and a maximum of thirteen. So for example, a strategy which has a threshold of two will move all in with any hand from the first two groups shown above and fold all hands from the lower eleven. Hence a player using a threshold of zero will fold all hands, and one with a threshold of thirteen moves all in every time.

For the first set of experiments utilizing hand strength alone there are fourteen possible strategies. We can denote these by the threshold value (x), where $x \in [0, 1, \ldots, 13]$.

The simulations which follow on from these use hand strength in conjunction with one of the decision making factors discussed in the previous two chapters. The four factors we examine in this manner are seating position, opponents’ prior actions, tournament level, and chip stack amount.

For parsimony the experiments treat each of these factors as a binary variable as follows:

- Seating position: The test player is in an early/late position with respect to the dealer.
- Opponents' prior actions: There are no bets/at least one bet prior to the test player's action
- Tournament level: The decision is made early/late in a tournament
- Chip stack amount: The test player has few/many chips

The precise definitions of these distinctions are made clear in the following sections.

In accord with our previous notation, a strategy within this expanded framework can be represented by the tuple (x,y), with $x,y \in [0,1,\ldots,13]$. The x-value denotes the threshold hand group when the binary variable is in its first state, and the y-value gives the minimum playable hand group when the binary variable is in its second state. Therefore in each of these experiments our search space is squared to yield a total of 196 possible strategies.

In all of the experiments our test player competes against nine identical opponents. The player then plays numerous tournaments against these opponents, and the number of wins totalled. We perform multiple tournaments for the same strategy to reduce the effects of luck. The more tournaments played, the more we can be sure of the effectiveness of a given strategy.

In each suite of experiments we sequentially cycle our test player's strategy through all allowable threshold values: there are fourteen such possibilities based on hand strength alone, and 196 with the inclusion of a binary variable. We use the term exhaustive simulation to convey the fact that all permissible strategies within our framework are enumerated over.

4.1.4 Tournament Opponents

There are three different opponents used in all the simulations. These encode the original and improved Sklansky *Systems*, and the Kill Phil Rookie (KPR) strategy. We refer to the former as Sklansky Basic (SB) and Sklansky Improved (SI) strategies. All three are termed "static opponents" to reinforce the fact that these strategies are rule-based and unchanging.

The first two strategies are taken from the book “Tournament Poker for Advanced Players” [110], and were previously discussed in Section 3.3. To recap, the Sklansky Basic strategy is highly restrictive, and will only bet the very best starting hands once another player has already opened the pot. If no other player has yet bet, this strategy will move all in with a slightly larger subset of hands.

The Sklansky Improved strategy is similarly restrained in the hands it will play if an opponent has entered the pot. Where this strategy differs from the first is in its use of a key number to determine playable hands when no other player has yet made a bet. The key number is calculated primarily as the ratio of the player’s stack to the total amount of the blinds. Each starting hand is given a threshold value, and a comparison between this value and the key value determines whether the player should move all in or fold.

The third opponent employed is based on one from “Kill Phil” [101]. This book contains several strategies of increasing complexity, and our experiments use the simplest “Rookie” strategy , met previously in Section 3.5. Similar to the two Sklansky strategies, the Kill Phil Rookie strategy contains instructions on which hands are playable depending upon whether or not an opponent has yet bet into the pot. The major difference in this strategy is that the classification of playable hands is determined by the number of players remaining and the tournament level.

For our experimental purposes the Kill Phil Rookie tournament stages “early” and “middle” shown in Table 3.7 were interpreted as the first and the following two levels respectively. Where the table cites the “Final 4-6 players”, we extended this for any number of players greater than or equal to four. This is because it is possible to have more than six players remaining after the first three levels.

4.2 Exhaustive Simulation Results

4.2.1 Hand Strength Only

The first suite of simulations seek to determine how well a player can fare when their betting action is based solely upon knowledge of their own cards. These results are important to form a baseline for comparison with the future experiments, in which extra information is combined with hand strength in the decision making process.

Each simulation contains a test player with a strategy consisting of a simple threshold representing which groups of hands are playable. If the hand they are dealt is contained within a group numbered less than or equal to that specific threshold value, the player will move all in. Otherwise, they will fold.

Within this framework there are fourteen possible strategies, and each was tested over 1,000 tournaments against each of the three opponents. Every individual competition contained nine such opponents, and a record was made of how many tournaments the test player won.

The results from this experiment are shown in Figure 4.1.

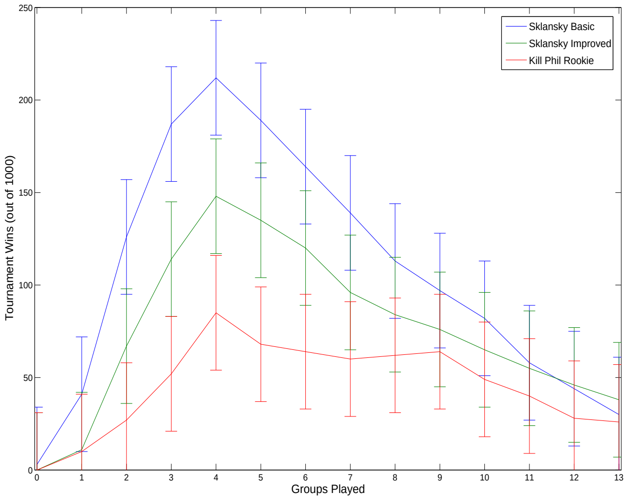


Figure 4.1: Tournament wins with corresponding 95% confidence intervals against the three static opponents using hand strength knowledge only.

The first point to note about this graph is that there appears to be a clear ranking in the relative strength of the opponents. The most wins were achieved against the Sklansky Basic strategy, followed by the Sklansky Improved and then the Kill Phil Rookie ones. This result shows that the win rates of the three opposing strategies mirror their own relative complexities.

Trends are clear in the results against all the opponents. When no hand groups are played the test players unsurprisingly fare poorly. Against the Sklansky Improved and Kill Phil Rookie strategies the test players never win a tournament. Fortuitously, the test player does win three tournaments against the Sklansky Basic strategy by never betting a hand. This outcome can be explained by the structure of tournament play. A player who is all in on posting his blind is still able to compete in the showdown, and hence retains the chance to win the pot. Although highly unlikely, we see here that it is possible for a player to win a tournament solely through winning sufficient hands in this manner.

The best results for each player occur when they play with a threshold of four. These strategies relate to moving all in with approximately the top 30% of starting hands. Playing more or fewer hands than this causes a decline in the test players' scores. The explanation for this result is straightforward. Betting with too few hands means that the test player will frequently be anted-away by the increasing blind structure employed. At the other end of the spectrum, a player who bets too many hands is employing a high risk strategy, and will eventually face an opponent with a playable hand who knocks them out.

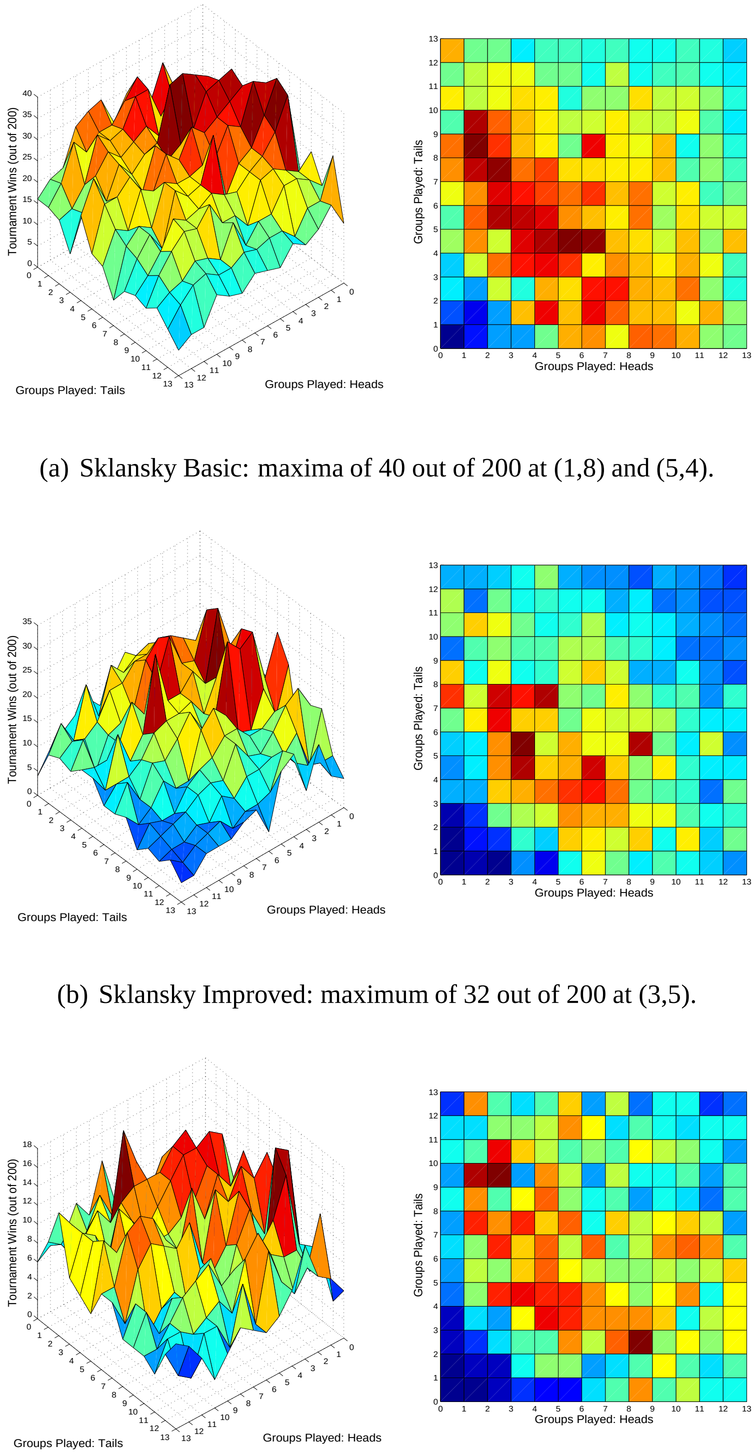
It should be noted that the trend within the Kill Phil Rookie graph is not precisely monotonically decreasing to the right of the optimum value. The results for the points representing thresholds of eight and nine against this strategy would appear to be anomalous. The reason for this is almost certainly due to the inherent noise contained within the experiment. It is expected that a re-sampling rate larger than the figure of 1,000 tournaments used here would correct this irregularity, although this was not attempted.

4.2.2 Hand Strength and Coin Toss

Following on from the previous experiments we now include extra information in addition to hand strength for the test players to base their bet or fold decisions on. These extra criteria take the form of a binary variable, and allow the player to make a different decision depending on the variable's state. Unlike the previous experiments where there were a total of only fourteen possible strategies, incorporating the state of the binary variable squares the strategy space to 196 possibilities. Due to the increased computational burden of simulating over this higher number, the re-sampling rate for this and all subsequent experiments in this chapter was chosen to be 200 tournaments. This figure was selected to keep the run-times down, whilst being large enough to elicit trends in the results. On a Pentium IV 2.60GHz machine the evaluation of, for example, a single Sklansky Basic counter-strategy over 200 tournaments takes an average of approximately four minutes[2].

Before we start to include game knowledge that we believe may be useful in a player's decision making, we first experiment with a piece of dummy information for use as a control experiment. In these first simulations we imagine that the test player

[2] The length of a single tournament varies since a tournament ends once the test player is eliminated, meaning different numbers of hands are required in different cases.



(a) Sklansky Basic: maxima of 40 out of 200 at (1,8) and (5,4).

(b) Sklansky Improved: maximum of 32 out of 200 at (3,5).

(c) Kill Phil Rookie: maxima of 18 out of 200 at (2,9) and (8,2).

Figure 4.2: Tournament wins against the three static opponents using hand strength and the outcome of a coin flip.

flips a fair coin before choosing his action. If the coin lands heads, he compares his hand to the x-value in his strategy (x,y). Similarly, if the coin lands tails he uses the y-value as his betting criterion. Clearly the result of the coin toss is independent of any information contained within the game. The results from this experiment allow us to gauge the impact of extra knowledge when we replace the binary variable from the outcome of a coin toss to other game-related information.

Figure 4.2 plots the number of tournaments won using hand strength and the outcome of the coin flip to make betting decisions against each of the three opponents. This figure contains three pairs of graphs, one pair for each of the three aforementioned opponents. The left hand graph of each pair shows a surface plot of the number of tournament wins for each of the 196 allowable strategies. The complementary graph gives a two-dimensional depiction of the same data.

Since the outcome of a coin toss is independent of any game-related information, we would expect the scores along the $y = x$ diagonal to show a similar profile to the hand strength only experiments. This is because a strategy (x,x) along the diagonal uses the threshold value x, regardless of the coin flip.

Previously we observed that the total number of tournament wins was reduced by using a threshold value other than four in the hand strength only experiments. Therefore in the limit as the number of tournaments rises, we would expect to see peaks in all graphs at $(4,4)$ and lower values elsewhere. All three pairs of plots do show high scores around this value, but the globally highest peaks are typically elsewhere. This is due to the increased amount of noise resulting from re-sampling over just 200 tournaments in these experiments.

One way of testing whether the extra information conveyed in the binary variable has moved the maxima off-diagonal is by assessing the symmetry of the resulting plots. If we consider a general strategy (x,y) in these experiments, we note that with an unbiased coin we would expect to select each threshold 50% of the time. This is exactly the same for the corresponding strategy (y,x). Therefore we should expect to observe a degree of symmetry in the diagonal $y = x$, subject to noise.

Symmetry in the results can be tested for using the statistical method of paired comparisons. In this test the null hypothesis states that the mean difference, μ_d, between paired observations is equal to zero. The test statistic is given by

$$t = \frac{\overline{d}-\mu_d}{s/\sqrt{n}}$$

where $d_i = z_{xy} - z_{yx}$ is the difference in the n paired observations, with z_{xy} the score achieved by strategy (x,y). The value of s is the standard deviation of the d_i.

The method for paired comparisons applied to the scores resulting from hand strength and coin toss knowledge yields the p-values shown in Table 4.4.

	SB	**SI**	**KPR**
P-value	0.7459	0.4653	0.8615

Table 4.4: P-values for the method of paired differences test applied to the hand strength and coin toss knowledge results against the three static opponents.

Since these values are so large we fail to reject the null hypothesis that the mean difference between paired observations is equal to zero. From this we conclude that the inclusion of information relating to the toss of a coin has not moved the location of the peaks off the diagonals with statistical significance. We shall return to this result when considering the inclusion of game-related knowledge in the following sections.

4.2.3 Hand Strength and Seating Position

Now that we have seen the impact of incorporating a dummy binary variable into the test player's strategy we move on to replacing it with information that we believe may positively affect tournament poker decision making. The first such piece of information we include is seating position. The term seating position is always used in the context of a player's location relative to the dealer. In these experiments the binary variable takes the values “early” or “late”, depending upon whether the test player is one of the first or last to act on a particular hand. Recall that since the deal passes with every new hand the player's relative position continually rotates.

The classification of the test player's position depends upon the number of players competing in a hand. With an even number of competitors exactly half of the players are termed both early and late position in our experiments. If an odd number of players remains we place the surplus player in the late classification.

Table 4.5 shows the definitions used for all possible numbers of players during our single table tournaments.

Note that for three or more players the competitors posting the small and big blind are classified as late position. In a regular Texas hold'em game these seats are more usually defined as being in early position, since in three of the four betting rounds the small blind player is the first to act. In our all in or fold framework, however, we

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
10 players	L	L	L	E	E	E	E	E	L	L
9 players	L	L	L	E	E	E	E	L	L	
8 players	L	L	L	E	E	E	E	L		
7 players	L	L	L	E	E	E	L			
6 players	L	L	L	E	E	E				
5 players	L	L	L	E	E					
4 players	E	L	L	E						
3 players	E	L	L							
2 players	L	E								

Table 4.5: Classification of seating position as early (E) or late (L) position with respect to the dealer by the number of players. SO represents the dealer seat, S1 the small blind, S2 the big blind, and so on sequentially around the table.

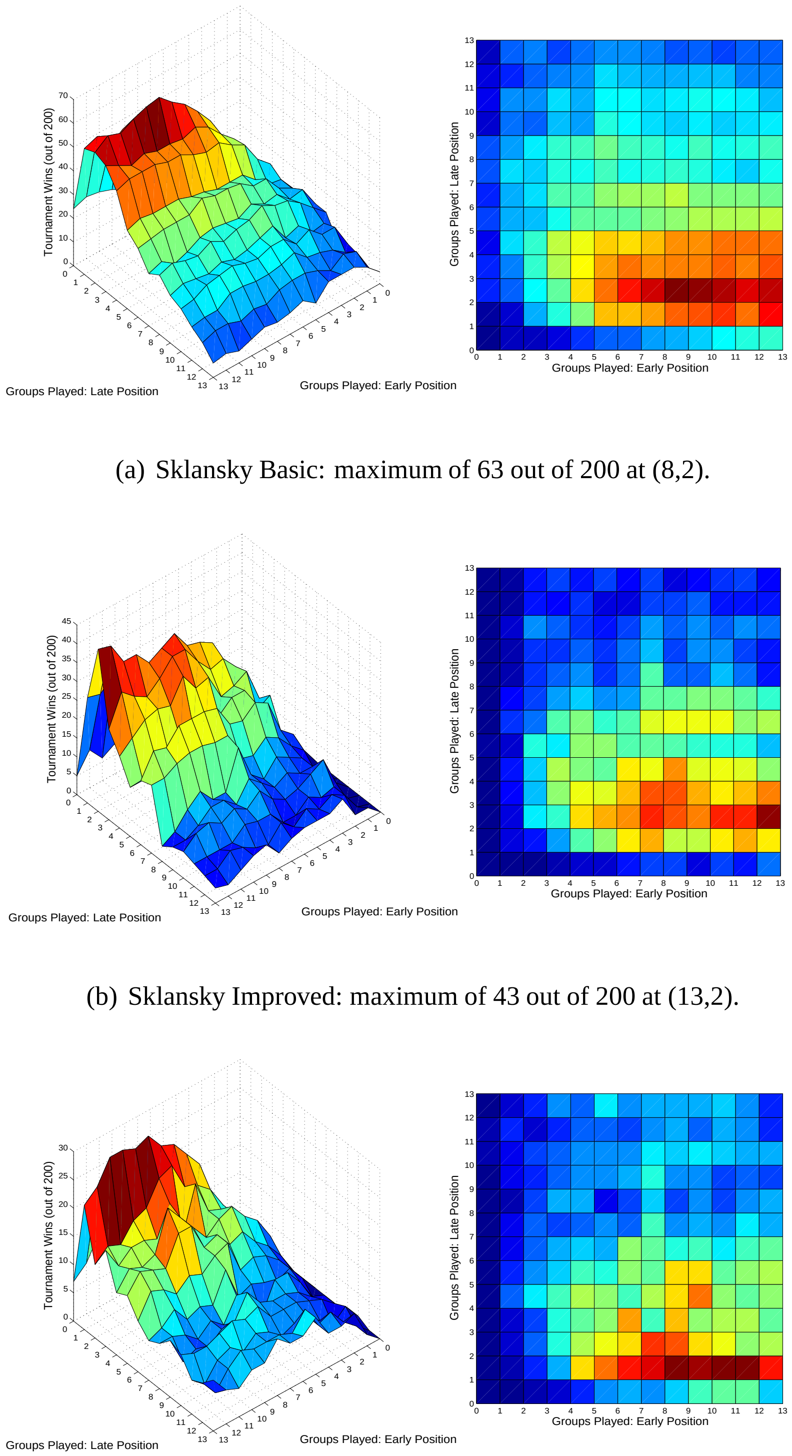
are simulating Texas hold'em with only the pre-flop betting round. Having posted the blinds, the players in the two seats to the left of the dealer are the last to act on this round. This explains their late position classification.

A further point to note is that when a tournament is reduced to two players the dealer becomes the big blind. This is a standard rule in Texas hold'em. In such a situation the small blind acts first, followed by the big blind/dealer. Therefore we classify the small blind to be in early position, and the big blind/dealer to be in late position.

We observed in Section 2.7.2 that seating position is often mentioned as being important in the non-academic literature. There is a supposed benefit to being seated in late position, since in this case a player gets to observe the actions of many more of his opponents before it is his turn to act. Similarly, since there are fewer players seated behind him, a late position player is boosted by the reduced potential of strong hands coming into the pot after him. A player in early position has the least amount of knowledge of his opponents' likely holdings, and faces a greater risk of coming up against a premium hand.

Graphs of the number of tournament wins resulting from the inclusion of seating position knowledge against the three static opponents are shown in Figure 4.3.

Examining the plots shows that the best results come from playing very differently



(a) Sklansky Basic: maximum of 63 out of 200 at (8,2).

(b) Sklansky Improved: maximum of 43 out of 200 at (13,2).

(c) Kill Phil Rookie: maxima of 28 out of 200 at (8,1), (10,1) and (11,1).

Figure 4.3: Tournament wins against the three static opponents using hand strength and seating position knowledge.

in either early or late position. If the test player is one of the first to act the results suggest that many hands should be played, with between five and all thirteen groups yielding the best scores. Conversely, only the top one or two groups should be played when sitting in late position.

These results appear to contradict those given in the non-academic poker literature. There we find repeated recommendations that a player should be more conservative in early position, and be apt to play more hands in the later seats. There are three fundamental reasons why our results are at odds to the guidance of poker professionals. The first of these is a consequence of the simplification to the betting that we have employed, the second is due to the nature of the opponents, and the third is an effect of limited information.

Late position players retain an informational advantage throughout all four betting rounds in a hand of Texas hold'em. In the case of the all in or fold variant employed in these simulations, however, there is only ever one round of betting. Hence the restricted nature of the betting in this game greatly reduces the positional edge.

All three of the opposing strategies play a different range of hands depending upon whether or not one of their opponents has bet into the pot. Specifically, all three strategies are extremely tight if the pot has already been opened. Our results show that the test players in early position benefit from this. By betting before anyone else acts, they are able to "scare" their opponents into folding. This strategy allows the test players to pick up the blinds uncontested. The "first-in vigorish" [60] is often enough to pick up a sufficient number of pots.

The third reason for the discrepancy between our results and the professionals' guidance is because the test players do not have the benefit of observing their opponents' prior actions. The non-academic literature's preference for late position is primarily based upon the informational advantage gained through seeing one's opponents act. Players in a late seating position with no prior action are commonly advised to attempt to steal the blinds with more marginal holdings. Our test players do not know whether or not an opponent has already opened the pot. It is safer, therefore, to bet fewer hands in late position than early.

It is clear by observation that all plots are highly asymmetrical about the line $y = x$, and this gives the first indication that the scores achieved are dependent upon the extra piece of information. We would expect that if the extra information was worthless there would be a symmetry about the diagonal, as we observed with the previous results incorporating coin toss knowledge.

Using the method of paired comparisons test explained previously the results in Table 4.6 show that there is a statistically significant asymmetry in the results.

	SB	**SI**	**KPR**
P-value	< 0.0001	< 0.0001	< 0.0001

Table 4.6: P-values for the method of paired differences test applied to the hand strength and seating position knowledge results.

In all three cases we can reject the null hypothesis that the plots are symmetrical about $y = x$, and conclude that the inclusion of seating position knowledge has impacted the location of the highest scoring strategies.

4.2.4 Hand Strength and Opponents' Prior Actions

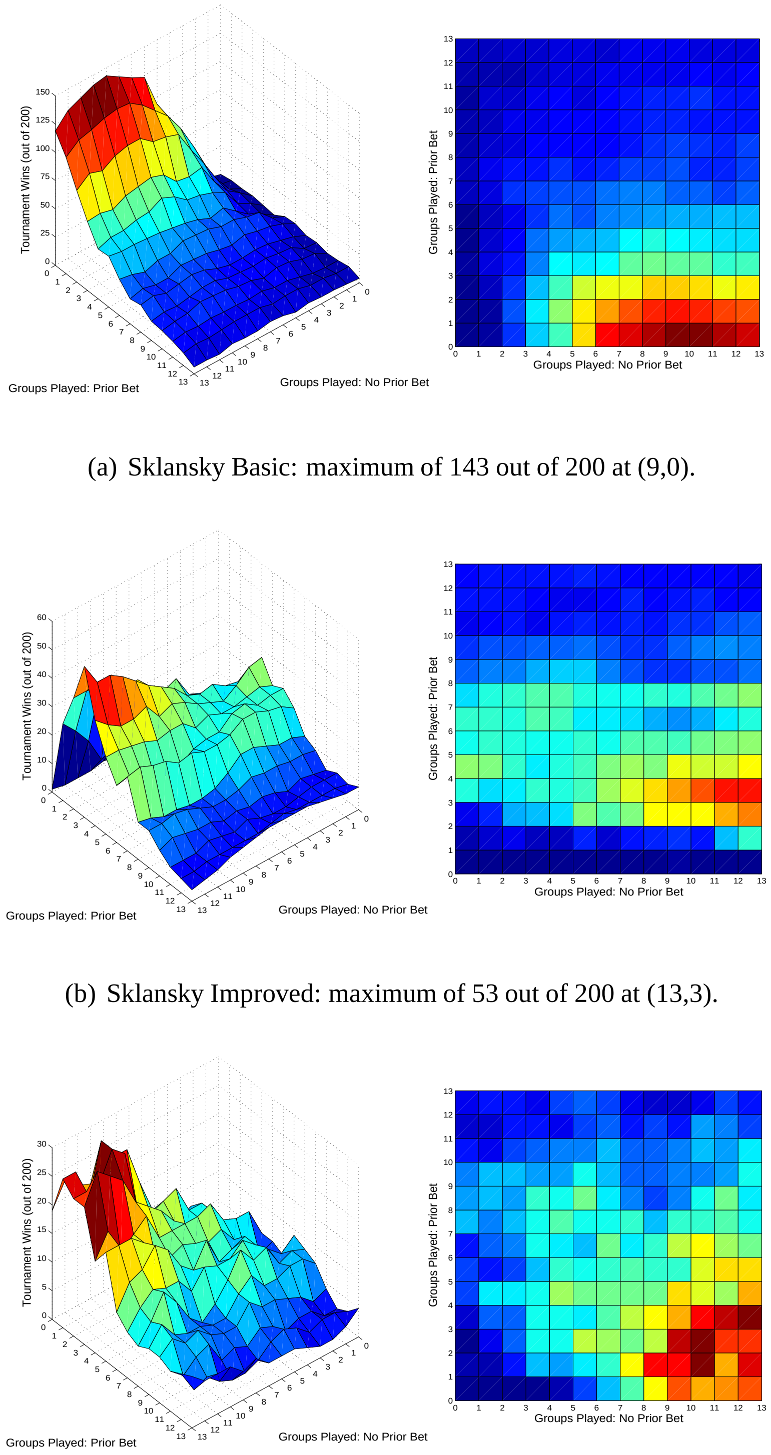
All in or fold hold'em simplifies the classification of opponents' prior actions. In the pre-flop betting of a real game there is the possibility of a player calling the big blind. Here this option is unavailable. The binary variable we next incorporate into our test player's decision making is whether at least one opposing player has already bet into the pot or not.

Figure 4.4 plots the number of tournaments won through the inclusion of prior action knowledge against each of the three static opponents.

The shape of all three pairs of plots suggest very clearly that a greater number of tournament wins results from playing only the best hand groups when an opponent has already bet, and playing almost all hand groups in the absence of a prior bet. Whilst there is undoubtedly noise in the results, the plots clearly show evidence that playing many hands after an opposing bet, or failing to bet hands into an unopened pot, leads to a lower overall tournament win rate.

Evidence from these experiments is in accordance with recommended tournament poker strategy, as discussed in Section 2.7.3. Experts point to the necessity for a player to be more restrictive in the hands they play once someone else has bet into the pot. Indeed, it is worth noting that the three static opponents themselves incorporate this facet into their own strategies.

What is surprising about these results are the very large numbers of hand groups played to achieve the best results in the cases of no prior bet. The findings suggest that



(a) Sklansky Basic: maximum of 143 out of 200 at (9,0).

(b) Sklansky Improved: maximum of 53 out of 200 at (13,3).

(c) Kill Phil Rookie: maxima of 29 out of 200 at (10,1), (10,2), and (12,3).

Figure 4.4: Tournament wins against the three static opponents using hand strength and opponents' prior bet knowledge.

each of the three opponents' strategies are possibly too restrictive, and that simply betting many hands and causing them to fold is the route to a large number of tournament successes against them.

Notice again that these plots are unlike those seen from the inclusion of coin toss knowledge. The asymmetry about the main diagonal is again evident, as it was with the knowledge of seating position. Statistical significance in the impact of opponents' prior bet information is confirmed in the results of using the method of paired comparisons test, shown in Table 4.7.

	SB	**SI**	**KPR**
P-value	< 0.0001	0.0001	< 0.0001

Table 4.7: P-values for the method of paired differences test applied to the hand strength and opponents' prior bet knowledge results.

Again in all three cases we reject the null hypothesis that the plots are symmetrical. Knowledge of opponents' prior actions has moved the location of the highest scoring strategy.

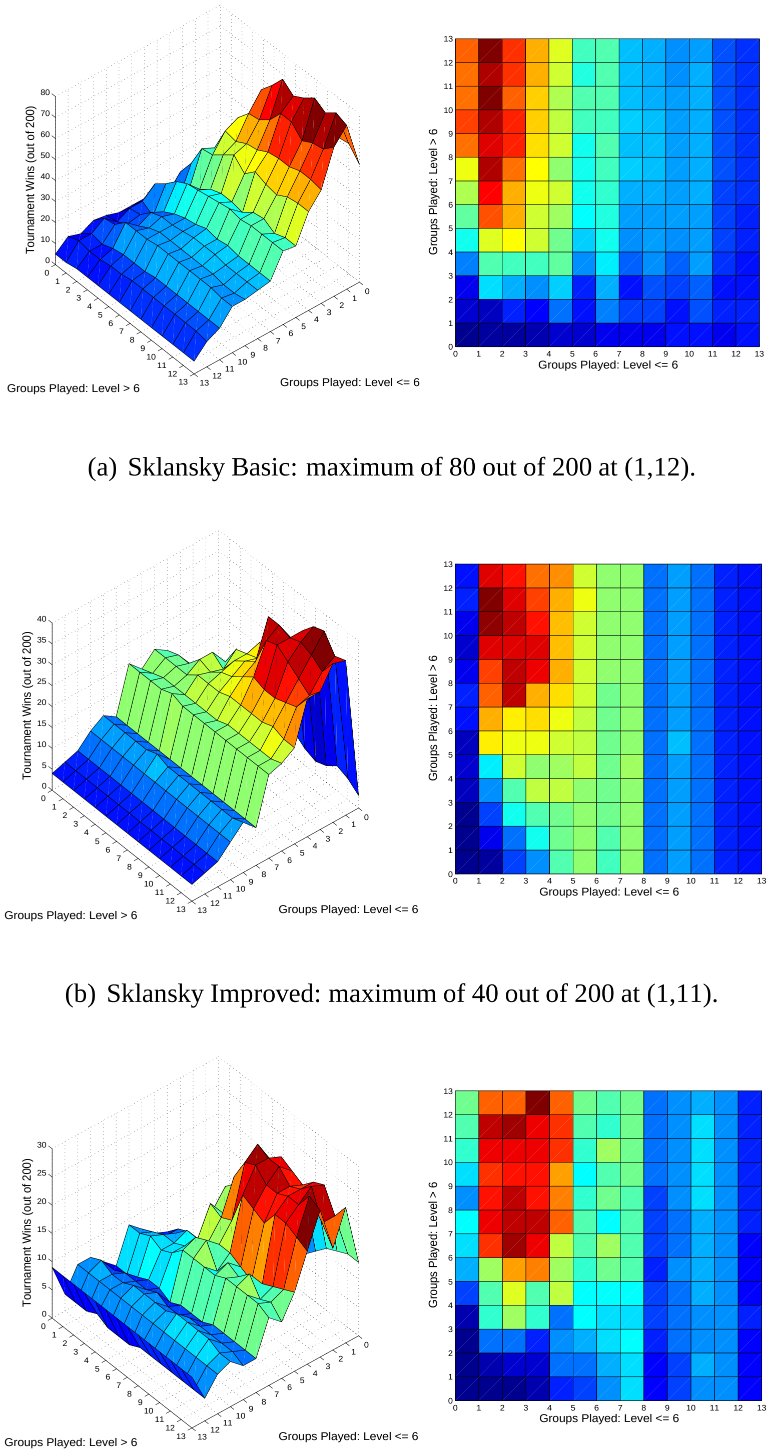
4.2.5 Hand Strength and Tournament Level

The next game-related information that we incorporate into our test player's decision making classifies the stage of the tournament. This factor allows the player to make a different decision with the same cards dependent upon which level has been reached.

The binary variable representing tournament level is allowed the two stages "early" and "late". With reference to Table 4.1 the former classification arises when the tournament is in a level up to and including level six. The latter criterion applies in levels seven through eleven.

The purpose behind these experiments is to demonstrate whether knowledge of the stage of a tournament affects the best strategy available to our test player, and in what ways. The non-academic poker literature reviewed in Section 3.2.1 suggests that, in general, a player should be more conservative in the early stages of a tournament and that only the best hands should be played. As a tournament progresses a player should be more aggressive with a larger number of starting hands.

This strategy has an accompanying rationale. Small blinds at the start of a tournament tend to produce small pot sizes, and these are not deemed worthy of risking



(a) Sklansky Basic: maximum of 80 out of 200 at (1,12).

(b) Sklansky Improved: maximum of 40 out of 200 at (1,11).

(c) Kill Phil Rookie: maximum of 27 out of 200 at (3,12).

Figure 4.5: Tournament wins against the three static opponents using hand strength and tournament level knowledge.

one's tournament survival for. As the tournament progresses and the blinds increase, then a player should become more willing to risk elimination since the pots won will represent an increasingly larger percentage of the chips in play.

The results of playing each of the 196 possible strategies over 200 tournaments against the three opponents are shown in Figure 4.5.

All three graphs show a remarkably similar shape. We can see that the maximum number of tournament wins are achieved when limiting the number of groups played to only the upper three in the early stages of a competition. In the concluding levels the test players that bet with a vast majority of their hands fare best.

As more groups are played in the early levels performance typically drops. For the late levels the results against all three static opponents show a relatively steady and high score for those strategies with a threshold of thirteen up to five. However, the win rate drops dramatically as the threshold increases still further. The players which act in such a way are being knocked out of tournaments as they wait to receive a hand sufficiently strong enough to bet with.

Again the maxima in these plots are off-diagonal. P-values from the method of paired comparisons are presented in Table 4.8. These verify that we should reject the null hypothesis that the difference between paired scores about the main diagonal is equal to zero against all three static opponents.

	SB	**SI**	**KPR**
P-value	< 0.0001	< 0.0001	< 0.0001

Table 4.8: P-values for the method of paired differences test applied to the hand strength and tournament level knowledge results.

Hence from the strategies giving rise to the maximal results in these experiments we observe a marked difference between the number of groups that should be played in early and late tournament levels. This distinction is in line with poker professionals' suggestions.

4.2.6 Hand Strength and Stack Size

The final piece of game-related information that we wish to assess is whether and in what ways knowledge of a player's stack size affects their tournament performance. For this set of experiments the binary decision variable represents whether the test player has few chips or many. To make this distinction we employ a method for measuring stack size previously mentioned in Section 3.2.2.

Harrington [60] refers to the ratio between one's stack size and the total value of the blinds as the value *M*. Large and small values of *M* signify that a player has a large or small stack respectively. To make our decision variable binary, we classify a small stack as one which has an *M* of less than or equal to five. This value is consistent with the definition given in the book.

One potential criticism of using this ratio is that it inherently incorporates knowledge about the tournament level, since the denominator of *M* sums the blinds. However, we still use this method of determining stack size since it is commonly used by poker practitioners. It is useful since it equates a player's stack to their remaining tournament life were they not to play a hand. If we assume that the blinds keep at a constant level for long enough, *M* represents approximately how many more orbits of the table a player can survive before being anted-away.

As with the previous experiments, we scored the total number of tournaments won by each of the 196 possible strategies by playing each in 200 tournaments against tables comprising the three opposing strategies. The scores are graphed in Figure 4.6.

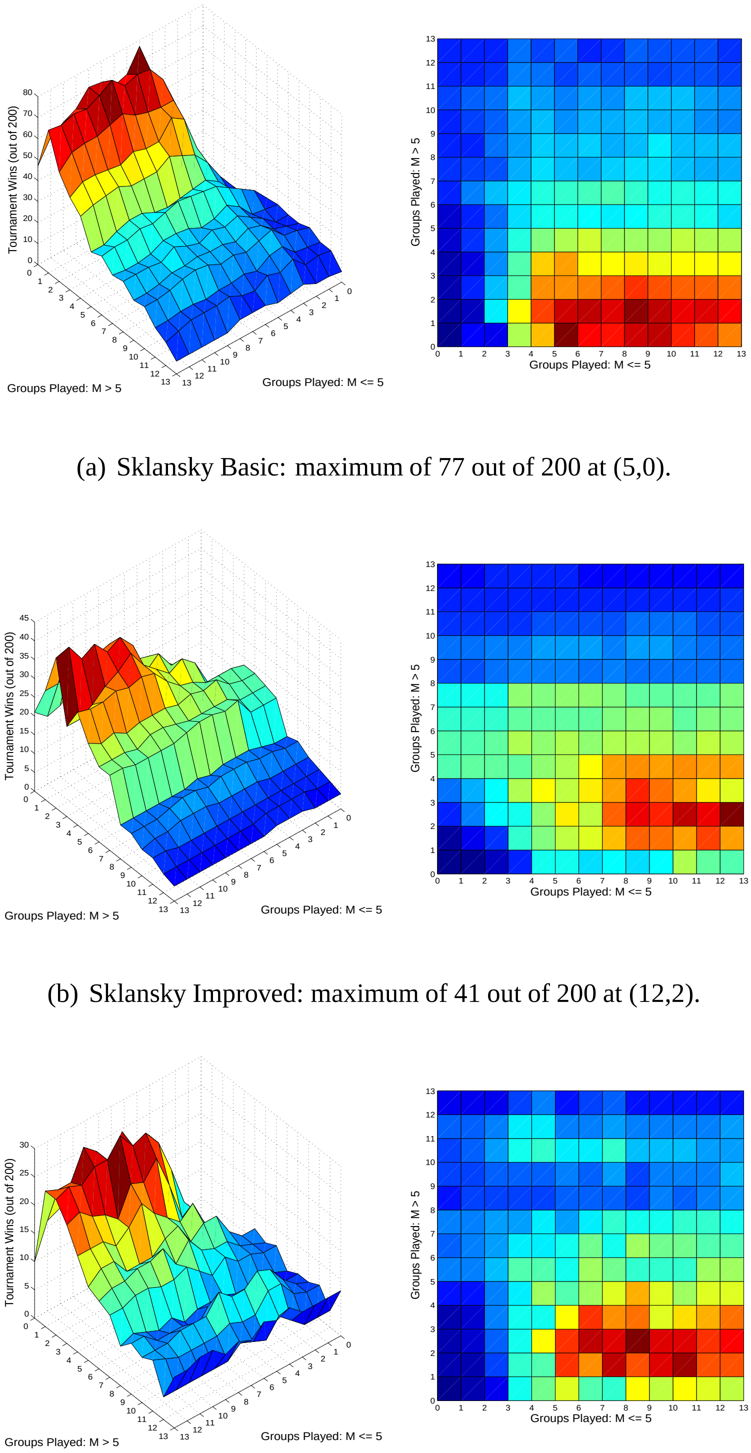
Examining the plots shows that against all three opponents the strategy which fares best is to play very few groups with a large *M*, but to play many more starting hands with a small *M*. The maximal results are typically to play the top two or three groups with a large stack, and anywhere between four and all thirteen groups with a small stack.

The non-academic poker literature suggests two differing styles for a player with a large stack. The first of these is to play more hands than usual, especially against opponents with a medium stack size. This is because these adversaries will often be faced with a decision which could end their participation in the tournament if unsuccessful. In this way the player with a large stack is capable of "bullying" his opposition.

The second recommendation for a player with a large stack is to play fewer hands than normal. Here the reasoning is that such a player is not in any immediate danger of tournament elimination from the increasing blinds. Having a large stack therefore allows that player the luxury of waiting for better hands and situations than normal. Additionally, this method reduces the risk that they will lose chips due to bad luck.

From the graphs we observe the latter of these two tactics in evidence. Playing too many hands with a large stack is seen to be detrimental to tournament success.

The case with a small stack is clearer in the non-academic literature. Here all



(a) Sklansky Basic: maximum of 77 out of 200 at (5,0).

(b) Sklansky Improved: maximum of 41 out of 200 at (12,2).

(c) Kill Phil Rookie: maximum of 29 out of 200 at (8,2).

Figure 4.6: Tournament wins against the three static opponents using hand strength and stack size knowledge.

authors agree that such a player should be willing to make a stand with a weaker starting hand than usual. This advice is particularly appropriate for situations in which the small stacked player is first to act. The benefit of betting in such a circumstance is that there is always the possibility that all subsequent players will fold, thereby winning the pot uncontested. The results from these experiments and those previously seen on players in early seating position mirror this advice.

For completeness we show the p-values in Table 4.9 resulting from the null hypothesis that the mean differences between observations paired by symmetry across the diagonal are equal to zero.

	SB	**SI**	**KPR**
P-value	< 0.0001	< 0.0001	< 0.0001

Table 4.9: P-values for the method of paired differences test applied to the hand strength and stack size knowledge results.

We reject the null hypothesis in all three cases. The conclusion that the test player's best strategy is affected by stack size knowledge parallels those from the inclusion of seating position, opponents' prior actions, and tournament level information.

4.3 Further Statistical Analysis of Exhaustive Simulation Results

The results presented in the previous subsections show that the inclusion of knowledge related to seating position, opponents' prior actions, tournament level, and stack size affect a player's best strategy in comparison to that founded on hand strength information alone.

The counter-strategies leading to the best results against the Sklansky Basic, Sklansky Improved, and Kill Phil Rookie strategies are summarized in Table 4.10.

It is clear that the state of each binary variable polarizes the test player's best counter-strategy in all non-control cases against all opponents. As we discussed, the interpretations of these strategies generally bear favourable comparison to suggested tournament play in the non-academic poker literature.

To extend our analysis we now assess what quantitative impact the inclusion of the extra information has on the highest scores achieved. Table 4.11 gives the percentage

Game Knowledge	**SB**	**SI**	**KPR**
Hand strength only	(4)	(4)	(4)
Hand and position	$(8,2)$	$(13,2)$	$(8,1)$, $(10,1)$ & $(11,1)$
Hand and prior bet	$(9,0)$	$(13,3)$	$(10,1)$, $(10,2)$ & $(12,3)$
Hand and level	$(1,12)$	$(1,11)$	$(3,12)$
Hand and stack	$(5,0)$	$(12,2)$	$(8,2)$

Table 4.10: Best strategies found by exhaustive simulations against the three static opponents.

of tournaments won by the best counter-strategies in each of the above scenarios.

Game Knowledge	**SB**	**SI**	**KPR**
Hand strength only	21.2% ± 3.1%	14.8% ± 3.1%	8.5% ± 3.1%
Hand and position	31.5% ± 6.9%	21.5% ± 6.9%	14.0% ± 6.9%
Hand and prior bet	71.5% ± 6.9%	26.5% ± 6.9%	14.5% ± 6.9%
Hand and level	40.0% ± 6.9%	20.0% ± 6.9%	13.5% ± 6.9%
Hand and stack	38.5% ± 6.9%	20.5% ± 6.9%	14.5% ± 6.9%

Table 4.11: Proportion of tournaments won by the best counter-strategies found by exhaustive simulations against the three static opponents, with corresponding 95% confidence intervals. Note that the values in the first row are calculated based on 1,000 tournaments, whilst all other rows used 200 tournaments.

First we note the consistently diminishing trend in tournaments won moving from left to right across the three opponents. This effect mirrors the opponents' increasing complexities.

Looking down the columns we observe that all best scores resulting from the inclusion of game-related information are higher than those based on hand strength knowledge alone. The greatest increases in tournaments won against all three opponents come from the inclusion of prior bet information. Against the Sklansky Basic strategy there is a tremendous improvement of over 230%, whilst gains of over 70% are observed against both the Sklansky Improved and Kill Phil Rookie opponents.

To formally evaluate these apparent gains we return to statistical hypothesis testing. We wish to test whether there is a statistically significant improvement in the proportion of tournaments won by those strategies which incorporate extra knowledge over those that utilize hand strength information alone. To do this we use the Z-test for the equality of two proportions [69].

In this test the null hypothesis states that the proportions π_1 and π_2 of tournaments won from two populations are equal, based on one sample from each population. With random samples of sizes n_1 and n_2, and corresponding proportions of tournaments won of p_1 and p_2, the test statistic is

$$Z = \frac{(p_1 - p_2)}{\left\{P(1-P)\left(\frac{1}{n_1}+\frac{1}{n_2}\right)\right\}^{\frac{1}{2}}}$$

where

$$P = \frac{p_1 n_1 + p_2 n_2}{n_1 + n_2}$$

Under the null hypothesis Z is approximately distributed as a standard normal. To test for an increase in the proportion of tournaments won we select the one-tailed test. At a 5% significance level the null hypothesis is rejected for Z-values greater than 1.65.

To improve the accuracy of the statistical tests we take each best counter-strategy and repeatedly play them against their respective opponents over a total of 5,000 tournaments. The effect of this is to increase the “signal-to-noise ratio” in the results above those returned through only sampling 200 tournaments. The proportion of tournament wins from these new experiments are shown in Table 4.12.

Game Knowledge	**SB**	**SI**	**KPR**
Hand strength only	20.1% ± 1.4%	11.3% ± 1.4%	6.4% ± 1.4%
Hand and position	28.3% ± 1.4%	14.3% ± 1.4%	9.2% ± 1.4%
Hand and prior bet	71.2% ± 1.4%	21.9% ± 1.4%	11.9% ± 1.4%
Hand and level	40.0% ± 1.4%	14.9% ± 1.4%	11.5% ± 1.4%
Hand and stack	32.7% ± 1.4%	16.4% ± 1.4%	10.7% ± 1.4%

Table 4.12: Percentage of tournament wins of the best found strategies against the three static opponents, with corresponding 95% confidence intervals. Note that all values are calculated based on 5,000 tournaments.

Before continuing with the statistical analysis, we first highlight an issue found by comparing these values with those in Table 4.11. It should be noted that we would not always expect the best counter-strategy found over 200 tournaments to score a similar proportion of successes over 5,000 tournaments. Although 200 tournaments is sufficient to differentiate between good and bad strategies, it is possible that the highest scoring counter-strategies have had a relatively large amount of luck compared to other high scorers. By allowing the luck to “even out” over a greater number of tournaments, therefore, such strategies will achieve a lower percentage of tournament wins. This is reflected in our results.

Returning to the analysis, the null hypothesis that we test is that the proportion of tournaments won with the inclusion of extra game-related information is the same as the respective proportion won based on hand strength knowledge alone. Therefore in the notation set out above both n_1 and n_2 take the value of 5,000. The proportion π_1 is taken from the hand strength row of Table 4.12, with the value of π_2 being taken from the row corresponding to the binary variable under investigation. The results of the Z-tests are summarized in Table 4.13.

Factor	**SB**	**SI**	**KPR**
Position	< 0.0001	< 0.0001	< 0.0001
Prior bet	< 0.0001	< 0.0001	< 0.0001
Level	< 0.0001	< 0.0001	< 0.0001
Stack	< 0.0001	< 0.0001	< 0.0001

Table 4.13: P-values for the proportion of tournaments won against the three static opponents with the inclusion of an extra factor compared to the respective hand strength knowledge alone score.

Given the low p-values observed we consistently reject the null hypotheses. There is sufficient evidence to show that the inclusion of extra knowledge in all twelve cases has benefited the test players, leading to significantly higher best scores.

4.4 Conclusions

Our preliminary investigations into all in or fold Texas hold’em tournament strategy have focused on testing whether the inclusion of game-related knowledge affects and improves the performance of a player above what they can achieve basing their betting actions on hand strength alone. To assess this we have used exhaustive simulation to enumerate over all possible counter-strategies to three plausible all in or fold systems taken from the non-academic poker literature.

We have demonstrated that knowledge of seating position, opponents' prior actions, tournament level, and stack size all affect a player's best strategy dependent upon the state of the respective binary variable. These were confirmed statistically. The interpretations of the best counter-strategies found were also seen to concur well with the recommendations of poker professionals in the non-academic literature.

The resulting high scores were also observed to be higher than those achieved through hand strength knowledge alone. To ensure thoroughness, we conducted hypothesis tests on the results and established their statistical significance. We therefore conclude that information relating to these four factors should all be important considerations in a player's decision making, and that a player can positively benefit from this knowledge.

The next logical step is to use all factors in concert. The exhaustive simulation approach is expedient in the cases examined here, but with every additional binary variable the strategy space, and therefore the time taken to enumerate over it, grows exponentially. For this reason we seek an alternative approach.

Chapter 5

Evolutionary Algorithms

Searching the enlarged strategy space resulting from the inclusion of multiple game-related factors becomes less feasible with the exhaustive simulation approach adopted in the previous chapter. This chapter starts with a discussion on its limitations, and introduces evolutionary algorithms as a potential replacement.

We continue by explaining the terminology used in EAs and its parallels with natural selection. We summarize the key components of the four major classes of EAs: genetic algorithms, evolutionary programming, evolution strategies, and genetic programming.

Following this we review previous research employing evolutionary methods to develop game players, and then focus specifically on their application to poker. The chapter concludes by illustrating how such methods can be implemented for a simplified poker, called the Jack-Queen-King game.

5.1 The Difficulties of Extending the Exhaustive Simulation Approach

We saw in the previous chapter how players who are able to differentiate their betting strategy based on extra game-related information fare better than those who simply act upon hand strength knowledge alone. The factors of seating position, opponents' prior bets, tournament level, and stack size were all shown to positively influence the number of tournaments that could be won against tables of three different all in or fold opponents. We now wish to determine whether a player who utilizes the complete knowledge set is able to score an even greater number of wins.

Recall that when we first started the simulations we represented a player's strategy by a single value (x). This integer was able to take a value from $\{0,1,...,13\}$, and denoted the threshold hand group below which a player would fold. Within this framework there were 14 distinct strategies.

We further extended the representation when we introduced the extra game information in the form of a binary variable. In this case we used the tuple (x,y), with the first value denoting the minimum playable hand group when the binary variable was in the first state, and the second value denoting the minimum playable hand group when the binary variable was in its second state. The space of possible strategies was then squared to 196.

If the factors occurred independently and in isolation it would be very simple to extend this representation in the following manner. We would be able to denote our test player's strategy using n factors by the tuple

$$\left(t_1^T, t_1^F, t_2^T, t_2^F, ..., t_n^T, t_n^F\right)$$

where t_i^T indicates the threshold hand group when binary variable i is true, and t_i^F gives the threshold hand group when binary variable i is false.

However, the game factors are neither independent nor do they occur in isolation. The states of every one of the four binary variables are simultaneously known to the test player at all points during the play of a hand.

Enumerating over all possible combinations of the binary variables, we find there are a total of $2^4 = 16$ unique game scenarios for any given starting hand within our framework. A strategy which includes knowledge of all the factors therefore needs to ascribe a betting action to any starting hand in all of these possible situations. Modifying the notation above, our test player can be represented as a 16-tuple

$$(t_1, t_2, ..., t_{16})$$

where t_i is the threshold value between moving all in and folding for unique scenario i.

Since each of the t_i can take one of 14 possible values, there are $14^{16} \simeq 2.18 \times 10^{18}$ distinct strategies. This strategy space is too vast to enumerate over. To run a simulation of 200 tournaments over every one of these scenarios, each averaging around four minutes, would take over 1.65×10^{13} years to complete.

The exponential explosion in the strategy space would be even further prohibitive should we wish to increase the number of dimensions, or expand our representation in any dimension. For example we may wish to augment the possible strategies by allowing the variables containing game-related information to take on more than just two states. One potential benefit of this would be to permit separate betting action thresholds for each individual level in tournament, not just “early” and “late” stages.

The greatly inflated run-times required to advance the exhaustive simulation-based approach reduces its desirability. What we require is a faster method of searching the strategy space; one which returns strong but not necessarily optimal solutions. For this reason we turn to evolutionary algorithms.

5.2 Introduction to Evolutionary Algorithms

The term evolutionary algorithm (EA) refers to a class of computational problem solving systems that are characterized by their use of models of evolutionary mechanisms taken from the fields of biology and genetics. Such techniques have been successfully applied to a diverse set of domains, including scheduling [82], protein folding [78], and games [66, 113]. Evolutionary computation is therefore used by a wide spectrum of researchers, from computer scientists and engineers, to physicists, chemists and biologists.

An EA is only usually employed when more efficient methods of searching the solution space are unknown. An algorithm specifically designed to solve a given problem will usually outperform an EA, and as such EAs are often used as a last resort when more suitable techniques are not available. This is frequently the case in problems for which the solution space is large and multi-modal.

The directed stochastic search within an EA seeks to balance the exploration of the solution space with the exploitation of previously found strong solutions. The heuristic nature of EAs is of particular use when it suffices to find a good, but not necessarily optimal, solution to a problem. The strength of an EA’s solution is often only limited by the available computing time. To improve the performance of an EA it is common to see them combined with local search techniques to form memetic algorithms [32].

In keeping with its roots, practitioners have borrowed terms from biology and genetics to describe the elements of an EA. A potential solution to a given problem is known as a *candidate* (also *chromosome* or *individual*). Each candidate consists of a collection of *genes*. These genes represent fundamental units in the structure of a solution, and may be encoded as binary, real, or alphabetic tokens. The value of a single gene is known as an *allele*. The *genotype* of a candidate denotes its specific genetic makeup, and an individual’s outwardly observable manifestation is termed its *phenotype*.

To assess the quality of a candidate, an EA employs a *fitness function*. This function scores how well a solution meets the problem by providing a mapping from any candidate to a real number. The use of a fitness function thereby allows for an ordering amongst many potential solutions, so that strong and weak candidates can be differentiated. An EA maintains many candidates together in a *population* of potential solutions.

Operators act on the population members according to evolutionary rules. As in biological evolution, two candidates can represent *parents* and combine to produce *offspring* through *reproduction*. This recombination operator is known as *crossover*, and parts of each parent's genotype are selected and fused to produce a new individual. One recurrent method for choosing the parents is fitness-proportionate selection [84], in which fitter individuals have a greater probability of being selected. Such methods lead to a "survival of the fittest" [36], in which genetic material from strong candidates is more likely to endure.

Diversity is often incorporated into the starting population of an EA by selecting randomly generated candidates. However, the acts of fitness-proportionate selection and crossover lead to a *convergence* in the material contained within the gene pool. To overcome this effect and maintain diversity the *mutation* genetic operator transforms a small number of the alleles in some candidates chosen by random chance. This operator ensures that new genetic material is able to enter into the population thus allowing for further exploration of the solution space.

EAs are usually implemented in a cycle involving the evaluation of all the candidates' fitnesses followed by the application of genetic operators to create a new population. One iteration of this loop is referred to as a *generation*.

Several methods may be employed to determine the termination of an EA. One is to conclude a simulation once a sufficiently good solution has been found, as determined by the experimenter. Another method is to use a stopping criterion based on a measure of the population convergence. The most common approach is simply to run the EA for a predetermined number of generations. In all cases the best candidate found over the span of the EA is returned as the proposed solution.

5.3 Subclasses of Evolutionary Algorithm

There are many computational techniques which fall under the umbrella term of evolutionary algorithms. The main ones are: genetic algorithms, evolutionary programming, evolution strategies, and genetic programming. A brief description of each subclass follows.

5.3.1 Genetic Algorithms

A genetic algorithm (GA) is an evolutionary computation model which closely mimics the evolutionary methods found in nature. GAs were first proposed by John Holland [67] at the University of Michigan, where much of the theoretical foundation for the algorithms subsequently originated.

A population of individuals is maintained by the GA, with each candidate represented by chromosomes which encode for the solution to a given problem. These chromosomes are typically of fixed length, although this is not mandatory. The canonical GA uses only binary digits for its representation of solutions, but it is possible to use any alphanumeric character to denote certain parameters or behaviours.

The implementation of a GA usually conforms to the following rubric. An initial random population is created, with the fitnesses of all candidates assessed using a fitness function. Following this a new generation is created by the application of genetic operators such as fitness-proportionate reproduction, crossover, and mutation. The old population is discarded, and the procedure iterates on the new population.

GAs have found particular application in timetabling and scheduling problems, but are often applied to solve many other kinds of global optimization problems.

5.3.2 Evolutionary Programming

Evolutionary programming (EP) was conceived by Lawrence Fogel [49], and shares many similarities to GAs. Unlike a GA, however, EPs tend to focus on the phenotypes of solutions rather than the genotypes. Candidates within an EP are not constrained to being character strings, and as such are often able to encode for the problem solution in more apt forms. Also, mutation is not limited to switching between a discrete set of possible values. Instead this operator often changes parts of a potential solution by statistical perturbations.

The methodology of EP is similar to that of a GA. An initial population of random individuals is created, and replications of these are mutated according to a statistical distribution. Note that unlike GAs, crossover is not normally found within an EP. The resulting offspring's fitnesses are evaluated, after which a tournament procedure determines which of the candidates is retained for the subsequent generation.

Like GAs, EPs are used in optimization problems where direct methods such as gradient descent are not applicable due to extreme unevenness of the fitness landscape.

5.3.3 Evolution Strategies

Rechenberg is accredited with originating evolution strategies (ES) in his work on optimization within engineering in the late 1960s [98]. The development of ES occurred independently from the work on EP, but the two subclasses of EA share many similarities. Both typically operate on real-valued encodings, and apply Gaussian mutations followed by a selection mechanism to determine which candidates survive to the next generation. ES differs from EP in two major concerns. Firstly, EP typically uses tournament selection whereas ES uses deterministic selection. Secondly, ES does contain recombination operators, as with GAs.

The original formulations of an ES use only one parent to generate one offspring per generation by the application of mutations. These are known as (1 + 1) ES. This procedure was later expanded to allow for multiple parents - $(\mu + 1)$ ES - and incorporated recombination. Schwefel [104] subsequently generalized the preceding work to create multi-membered plus and comma ES, denoted by $(\mu + \lambda)$ and (μ, λ) respectively. In this notation μ represents the population size, and λ the number of offspring created per generation. With plus strategies the parental generation is included in the selection procedure, whereas the comma strategies only select from amongst the offspring. In both of these formulations the individuals may recombine through random mating and are subject to mutation and selection.

5.3.4 Genetic Programming

Genetic programming (GP) is the final main subclass of EAs. Whilst the first related efforts came in the early 1980s [33, 115] the field only truly gained recognition on the publication of a book by John Koza in 1992 [79].

Traditionally, the individuals maintained within GP are programs expressed in a tree structure of suitable data values and functions. The programs are composed of elements of symbols which are deemed to be appropriate to the problem at hand. Originally GPs were often written in LISP, due to the compatibility between this programming language and the implementation of the necessary parse trees.

As with all EAs, GP is an iterative process which evaluates potential solutions by use of a fitness function. The crossover operator within GP uses fitness-proportionate selection to mate subtrees of two candidates and produce new offspring. The nature of the programs means that candidates within a GP are not of fixed length, and this flexibility provides GP with one of its greatest strengths over other EA techniques.

5.4 Evolutionary Algorithms Applied to Games

In Chapter 2 we examined many of the different techniques from the field of artificial intelligence which have been successfully applied to the development of game players. In this section we look at the utilization of evolutionary computation methods for the same purpose.

The use of such techniques is particularly appropriate for creating a player for a game. The rules of the environment are known and unchanging, and the space of possible solutions (in this case strategies) is extremely large and irregular. Only relatively trivial games lend themselves to an analytic solution, for example through the use of such methods as linear programming. Most commonly played games require other techniques which are able to search the strategy space and discover strong results. All of the EA techniques discussed above have been applied in the domain of games, and there follows a review of some of the most important results.

One of the first successful applications of EAs to games came with Robert Axelrod's research on the iterated prisoner's dilemma (IPD) [2]. The equilibrium for the non-iterated version of the game demands that both agents defect, since this strategy strictly dominates cooperating. However, the iterated variant introduces more complexity, since each player has the opportunity to "punish" their opponent for previous non-cooperative play. The question arises as to whether the incentive to defect is overshadowed by the threat of punishment.

Axelrod ran a series of tournaments, for which he invited many academic researchers to design a strategy for the IPD. The target for the competitors was to maximize the payoff across a large but unknown sequence of games. Each entry was allowed to maintain knowledge of its previous three games, upon which it could base a strategic selection for its next game.

The winner of the tournament was Anatol Rapoport's TIT FOR TAT program. This strategy cooperated on the first turn, but then repeated its opponent's action from the previous game thereafter. Playing in this way directly seeks cooperation with its adversary, since any non-cooperation will be punished by a defection on the succeeding turn.

Axelrod [3] then sought to find whether a GA could improve upon TIT FOR TAT's performance. He encoded strategies as a 64-bit string, with each gene representing the player's action for any of the possible prior three-game strategy pair sequences. From this he was able to discover strategies which outperformed TIT FOR TAT in his chosen static environment. Whilst not necessarily better than TIT FOR TAT against all possible opponents, Axelrod's evolved strategies showed the power of a GA in exploiting its given environment.

An interesting use of coevolving neural networks is found in the development of the checkers player *Blondie24* [27, 28, 48] by Kumar Chellapilla and David Fogel. A competing population of individuals is evaluated based on their ability to play the game using only the positions of the pieces on the board and the piece differential (i.e. the material superiority of one player over the other).

For the purposes of the program a checkerboard is represented by a vector of length 32, with each component corresponding to a permissible position on the board. Components in the vector are elements from the set $\{-K, -1, 0, +1, +K\}$, where 0 corresponds to an empty square, 1 is the value of a regular checker and K is the worth assigned to a king. The value of K is evolved by the algorithm. The sign of the value indicates which player the piece belongs to: positive for the player, negative for the opponent. A player's move is determined by evaluating the presumed quality of the resulting future positions.

After several months of coevolution the best neural network was seen to attain an expert-level status. The program's authors challenged the world champion program *Chinook* to a ten-game match, with *Blondie24* winning two and drawing four. Unlike its opponent, *Blondie24* contained neither expert rules nor databases of openings or endgame positions. Although losing the match, *Blondie24*'s performance was impressive and highlights how well evolutionary methods can do with little or no domain knowledge.

Another application of evolving neural networks is by David Moriarty and Risto Miikkulainen [85], for the purpose of strategy discovery in the game of Othello. The neural networks are evolved using a GA, with only the piece placements as inputs to the candidate solutions. The marker-based scheme employed allows the net architecture and connection weights to evolve. The lack of expert rules or heuristics means that any strategies that form do so purely through self-discovery of the game.

The networks are first tested against a player employing random moves, from which they quickly develop what is known as a "positional strategy" for the game. This way of playing essentially seeks to maximize the number of opponent's pieces reversed on each move, and is the way that novice human players often approach the game.

Following this the competitor program is itself encoded with a positional strategy. The newly evolved players then discover the more complex "mobility strategy" for Othello. This strategy, which is characteristic of human masters, seeks to limit the opponent's number of good moves on their next turn. Again the evolutionary approach shows itself to be capable of creating a game player of a high level.

Many other games have provided the test bed for research on evolutionary algorithms. Graham Kendall at the University of Nottingham has applied such methods to the African game of awari [40], cribbage [71], blackjack [72], and chess [73]. In all such cases the strategies evolved show themselves to be at least as strong as amateur human players.

One neoteric use of genetic algorithms is Colin Frayn's [51] research on the board game of Monopoly®. In this work a GA is used to determine a valuation for all of the properties on the board, both singly and as part of a complete set. Other game playing parameters are evolved, including whether or not to exit jail, which properties to mortgage in the event of bankruptcy, and the desired minimum cash position. The results of the GA give rise to very strong computer players which the author contends are superior to those found in commercially available Monopoly® software.

We noted in Section 2.3 that Go is arguably the most difficult board game to develop competent computer players for, and it too has been the subject of research applying evolutionary computation. Genetic algorithms are used in the research of Donnelly *et al.* [95] to evolve a Go position evaluation network for a 9-by-9 board. This reduced game is often played by beginners, and here helps to reduce the game space so that the experimental runs are less time-consuming. Crossover and mutation are employed to train the network, and the individuals compete in all-play-all tournaments.

Richards *et al.* [100] describe a method they christen SANE (Symbiotic, Adaptive Neuro-Evolution) to evolve neural networks capable of playing Go on similarly truncated boards with no pre-programmed knowledge. Two separate populations are evolved: one of neurons for the networks' hidden layer, and the other for blueprints of network architecture. Networks are evaluated, selected, and recombined based on their overall performance in the game. After initially playing moves randomly in the experiments, the program eventually develops some well known Go strategies. Neither Donnelly's nor Richards' evolved players exhibit noteworthy strength in their ability to play the game, but this is in keeping with all computer Go results at present.

5.5 Evolutionary Algorithms Applied to Poker

The first, and to date most extensive attempts to apply evolutionary computation to poker have been performed by Luigi Barone and Lyndon While at the University of Western Australia. In their work they develop poker players that are able to adapt strategically given inputs from their environment, such as their own hand strength, seating position, bet size, and a measure of their opponents' playing styles.

The simulations performed in their early research [4] use a simplified ring game Texas hold'em with limit betting. The adaptive player uses the "competencies" of hand strength, seating position, and risk management to effect decision making. Each component generates a probability triple over each possible action (bet, call, or fold), which are then passed to a "resolver". The resolver weights the inputs and determines which action is made.

One evolving player is seated at a table of static opponents encoding expert rules. Many different types of opponent are used to cover a range of possible playing styles, although each table only consists of multiple copies of one such adversary. Evolution on the weightings of each competency and the maintenance of the population of candidates is performed using a (1 + 1) evolution strategy. The results show that the adaptive players are able to increase their strength sufficiently to compete against and beat their opposition.

Further research by Barone and While [5] alters the decision making structure and dispenses with the resolver. Instead, the population of candidates is segmented into a hypercube with dimensions relating to seating position and risk management. Each candidate in a hypercube element consists of real values which are used as inputs into a series of formulae to calculate the probabilities of a bet, call, or fold. Evolution occurs on these values and determines the player's action. During the play of a game, a candidate is selected from the appropriate hypercube segment to carry out a betting action. After all candidates have been cycled through a predetermined number of times the (1 + 1) evolution strategy creates a new generation.

The work of Barone and While gives encouragement to the use of evolutionary algorithms for designing a poker player. There are many possible extensions to their work, such as using a mixture of opponents at the same table and employing competitors which themselves use non-stationary strategies. Whilst their research solely focuses on the ring game format, its findings are propitious for the applicability of EAs to tournament poker.

In similar work by Graham Kendall and Mark Willdig [74], evolutionary methods are shown to be able to learn to play a simplified draw poker game. Candidates from the population are again played at tables containing opponents of different styles, and the adaptive players are seen to adjust their strategies appropriately to each situation. As in the evolutionary poker research cited above, however, no results concerning the performance of the resulting players against human opposition are published. In the work of both Barone and Kendall the encodings of hand strengths used are very coarse, and it is almost certainly the case that the players found would not match the relative abilities of the GAMES Groups' poker programs discussed in Chapter 2.

Texas hold'em has also been used in research on coevolutionary techniques by Jason Noble [91]. Rather than play against static opposition, his work seeks to develop strong poker players through self-play. The author states that the resulting individuals do not possess strong playing abilities when measured against human opposition. However, the focus of this study is the comparison of coevolutionary methods and is not intentioned to produce expert-level poker strategies.

In further work, Noble [90] uses a more detailed representation for the individuals' playing strategies. Rather than the rule-based system of threshold hand strengths and desired betting levels in the previous paper, this work employs a sparsely encoded neural network with a large number of possible inputs. The results show that deterministic crowding works well in conjunction with Pareto coevolution for maintaining a diverse population and improving strategies. The algorithm was not particularly successful in finding strong poker strategies, but again this was not the specific aim.

This review confirms two salient points for our present research. Firstly, EAs have been shown to work well in application to developing computer games players. Secondly, such techniques have been applied to various forms of poker, but none have so far investigated the game within a no-limit tournament setting.

5.6 An Example of Evolutionary Algorithm Design for a Toy Poker

We next illustrate the implementation of an EA in application to the Jack-Queen-King game (JQK game) [105], before returning to pre-flop Texas hold'em in the following chapter. The ensuing subsections introduce this toy poker, discuss the implementation issues, and analyse the results of some evolutionary experiments.

5.6.1 Jack-Queen-King Game

The Jack-Queen-King game is a simplified form of ring game poker with limit betting for two players, hereafter known as Player 1 and Player 2. Both competitors ante \$1, then receive a card from a deck consisting of a jack, a queen, and a king.

Player 1 is first to act, and after looking at his card may either bet \$1 or fold. If he folds, Player 2 wins the pot. If he bets, Player 2 must then either call for \$1 or fold. Should Player 2 fold, Player 1 wins the pot. If Player 2 calls, the winner is determined by a showdown (king $\succ$ queen $\succ$ jack).

The extensive form of the JQK game is shown in Figure 5.1. In this diagram the conditional probabilities of the cards dealt are shown along the upper branches of the tree. The actions f and b for Player 1 correspond to a fold and a bet respectively. Similarly, the f and c for Player 2 relate to fold and call actions.[1]

The presence of imperfect information in this toy poker is relayed by the multi-nodal information sets, shown as dashed ovals. At any point during the game a player knows which information set within the tree he currently resides given his card and the previous betting action. However, he cannot identify with certainty the precise node since he does not know his opponent's card.

The payoffs to Player 1 are given at the terminal nodes, and are the negative of those pertaining to Player 2.

The optimal strategy for each player in the JQK game can be determined by the use of linear programming. Firstly, a payoff matrix is constructed for all possible pairs of pure strategies available to each player. A pure strategy, Z_i, for a player is a triple (z_J, z_Q, z_K) ascribing an action to each information set. Each $z_i \in \{f, b \vee c\}$, with the latter element dependent upon whether the player acts as Player 1 or 2.

[1] Note that since the players act in sequence, a competitor may win pots as Player 2 with a card that he would have folded to a bet.

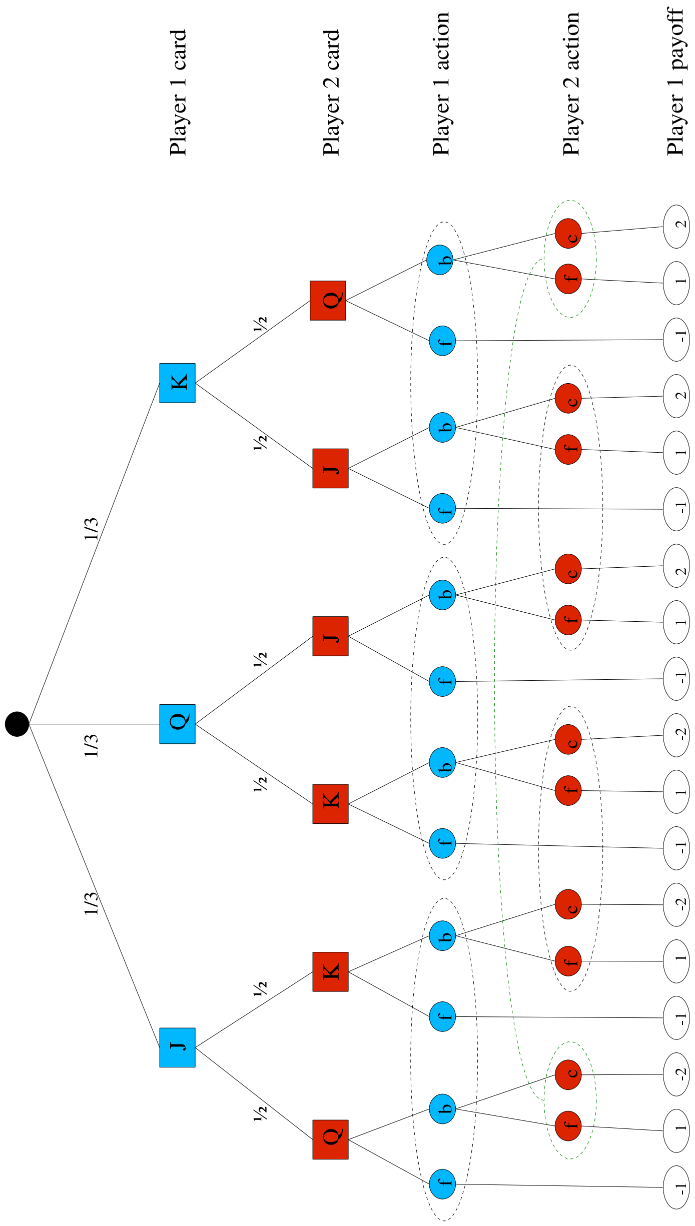


Figure 5.1: Extensive form for the JQK game.

The payoff matrix to Player 1 for the JQK Game is shown in Table 5.1.

	(f, f, f)	***(f, f, c)***	***(f, c, f)***	***(f, c, c)***	***(c, f, f)***	***(c, f, c)***	***(c, c, f)***	***(c, c, c)***
(f, f, f)	-1	-1	-1	-1	-1	-1	-1	-1
(f, f, b)	$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{1}{6}$	$-\frac{1}{6}$	0	0	0	0
(f, b, f)	$-\frac{1}{3}$	$-\frac{5}{6}$	$-\frac{1}{3}$	$-\frac{5}{6}$	$-\frac{1}{6}$	$-\frac{2}{3}$	$-\frac{1}{6}$	$-\frac{2}{3}$
(f, b, b)	$\frac{1}{3}$	$-\frac{1}{6}$	$\frac{1}{2}$	0	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{5}{6}$	$\frac{1}{3}$
(b, f, f)	$-\frac{1}{3}$	$-\frac{5}{6}$	$-\frac{5}{6}$	$-\frac{4}{3}$	$-\frac{1}{3}$	$-\frac{5}{6}$	$-\frac{5}{6}$	$-\frac{4}{3}$
(b, f, b)	$\frac{1}{3}$	$-\frac{1}{6}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{6}$	$-\frac{1}{3}$
(b, b, f)	$\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{1}{6}$	$-\frac{7}{6}$	$\frac{1}{2}$	$-\frac{1}{2}$	0	-1
(b, b, b)	1	0	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$	1	0

Table 5.1: Expected payoffs to Player 1 in the JQK game.

To solve the game constraints are placed on the pure strategies such that the sum for each player totals one. This is because we wish to find the probabilities with which each should be played. A software package such as Maple can then solve the constrained optimization problem to yield the optimal strategies for each player shown in Table 5.2.

Player 1	$\frac{2}{3}\,(f,b,b)$ and $\frac{1}{3}\,(b,b,b)$
Player 2	$\frac{2}{3}\,(f,f,c)$ and $\frac{1}{3}\,(f,c,c)$

Table 5.2: Optimal strategies in the JQK game for each player.

We observe that the optimal strategies for both players take the form of mixed strategies. Player 1 should always bet the queen and king, and the jack only one third of the time. Likewise Player 2 should always fold the jack, call one third of the time with the queen, and always call with the king.

Both the explanation and analysis above describe a single hand of the JQK game. To imbue this with greater realism we now suppose that the two players compete in repeated hands, with the deal and thus the seating position of each alternating. Each competitor is therefore required to act cyclically as Player 1 and Player 2.

5.6.2 Player Representation

When implementing an EA there exists a large amount of flexibility open to the practitioner in determining the exact formulation of the algorithm and its elements. The first decision that is required concerns the representation of the individuals. Arguably the simplest EA encoding of a JQK game player is to use the canonical genetic algorithm [56] representation containing six bits. The ordering of the genes on the chromosome is discretionary. Here we choose the first three bits to denote the player's action when he acts as Player 1, with the genes relating to playing the jack, queen, and king in that order. This representation is then repeated for the following three genes for the betting actions as Player 2. This representation is summarized in Table 5.3.

Gene	1	2	3	4	5	6
Player	1	1	1	2	2	2
Card	J	Q	K	J	Q	K

Table 5.3: Encoding of a strategy for the JQK game.

Within the canonical GA each gene can take on an allele of one of two values. A `0` signifies that the player folds with the given card in the appropriate position. An allele of `1` relates to two different actions depending upon the position of the player. A `1` denotes either a bet for Player 1, or a call in the case of Player 2.

With this representation all possible game strategies can be encoded as a chromosome. As an example we examine the candidate `011001`. With reference to Table 5.3, we first observe that this individual will never bet the jack and always bet the king. The second and fourth alleles relate that this player will bet the queen as Player 1, but not call a bet with it as Player 2.

5.6.3 Implementing the EA

In accordance with the algorithm design the next stage is to ascribe the evolutionary operators to the GA. These elements are set out in Algorithm 1.

As previously mentioned, the exact specifications used within the algorithm are left to the practitioner's discretion. Parameters relating to the sizing of the population, rates for the crossover and mutation operators, and the selection of the termination criterion are all required. Several theoretical results have been expounded, but the choices made are often based either on values that are known to have worked well in previous experiments, or are designed to be compatible with the available computational power or a desired run-time.

Algorithm 1 A Genetic Algorithm for the JQK Game

```
Initialize the population
while termination criterion unsatisfied do
    while new population not full do
        Evaluate the individuals' fitnesses
        Select individuals from the current population
        Perform crossover and reproduction on the selected individuals
        Perform mutation on offspring
        Add offspring to new population
    end while
end while
Return best individual found as solution
```

To continue the example of the JQK game we select a population of just ten individuals. A result of Goldberg *et al*. [57] suggests using a population size of the order of the number of genes over which the algorithm will operate.

We shall use the GA to evolve strategies against a static opponent. The chance element of the deal of the cards means that one hand is not sufficient to give an accurate assessment of whether one player is better than another. To reduce the effects of luck, we require that an individual plays against the opponent for 100 hands. Therefore the natural fitness function to use in this example is the total number of hands won. This provides any candidate with an evaluation in the interval $[0, 100]$.

For the recombination operator we choose simple one-point crossover. This takes two parents and splits each at the same randomly chosen point along the chromosome. An offspring is formed by fusing the first part of one parent with the second part of the other. Suppose, for example, we had the parents `101010` and `110001` and the algorithm determines that each should split between the second and third genes. The child of such a reproduction taking the first two genes of the first parent and the last four genes of the second would be `100001`. The crossover rate used within the GA is chosen to be 70%. This number is in keeping with values found to work well in many other experiments.

The mutation operator in a canonical GA flips a selected bit. However, we are still required to choose the rate at which such mutations occur. High mutation values are often found to be disruptive in the search for strong solutions, and so here we choose a rate of 5% in keeping with similar prior research.

For the termination criterion we choose to run the GA for twenty generations.

Again this number is discretionary, but is found to be sufficient for the given example.

5.6.4 Results and Conclusions

The static opponent we use to illustrate the GA is `001001`. This can be interpreted as an extremely cautious player, since he will only bet and call with the best possible card and fold otherwise.

The best solutions found using the above implementation are the individuals `111000` and `111001`. The strength of these candidates can be understood in the context of the counter-strategies that they encode for in comparison to the static opponent's strategy.

Both of these players will bet every card as Player 1. This has positive expectation since the static opponent will only call with the king. Similarly, both solutions will never call a bet as Player 2 with either the jack or queen. This is because the opponent only ever bets the king as Player 1. Notice that although both solutions differ in the final gene, this value is rendered irrelevant due to the opponent's strategy. That player will fold the jack and queen every time as Player 1, and so any competing individual will not be required to act if it is dealt the king.

This result can be validated mathematically by calculating the expected payoffs for each possible counter-strategy against the `001001` opponent. These are shown in Table 5.4.

P1 Strategy	P1 Payoff	P2 Strategy	P2 Payoff
(f,f,f)	-1	(f,f,f)	$\frac{1}{3}$
(f,f,b)	$-\frac{2}{3}$	(f,f,c)	$\frac{1}{3}$
(f,b,f)	$-\frac{5}{6}$	(f,c,f)	$\frac{1}{6}$
(f,b,b)	$-\frac{1}{6}$	(f,c,c)	$\frac{1}{6}$
(b,f,f)	$-\frac{5}{6}$	(c,f,f)	$\frac{1}{6}$
(b,f,b)	$-\frac{1}{6}$	(c,f,c)	$\frac{1}{6}$
(b,b,f)	$-\frac{2}{3}$	(c,c,f)	0
(b,b,b)	0	(c,c,c)	0

Table 5.4: Expected payoffs as Player 1 and Player 2 for each possible counter-strategy to a `001001` opponent in the JQK game.

The profit-maximizing strategy as Player 1 is seen to be (b,b,b), whilst both (f,f,f)

and (f, f, c) are equally optimal as Player 2. Concatenating these strategies leads to the two best evolved solutions cited above: `111000` and `111001`. This confirms that the GA has successfully discovered the global optima.

The example we have used to illustrate an EA is very simple. However, it serves to demonstrate the basic methodology and implementation requirements of an EA. Also, we gain confidence in the technique since the EA finds global optima in this small test case.

We can also envisage how this example could be extended for the JQK game played within a tournament. The length of the chromosomes could be enlarged to allow for different betting actions based, for example, on the tournament level as we did in Chapter 4. A simple illustration of this is given in Table 5.5, expanding the encoding used in Table 5.3.

Gene	1	2	3	4	5	6	7	8	9	10	11	12
Player	1	1	1	2	2	2	1	1	1	2	2	2
Card	J	Q	K	J	Q	K	J	Q	K	J	Q	K
Level	E	E	E	E	E	E	L	L	L	L	L	L

Table 5.5: Encoding of a tournament strategy for the JQK game. Tournament levels could be divided into early (E) and late (L) categories to allow for different actions.

We develop this idea when returning to our more realistic all in or fold pre-flop Texas hold'em test bed in the following chapter.

5.7 Summary

Evolutionary algorithms replicate mechanisms found in nature to derive solutions to difficult problems for which more effective approaches are unknown. They are strong general purpose problem solvers and, as such, have a wide range of applicability. Four main subclasses of EA exist, although the different strands share many similarities in their implementation and procedures. EAs are not simply black box techniques, and their use requires many specifications and parameterizations from the practitioner.

The computer game playing domain has benefited greatly from the application of EAs. Economic games such as the iterated prisoner's dilemma and classic games like checkers have all had EAs used on them to uncover strong strategies. The resulting players may not in some cases be as strong as those derived through other artificial intelligence techniques, but the lack of a requirement for domain knowledge means that the players are not unintentionally handicapped by any imperfections in pre-programmed expert rules.

In recent years poker has started to be used as a test bed for evolutionary algorithms. The ramifications of this work show promise, but the dearth of results against either human or externally developed computer opposition leaves the strength of the resulting programs open to question.

It is also the case that no evolutionary computation research has yet been applied to poker in a tournament setting. The following chapter addresses this directly.

Chapter 6

Discovering More Complex Strategies Using Evolutionary Algorithms

In this chapter we employ evolutionary algorithms to discover more complex strategies for our pre-flop Texas hold'em tournament domain. This we achieve by combining all four game-related factors found to positively influence decision making in Chapter 4. Note that this is non-trivial, since the best counter-strategies shown in Table 4.10 contain potential conflicts for all-encompassment. For example, if playing against the Sklansky Basic opponents in the latter half of a tournament with a large stack size, the best counter-strategies would suggest playing any hand in the top twelve groups due to the tournament level, and contradictorily folding all hands because of the large stack.

We first explain how the all in or fold tournament poker problem can be encoded within an EA, and discuss the issues of representation, genetic operators, and parameterization. Analysis of the evolutionary runs addresses whether use of the amalgamated factors yields stronger tournament poker players than those which used only a single game factor found previously, and whether an EA can resolve potentially conflicting inputs such as in the example above.

6.1 Evolutionary Algorithm Experimental Framework

Evolutionary algorithms are general purpose problem solvers that use a guided stochastic search to optimize a fitness function which assigns a numerical value to potential solutions. The different types of EA were outlined in the previous chapter, but the choice of which to apply to a given problem is left to the practitioner's discretion. Often this decision can be founded on the representation of candidate solutions employed.

Once this has been determined there needs to be a selection on which genetic operators to use, and choices made on the parameterization of the EA.

6.1.1 Strategy Representation

In Section 5.1 we saw how a strategy for our all in or fold tournament poker environment could be represented by the tuple

$$(t_1, t_2, \ldots, t_{16})$$

with the threshold value between moving all in and folding for unique scenario i given by t_i.

On removing the brackets and commas this encoding is familiar as the representation of a chromosome we used in the genetic algorithm at the end of the last chapter. As in all EAs it is necessary for the genes to encode some behaviour or parameterization such that together they form a candidate solution to the problem. In our example all that is required to complete the representation is to form an injective mapping between the genes and the sixteen unique game situations. Then when a test player is competing in a tournament, the states of the four binary variables will indicate which gene encodes the threshold value for that particular scenario. A comparison of their own hand's starting group with the determined threshold value will then return a betting action of either all in or fold.

To make the injection between genes and scenarios we use the power set of the binary variables, as shown in Table 6.1. Generally, the mapping used can have consequences on the effects of certain crossover operators within an EA. Here this is not a concern as we shall employ uniform crossover, discussed shortly.

The meaning of each binary variable's state is explained in Table 6.2. The designations employed echo those of Chapter 4.

Rather than allowing the alleles to take integer values from the set $\{0, 1, \ldots, 13\}$, we choose instead to let them take real numbers on the interval $[0, 14)$. The exact reason for this will be explained in an ensuing subsection on the mutation operator. To translate an allele to a betting action we simply take the floor of its value as the respective threshold hand group. For example, an allele of 3.141 has a floor of 3, and so instructs the player to move all in with any hand contained within the top three groups and fold otherwise.

Now that we have formalized the strategy representation within the chromosomes, we examine the other issues involved in constructing our EA.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
P	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
L	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
S	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 6.1: Classification of potential game scenarios for use within the chromosomes of the EA. Each column heading is the respective gene number, with the state of each binary variable covering seating position (P), opponents' prior bet (B), tournament level (L), and stack size (S) shown by row.

Binary Variable	**State = 0**	**State = 1**
Seating position (P)	Early position	Late position
Opponents' actions (B)	No prior bet	Prior bet
Tournament stage (L)	Level ≤ 6	Level > 6
Chip stack amount (S)	$M \leq 5$	$M > 5$

Table 6.2: Explanation of the binary variables' states used in Table 6.1.

6.1.2 Population Initialization

Evolutionary algorithms rely on a population of candidates to maintain a diversity of potential solutions to explore the fitness landscape. In constructing an EA it is necessary to decide how large a population to maintain, and how this should first be initialized.

Deciding upon the exact size of population is the responsibility of the practitioner. Broadly speaking, however, it is important that the number of candidate solutions employed in an EA is neither too small nor too large. If too few individuals are used the EA will suffer from rapid convergence within the population. This means that individuals will quickly come to resemble one another, with the result being that only a few localized areas of the strategy space are explored. At the other extreme, a population of too many individuals suffers from a different problem. Each individual is required to be scored by way of the fitness function, and hence the more individuals maintained the longer this takes. An overly large population size vastly increases the run-time of each generation of the algorithm.

For the canonical GA a useful rule of thumb is to use a population size comparable to the number of genes within a chromosome [57]. Although we do not use the simple binary encoding of the genes found within the canonical GA in our representation, it was decided to use a population of 20 based upon the same reasoning.

In creating the initial population there are two main possibilities to select from. The first is to create a population of random individuals by randomly assigning a feasible allele to every gene of all chromosomes. The second is to seed the population by introducing some candidates which are pre-defined. Seeded individuals can be used as a way of introducing prior knowledge of likely strong solutions. This can help to reduce the time taken to termination of the algorithm, since the fruitless explorations performed by some of the randomly created individuals will be removed. The disadvantage of using seeded individuals is that they may handicap the EA by unintentionally providing poor candidate solutions. For example, individuals known to be locally strong may obfuscate the search from finding globally optimal strategies.

In our experiments it would be very simple to seed the initial population. By extending the representations of the best found strategies from the previous exhaustive simulation experiments we could create potential starting points for the search. For example, the best strategy discovered earlier to play against the Sklansky Basic opponents using hand strength and seating position knowledge was $(8,2)$. This relates to using the threshold hand groups of 8 in early position and 2 in late position. To convert this into a strategy for the chromosomal representation requires that we repeat these two alleles in the appropriate genes found by reference to Table 6.2. This example would yield the chromosome:

```
8 8 8 8 8 8 8 8 2 2 2 2 2 2 2 2
```

Rather than seed the population with prior solutions we choose to initialize the EA with random individuals. At the possible cost of increased run-times we remove the potential for adversely biasing the search with seeded candidates.

6.1.3 Selection and the Fitness Function

In common with the biological processes which EAs mimic, individuals from the existing population give rise to a subsequent generation. How the individual candidates are chosen is based on the process of selection.

Typically the method used for selection within an EA is based on each candidate's fitness. The purpose of a fitness function is to quantify each solution's ability to optimize the problem at hand. Therefore the fitness function simply needs to provide a mapping from the strategy space to a number, such that any possible solution can be quantified and ranked.

The form that the fitness function takes is heavily problem-dependent. In our domain of tournament poker we have a simple and natural possibility. We are ultimately interested in how well a particular individual within the population fares as a tournament poker strategy. To do this we measure each strategy's ability over several tournaments, since random chance affects the results of any individual competition. Re-sampling candidates is a commonly employed method for searching noisy environments [7, 20, 97]. In keeping with the experiments in Chapter 4 we choose to play each strategy off over 200 tournaments. The fitness of an individual is then simply the total number of tournaments won.

Alternative fitness functions could be employed. Rather than simply counting one point for a tournament win and zero otherwise, we could assign different numerical values depending upon the position finished within a game. For example, we could use a fitness function which gives ten points for a tournament win, nine for second place, and so on down to one point for last.

Similarly we could use a fitness function which mirrors the payoffs to a regular ten-player tournament, in which first place receives 50% of the prize money, second place 30%, and third 20%. A function which awards five points to first place, three to second, two to third and zero otherwise would suffice for this purpose. Ultimately the choice is dependent upon our aims, and what we consider to be the precise problem that we wish to solve. In our experiments we continue to use the one point per win fitness function as in the exhaustive simulations, for reasons explained in Section 4.1.1.

6.1.4 Reproduction

The method by which an EA creates successive generations of individuals is known as reproduction. The typical procedure by which candidate solutions reproduce consists of

- Fitness-based selection
- Crossover (also called recombination), and

- Mutation

Two individuals are selected from the population as parents, and combine their genes through crossover to produce a child. This offspring solution may then undergo mutation, which randomly changes some of the chromosome's gene values. The intent behind the method is that fit parents will combine to produce even fitter children, thus driving the optimization on to new maxima. Maintaining many of the genes of highly fit individuals allows the next generation to exploit previously found regions of the search space, whilst the inclusion of crossover and mutation also permits investigation of hitherto unexplored strategies.

Details regarding the specific implementation of reproduction within our experiments are given in the next three subsections.

6.1.4.1 Fitness-Based Selection and Elitism

All EAs search the space of potential solutions through a careful trade-off between exploration and exploitation. Fitness-based selection is a method for ensuring that the population maintains many of the genes of previously found strong solutions. Candidates with a higher fitness are more likely to be propagated into the next generation.

Two commonly applied methods of fitness-based selection are roulette-wheel selection and tournament selection [84]. In the former the fitness of each individual is used to assign a proportionate probability of selection. Fitter solutions are more likely to be selected, but less fit individuals are not barred from becoming parents.

Tournament selection is the method used in our experiments. As with roulette-wheel selection fitter individuals have a higher probability of becoming parents, but the processes for this are somewhat different. Our tournament selection randomly chooses two individuals as the first potential parent. The candidate with the highest fitness amongst these is selected. The process is repeated for the second parent, with the proviso that the two parents should not be identical.

Elitism is an optional constituent of an EA. It is a method by which certain chromosomes are maintained unaltered from one generation to the next. Its use can have both positive and negative effects on an evolutionary run. Of benefit is that very strong solutions are kept unchanged within the population. The components of strong candidates thereby remain within the gene pool. A potential downside to elitism is that it can encourage premature convergence. This is particularly problematic with a small population size.

After several preliminary tests were performed both with and without the use of elitism, it was decided to use it such that the two fittest individuals pass through to the next generation unaltered. There was little difference observed in the results of EA runs both with and without a small number of elites. However the use of a larger number was seen to lead to a faster rate of convergence.

6.1.4.2 Crossover Operator

Crossover is employed in an EA to share genes amongst candidates in the population. After fitness-based selection has chosen the two parents, an associated rate determines the probability of them undergoing crossover. Without crossover the parents pass into the next generation unaltered, otherwise they undergo a process described below to produce offspring. Our experiments use a crossover rate of 70%, a figure which is lower than those commonly employed [41, 58].

Different styles of crossover have been used in previous academic research. Perhaps the most commonly employed is simple one-point crossover, and was the one employed in our investigation of the JQK game. In this method the two parents are split between the same two genes along the chromosomes, with the genetic material to the left-hand side of one fusing to the right-hand side of the other. This technique can be extended to *n*-point crossover, by which several genetic swaps are made between the parents.

One-point crossover is particularly desirable when there is some implicit ordering along the chromosome so that the adjacency of consecutive genes is somehow important within the representation. In our situation this is not the case. Referring back to Table 6.1 we observe that only one of the four factors (that of seating position) has like binary states grouped together along the chromosome. A split in the chromosome between the eighth and ninth genes would cleanly separate the early and late seats. However, such a cleave would have mixed effects on the other three factors since their binary variables' identical states are not contiguous.

Rather than this operator, we choose to employ uniform crossover. This scheme almost eliminates the possibility of selecting contiguous blocks of genes in reproduction, and hence helps to search wider portion of the solution space. With uniform crossover the parent chromosomes are compared and similarly numbered genes are swapped with a fixed probability. We use a probability of 0.5, meaning that on average half of the genes are crossed over. The result of applying uniform crossover is two child offspring, each of which contains part of the genetic material from both parents.

6.1.4.3 Mutation Operator

The final operator used within a typical EA is that of mutation. This scheme mirrors its biological precedent, with some amount of genetic material in the population changed every generation through chance. This operator ensures that there is a positive probability of new material entering the gene pool, and helps both to search the solution space and guard against convergence.

In the canonical GA, whereby the representation of chromosomes consists of several binary digits, the operator's effect is straightforward. A fixed, usually small, percentage of the genes of the offspring is selected for mutation. In these cases the allele on the gene is flipped, with a `1` becoming a `0` and vice versa. For encodings allowing for a greater number of alleles than just two, slightly more elaborate mutation operators are required.

In our problem we have chosen to encode the strategies using real numbers on the interval $[0,14)$. Before explaining why, we consider the simpler alternative of using the integers $\{0,1,...,13\}$. If we did so, we could construct a mutation operator which flipped values in a similar way to the canonical GA. In our case the mutation would need to swap a given allele to one of the 13 alternate possibilities with equal probability.

This mutation could have been used, but in doing so we would lose some of the knowledge that we have contained within the representation. The hand group numbering means that players with similar numerical encoding reside within local areas of the strategy space. We can assist the search by biasing it to have a greater probability of exploring "close" strategies, and a lower probability of choosing those "further away". The uniform mutation operator, by contrast, is equiprobable in this respect.

To effect this kind of mutation the zero-mean Gaussian distribution is often used. The common bell-shaped distribution ensures that small absolute changes are more likely than large ones. Since samples from a Gaussian distribution are real-valued, we use a real-valued encoding for the strategies. Non-integer alleles are converted into hand group thresholds by taking their floor value.

A Gaussian with a standard deviation of two was employed. This parameter was chosen arbitrarily, and no other choices were evaluated. Without further modifications it is possible for some shocks to take the allele value outside of the allowable $[0,14)$ range. For example, an allele of 0.278 taking a negative shock of 0.5 would result in an impermissible value of -0.222. To prevent this we use reflection at the boundaries, a method shown to reduce bias by Bullock [23]. In this method the excess shock amount has its sign reversed once the boundary is reached. In the given example this would result in an allele of +0.222.

Genes are selected for mutation at a rate of 20%. It should be noted that since only the floor of the allele values affects behaviour at the phenotypic level, several of the random shocks have no outwardly observable effect. For example, an allele of 3.141 receiving a positive shock of 0.5 becomes 3.641, but it still represents a threshold value of 3 for the particular scenario.

An estimate of the effect of the mutation operator can be determined by applying shocks to random alleles from the interval $[0, 14)$, and noting the difference between their old and new values after any necessary reflection. Different alleles have different profiles, since those towards the boundaries are constrained in one direction. A simulated estimate of the chosen mutation scheme on 5,000 random alleles is shown in Table 6.3.

Allele Floor Difference	**0**	**±1**	**±2**	**±3**	**±4**	**±5**
Estimated Frequency	83.9%	7.2%	5.0%	2.6%	0.9%	0.4%

Table 6.3: An estimate of the effect of Gaussian shocks with a standard deviation of two on the floor values of alleles, using a mutation rate of 20%.

Totalling the non-zero shocks we have an estimated effective mutation rate of 16.1%. It is acknowledged that this is high in comparison with standard mutation rates [41, 58], and no tests were performed using different schemes.

6.1.5 Termination

The decision to end an evolutionary run is generally dependent upon one of the three following conditions, the choice of which is left to the practitioner.

- Minimal adequate solution found
- Predetermined number of generations reached, or
- Convergence of solutions / lack of appreciable improvement

For our purposes we decided to use a termination criterion based on the number of generations. We could not easily determine a minimum adequate solution, as a specific rate of tournament wins could be high against one opponent and low against the others. Also we did not wish to rely on convergence to cause the cessation of the genetic algorithm. Through the combination of operators and parameterization discussed we were able to construct an algorithm which was still capable of exploring the strategy space after many generations.

The actual termination criterion of 50 generations used in the following experiments was chosen after several preliminary studies were undertaken. After this amount of time the rate of discovery of new global best solutions had generally slowed. Rather than continue an experiment for longer, the machine could then be made available for a new run.

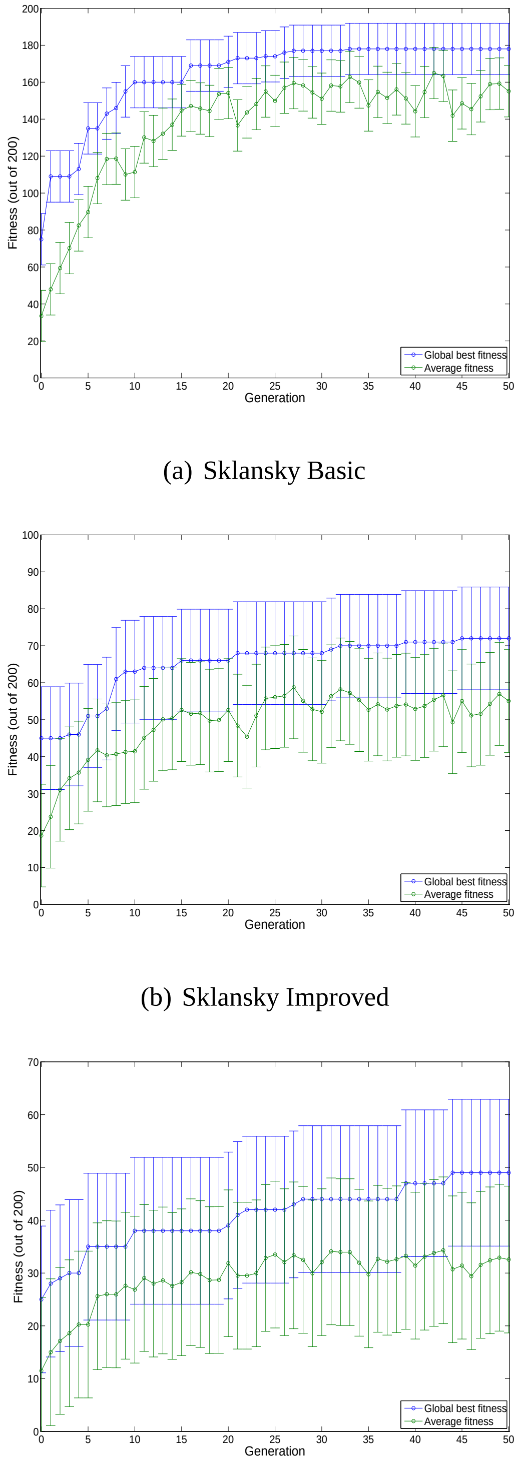
6.2 Evolutionary Algorithm Results

Given the set up and parameterization described above, we ran three EAs. The code which implemented the EA was a modification of a program developed by Graham Ritchie, a fellow research student at the University of Edinburgh. This was then linked to the Texas hold'em architecture previously used in Chapter 4. The evolutionary experiments were designed with an identical foundation to those of the exhaustive simulations in all poker-related settings. The tournament structure, opponents, and seating arrangements were all as previously described.

The fitness of each candidate was calculated as the score from 200 tournaments. All the chromosomes from every generation and their associated fitness evaluations were saved for analysis. From these figures we were able to ascertain the highest and average fitnesses per generation achieved within each run. By maintaining high waterlines on the iteration best fitnesses, we were also able to keep track of the incremental global best solutions. Upon termination of the algorithm this is the candidate which best optimizes our problem.

The average population fitness per generation, and the incremental global best solutions are shown for the evolutions against Sklansky Basic, Sklansky Improved, and Kill Phil Rookie opposition in Figures 6.1(a), 6.1(b), and 6.1(c) respectively.

The general shape of all three plots show similar trends. The starting points of the average fitness lines (i.e. the intersections with the ordinate axes) give a measure of how well a random counter-strategy performs in our experiments. The random counter-strategy averages decrease in the order of Sklansky Basic, Sklansky Improved, and Kill Phil Rookie opposition. This repeats previous statements about the comparative complexity of the adversaries.



(a) Sklansky Basic

(b) Sklansky Improved

(c) Kill Phil Rookie

Figure 6.1: Evolutions using knowledge from all four game factors against the three static opponents. Global best and average fitnesses per generation are shown with corresponding 95% confidence intervals.

It is perhaps surprising that the average fitness of random counter-strategies against Sklansky Basic players is able to score almost 17% (33 out of 200). Considering that ten players compete in each tournament a score of over 10% shows a better performance than a naive average. This demonstrates that against the space of all possible all in or fold strategies within our framework, the Sklansky Basic one is relatively weak. The similar intersections against the other two opponents are both below 10%, although only just so in the case of the Sklansky Improved opponents.

Observing the average population fitnesses given by the dashed lines we see that all are quick to increase in the first ten generations of the evolution. This is because in the early phase of our evolutionary algorithm the randomly assigned alleles get replaced with those from evolved solutions. In these formative generations the rate of finding new global best counter-strategies is at its fastest. These results are in keeping with generally observed trends in the use of EAs.

Improvement in the global best solution against Sklansky Basic opponents noticeably levels off by generation 25. The evolutions against the other two opponents were still finding new global bests approximately every five generations when the runs were terminated. The difference between the global best and average counter-strategies decreases as an evolutionary run nears convergence. In the case of Kill Phil Rookie opponents in particular it is clear that this stage was not reached. Therefore it is likely that even stronger counter-strategies would have been uncovered had the number of generations been increased.

The most remarkable aspect of these graphs is in observing the number of tournament wins achieved by the best found counter-strategies. At the termination of the run, the best solution against Sklansky Basic had won 178 out of 200 tournaments, a rate of almost 90%. This figure is a marked improvement on the comparable 212 tournament wins out of 1,000 gained using hand strength alone shown previously in Figure 4.1. The evolutions against Sklansky Improved and Kill Phil Rookie opposition also show marked improvements, with both more than tripling the top scoring counter-strategy found based on hand strength alone.

6.3 Statistical Analysis of the Evolutionary Algorithm Results

From the output generated by the three EA runs we observe that the best counter-strategies at termination have scores exceeding those found previously utilizing only a single game factor. To rigorously test this presumption we employ statistical hypothesis testing.

As in the Chapter 4, we first take the best found counter-strategies and play them off in 5,000 tournaments against their respective opponents. Again, the reason for this is that the results gain greater credence over a larger number of tournaments.

The number of tournament wins in these enlarged experiments are shown in Table 6.4.

	SB	**SI**	**KPR**
Score (out of 5,000)	4,340	1,165	786

Table 6.4: Tournament wins of best evolved counter-strategies against the three static opponents.

We now test whether the proportion of tournaments won by these evolved counter-strategies, which combine all four game factors, are statistically larger than the best of those found using a single factor through exhaustive simulation. For this we again compare the percentage of tournaments won, and therefore the Z-test for the equality of two proportions is appropriate.

From Table 4.12 we recall that the game factor whose inclusion leads to the highest scores against all three static opponents is the knowledge of one's opponents' prior bets. Therefore it is the proportion of tournament wins in these cases that we use in the tests.

The proportions we compare in the statistical tests are shown for each opponent in the columns of Table 6.5. We reiterate that all figures are calculated from a population of 5,000 samples.

Formally, our null hypotheses are that the proportion of tournament wins using all four game factors (evolution bests) are the same as those achieved using only prior bet knowledge (simulation bests), against the respective opponents.

The resulting p-values given in Table 6.6 reveal that all null hypotheses are rejected

	SB	SI	KPR
Evolution Best Percentage	86.8%	23.3%	15.7%
Simulation Best Percentage	71.2%	21.9%	11.9%

Table 6.5: Highest proportion of tournament wins found by EA and exhaustive simulation experiments against the three static opponents.

at the 95% confidence level. Indeed with the Sklansky Basic and Kill Phil Rookie opponents we have a far greater level of confidence from the tests.

SB	SI	KPR
< 0.00001	< 0.05	< 0.00001

Table 6.6: P-values from the equality of proportions tests using the values in Table 6.5.

From this we conclude that the incorporation of knowledge from all four factors has produced counter-strategies with a significantly higher win rate compared to those utilizing only a single factor. This result is important not only because it demonstrates that the amalgamated knowledge produces stronger counter-strategies, but also that these counter-strategies implicitly manage to resolve conflicting signals from the individual factors. This is one of several topics we investigate further in the following chapter.

6.4 Conclusions

In this chapter we have sought to discover strong players that are able to incorporate knowledge of seating position, opponents' prior bets, tournament level, and stack size in unison. Having previously highlighted the deficiencies in expanding the exhaustive simulation approach we applied the evolutionary computation methodology explained in Chapter 5 to our pre-flop no-limit Texas hold'em tournament domain.

It should be understood that the evaluation of a single strategy takes exactly the same amount of time within the evolutionary and exhaustive simulation settings. Whereas the exhaustive simulations simply cycle through every possible strategy by enumeration, the EA is able to sample the space more intelligently by basing its search on prior evaluations. As the strategy space grows so do the benefits to using the evolutionary approach.

We looked at the representation of strategies within our framework, and noted their similarity to the chromosomes employed within a genetic algorithm. Following the EA methodology we discussed the necessary decisions within the initialization and parameterization of the algorithm. Specifically, we selected an appropriate fitness function, selection method, crossover and mutation operators.

We then ran three EAs, one for each static opponent. On plotting the output from the evolutionary runs we saw how this technique is able to continually find stronger solutions by guiding the search into more promising areas of the fitness landscape.

The results were analysed by using an appropriate statistical hypothesis test. The evolved counter-strategies, which incorporated knowledge from all four game factors, were shown to win a larger number of tournaments than those which utilized only a single piece of game-related information. From this we concluded that the counter-strategies incorporating all four game factors achieved through use of the EA were the strongest yet found, and that they implicitly manage to resolve conflicting signals from each of the individual factors.

One of the benefits of utilizing evolutionary computation is it enables us to store information relating to the candidates in each generation. This can then be analysed to help understand what drives the search towards the best found solutions. In the following chapter we undertake such studies, and compare our findings and the resulting strategies to the non-academic poker literature.

Chapter 7

Additional Analysis and Interpretation of Results

We observed at the end of the last chapter that the evolved counter-strategies are significantly stronger than those found through exhaustive simulation. The amalgamation of all four game-related factors produces better pre-flop Texas hold'em tournament players than those discovered utilizing only a single factor.

In this chapter we seek to understand in what ways the former improve upon the latter, as we interpret the solutions in the context of poker strategy. We firstly compare the evolved counter-strategies against those from the exhaustive simulations, and then to professionals' advice taken from the non-academic poker literature.

Using data from the evolutionary runs we examine convergence in the populations and assess the relative importance of the candidates' genes in producing high scores.

The chapter concludes by questioning the necessity for a sophisticated technique to discover strong solutions, and asks whether a simple random search would suffice. The results from the random search are then used to confirm our previous findings on the relative importance of betting actions in one particular game scenario.

7.1 Recap and Comparison of the Evolutionary Algorithm and Exhaustive Simulation Results

In this section we reflect on the findings of Chapters 4 and 6. Firstly we recap the exhaustive simulation results. We saw in Table 4.10 that the hand strength only threshold leading to the best scores against all three static opponents was four. After introducing knowledge in the form of a binary variable we noted in all the four cases that the two new best thresholds diverge away from this value. Therefore if we take the threshold level for hand strength only as a reference point, we can say whether the new best found counter-strategies play looser or tighter (i.e. more or fewer hand groups) dependent upon the state of the binary variables. This information is shown in Table 7.1.

Scenario	**SB**	**SI**	**KPR**
Early position	↑	↑	↑
Late position	↓	↓	↓
No prior bet	↑	↑	↑
Prior bet	↓	↓	↓
Level ≤ 6	↓	↓	↓
Level > 6	↑	↑	↑
$M \leq 5$	↑	↑	↑
$M > 5$	↓	↓	↓

Table 7.1: The effect of the extra game information on the looseness or tightness of the best found counter-strategies against each static opponent compared to the number of hand groups played based on hand strength alone. "↑" represents more groups, and "↓" fewer groups.

There are very clear trends in these results across all opponents. The best scores are achieved by playing looser when in early position, when there has been no prior bet, in a late tournament level and when small stacked. It is similarly beneficial to play tighter in the complementary scenarios.

An interesting conundrum arises if we try to use the information from this table in the play of a hand when we can observe all four binary states. It is clear that the respective arrows will align in only two of the sixteen possible scenarios: either when all signal to play looser or all signal to play tighter than the hand strength alone value. In the other fourteen cases at least one of the factors will provide a conflicting signal to the others.

As an example, suppose we are in a situation with a large stack late in the tournament. In isolation, the first of these two factors suggests playing a low number of hand groups, whereas the second recommends playing a larger number. To make a betting decision which combines these two pieces of information requires a resolution to this incongruence.

The large scores of the best counter-strategies found in Chapter 6 show that the evolutionary algorithm mechanism is implicitly able to resolve such conflicts. In an attempt to understand how this happens, we first investigate how similar the evolved counter-strategies are to those found by exhaustive simulation.

For this analysis we first expand each best found counter-strategy from the exhaustive simulations to a chromosomal representation, in a manner previously explained in Section 6.2.2. We then treat these solutions and those from the EAs as points in a multidimensional space. To measure the "closeness" of the EA best[1] to each of the simulation bests, we take the Manhattan distance between each pair of points. This metric totals the absolute difference between each of the corresponding pairs of coordinates. The solutions that are closest together will be most similar in this sense. If it is the case that stack size, for example, dominates the other three factors against all opponents then we might expect to consistently observe a proximity between the evolved counter-strategies and those utilizing chip amount information.

The highest scoring chromosomes and their associated Manhattan distances are shown in Tables 7.2(a), 7.2(b), and 7.2(c).

The results in these tables are interesting for their dissimilarity. The best evolved solutions have varying comparabilities with the exhaustive simulation bests. With the Sklansky Basic opponents, for example, we find that the evolved counter-strategy most closely resembles that found using knowledge of the opponents' prior bets. Contrarily, with the Kill Phil Rookie counter-strategies it is the player founded on tournament level knowledge that is closest to its evolved counterpart.

There is no discernible pattern to suggest that the best evolved counter-strategies most closely resemble those resulting from the use of one particular factor. Nor is there an individual factor which is consistently distant from the best evolved counter-strategies. So although we noted consistently in Tables 4.11 and 4.12 that knowledge of prior actions and seating position led to the greatest and least improvements respectively for a single additional factor, when we allow the test player access to all four factors the significance of each one gets weighted according to the opponent.

We can gain a greater understanding of the play of each of the evolved counter-strategies by calculating the mean hand group over all states of the four binary variables. For example, to find the mean hand group played in early seating position (i.e.

[1] There were actually three slightly different counter-strategies which all achieved the same high score against Sklansky Basic opponents. We have used the first of those discovered in the analysis.

Table 7.2: Manhattan distances of evolved (EA) bests from exhaustive simulation bests (hand and position (P), opponents' prior bet (B), tournament level (L), stack size (S)) against the three static opponents.

(a) Sklansky Basic

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Dist**
EA	9	10	12	7	3	2	6	4	12	0	13	5	0	0	0	2	-
P	8	8	8	8	8	8	8	8	2	2	2	2	2	2	2	2	57
B	9	9	9	9	0	0	0	0	9	9	9	9	0	0	0	0	43
L	1	1	12	12	1	1	12	12	1	1	12	12	1	1	12	12	83
S	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	69

(b) Sklansky Improved

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Dist**
EA	6	12	13	13	7	3	9	6	1	4	8	4	5	2	5	10	-
P	13	13	13	13	13	13	13	13	2	2	2	2	2	2	2	2	60
B	13	13	13	13	3	3	3	3	13	13	13	13	3	3	3	3	68
L	1	1	11	11	1	1	11	11	1	1	11	11	1	1	11	11	60
S	12	2	12	2	12	2	12	2	12	2	12	2	12	2	12	2	82

(c) Kill Phil Rookie

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Dist**
EA	13	8	13	11	2	4	4	9	0	3	12	7	9	1	11	0	-
P	11	11	11	11	11	11	11	11	1	1	1	1	1	1	1	1	71
B	12	12	12	12	3	3	3	3	12	12	12	12	3	3	3	3	61
L	3	3	12	12	3	3	12	12	3	3	12	12	3	3	12	12	59
S	8	2	8	2	8	2	8	2	8	2	8	2	8	2	8	2	75

when P = 0 in the notation of Table 6.1), we average a chromosome over its first eight alleles.

The mean hand group values calculated in this way are presented in Table 7.3.

Scenario	**SB**	**SI**	**KPR**
Early position	6.6	8.6	8.0
Late position	4.0	4.9	5.4
No prior bet	8.5	7.6	8.4
Prior bet	2.1	5.9	5.0
Level ≤ 6	4.5	5.0	5.0
Level > 6	6.1	8.5	8.4
M ≤ 5	6.9	6.8	8.0
M > 5	3.8	6.8	5.4

Table 7.3: Average allele values in the best evolved counter-strategies against the three static opponents.

We can comprehend these figures by taking each factor in isolation and looking at the difference in thresholds with the binary variable in either state. In this way we can make a relative comparison in the style of play dependent upon the state of each individual piece of information.

Note that the players are looser on average in early seating position, with no prior bet, in late levels, and with a small stack[2]. These are exactly the same tendencies that we saw in Table 7.1. Hence, on average, the best evolved counter-strategies are also seen to respond to the state of each individual binary variable in the same manner as the exhaustive simulation bests.

We must be careful, however, not to exaggerate the usefulness of mean allele values. Averaging loses the context of how these strategies actually play in any one of the sixteen specific situations. To illustrate, the best counter-strategy found against the Sklansky Basic opponents plays an average hand group of 4.0 in late seating position. In reality the actual number of hand groups played ranges from none (genes 10, 13, 14, and 15) to all (gene 11), as was shown in Table 7.2(a).

This flexibility to use different threshold values in different situations is where

[2] With the slight exception that the averages are the same, to one decimal place, for the Sklansky Improved counter-strategy with respect to tournament level.

the evolved strategies have a clear advantage over their previously found counterparts. Each evolved player has sixteen degrees of freedom in its assignment of alleles, whereas the players found by exhaustive simulation only have two. The improved strength of the evolved counter-strategies results from this, with alleles most appropriate to specific situations against a given opponent able to be selected.

So from the parallel analysis of the best exhaustive simulation and evolved counter-strategies we have shown that there is a consistency in the relative number of hand groups played dependent upon the state of each binary variable. We have seen that, all other things being equal, the best solutions in both suites of experiments play more hands in early position, when there has been no prior bet, in a late tournament level, and when small stacked.

The results from the "distance" analysis also show that the evolutionary algorithm is able to find those solutions which best weight the importance of each game-related factor dependent upon the particular opponent. This increased adaptability leads to higher scoring players.

7.2 Comparison of the Evolutionary Results to Experts' Tournament Strategy Suggestions

In our discussion of the strategic factors cited by poker professionals in Sections 2.7 and 3.2, we encountered several recommendations for correct play in a no-limit Texas hold'em tournament. In Sections 4.2.3 to 4.2.6 we noted how the results of exhaustive simulations yielded counter-strategies which agreed with the experts' advice for three of the four factors. We also discussed several reasons why the anomalous factor of seating position should produce contrary results within our framework. Now that we have evolved even stronger counter-strategies to the three all in or fold opponents, we examine whether the tactics previously discussed are observed in our evolved players.

7.2.1 Seating Position

During our evaluation in Section 4.2.3 we commented on the fact that our pre-flop Texas hold'em tournament yields contrary results for players utilizing seating position alone to those we would expect based on the non-academic poker literature. Since these players cannot observe whether an opponent in an earlier position has already bet, a more conservative strategy is preferable.

Table 7.4 breaks down the best evolved counter-strategies according to seating position. We draw attention to the three columns headed “Diff”, which show the difference between the number of hand groups played in late compared to early position for each of the three best evolved counter-strategies.

			SB			SI			KPR		
B	**L**	**S**	**P=0**	**P=1**	**Diff**	**P=0**	**P=1**	**Diff**	**P=0**	**P=1**	**Diff**
0	0	0	9	12	3	6	1	-5	13	0	-13
0	0	1	10	0	-10	12	4	-8	8	3	-5
0	1	0	12	13	1	13	8	-5	13	12	-1
0	1	1	7	5	-2	13	4	-9	11	7	-4
1	0	0	3	0	-3	7	5	-2	2	9	7
1	0	1	2	0	-2	3	2	-1	4	1	-3
1	1	0	6	0	-6	9	5	-4	4	11	7
1	1	1	4	2	-2	6	10	4	9	0	-9

Table 7.4: The differences in hand groups played by the best evolved counter-strategies against each static opponent dependent upon seating position (P), for similar opponents’ prior bet (B), tournament level (L), and stack size (S) factors.

The predominance of negative values reaffirms that, typically, fewer hand groups are played by the best evolved counter-strategies in a late rather than early position.

It is possible for the difference values to take integer values from the interval $[-13,\ldots,13]$. If the values observed occurred with uniform probability, we would expect to see 13 / 27 = 48.1% of negative sign. Here we have 20 / 24 = 83.3%.

Using the Z-test for a proportion (binomial distribution) [69] previously explained in Section 3.4.6 we can assess the null hypothesis that the percentage of negative values observed is equal to the naive expectation given above.

The test statistic is calculated to be 3.25, which is highly significant. Therefore we reject the null hypothesis that the preponderance of negative values occurred by chance alone. There is sufficient evidence to suggest that the evolved players are tighter in late than in early position.

7.2.2 Opponents' Prior Actions

In the same manner as above we can tabulate the alleles of the three best evolved counter-strategies to highlight the differences in their play dependent upon the presence or otherwise of a prior bet. Table 7.5 shows this information for identical states of the other three factors.

P	L	S	SB			SI			KPR		
			B=0	B=1	Diff	B=0	B=1	Diff	B=0	B=1	Diff
0	0	0	9	3	-6	6	7	1	13	2	-11
0	0	1	10	2	-8	12	3	-9	8	4	-4
0	1	0	12	6	-6	13	9	-4	13	4	-9
0	1	1	7	4	-3	13	6	-7	11	9	-2
1	0	0	12	0	-12	1	5	4	0	9	9
1	0	1	0	0	0	4	2	-2	3	1	-2
1	1	0	13	0	-13	8	5	-3	12	11	-1
1	1	1	5	2	-3	4	10	6	7	0	-7

Table 7.5: The differences in hand groups played by the best evolved counter-strategies against each static opponent dependent upon opponents' prior bet (B), for similar seating position (P), tournament level (L) and stack size (S) factors.

It is again clear by observation that all three counter-strategies are tighter when there has been a prior bet: 19 of the 24 values are negative. The same hypothesis test as used above yields a test statistic of 2.84 for this proportion. The increased tightness after a bet is statistically significant, and is in keeping with the poker professionals' advice.

7.2.3 Tournament Level

In the Section 7.1 we commented on the fact that the average hand group played by the best evolved counter-strategies in the late tournament levels is greater than of the early levels against all three static opponents. This was in keeping with the best counter-strategies found in Section 4.2.5 for players utilizing tournament level knowledge in conjunction with hand strength.

The looseness of a player is a measure of how many hands he is prepared to play, whilst aggression measures the frequency of bets as opposed to calls. Within our all in or fold framework these two metrics can be treated as one and the same, since the call betting option is redundant.

We can evaluate the looseness/aggression in the evolved counter-strategies with respect to the stage of a tournament by noting the difference between the hand groups played in early and late levels. Table 7.6 orders the alleles of the best evolved counter-strategies against each opponent. The differences in the number of hand groups played dependent upon tournament level are shown in the columns headed “Diff”.

			SB			SI			KPR		
P	**B**	**S**	**L=0**	**L=1**	**Diff**	**L=0**	**L=1**	**Diff**	**L=0**	**L=1**	**Diff**
0	0	0	9	12	3	6	13	7	13	13	0
0	0	1	10	7	-3	12	13	1	8	11	3
0	1	0	3	6	3	7	9	2	2	4	2
0	1	1	2	4	2	3	6	3	4	9	5
1	0	0	12	13	1	1	8	7	0	12	12
1	0	1	0	5	5	4	4	0	3	7	4
1	1	0	0	0	0	5	5	0	9	11	2
1	1	1	0	2	2	2	10	8	1	0	-1

Table 7.6: The differences in hand groups played by the best evolved counter-strategies against each static opponent dependent upon tournament level (L), for similar seating position (P), opponents’ prior bet (B), and stack size (S) factors.

We again employ the Z-test for a proportion (binomial distribution), but this time to test the presence of the high percentage of positive values. Three quarters of the difference values are positive, and this figure is statistically significant at the 1% level. The evidence clearly suggests that the evolved players are more aggressive at the end of a tournament than at the start, in keeping with the advice in the non-academic literature.

7.2.4 Stack Size

The three highest scoring counter-strategies utilizing stack size information found using exhaustive simulations each showed a tendency to play more hands with a small stack than a large one. This observation was virtually repeated when we calculated the average hand groups played by the corresponding best evolved counter-strategies; one of the three cases showed equality.

Using the same method to those undertaken above, Table 7.7 shows the pairwise difference in the best evolved counter-strategies between those hand groups played with a small stack and a large one for otherwise identical situations.

			SB			SI			KPR		
P	B	L	S=0	S=1	Diff	S=0	S=1	Diff	S=0	S=1	Diff
0	0	0	9	10	1	6	12	6	13	8	-5
0	0	1	12	7	-5	13	13	0	13	11	-2
0	1	0	3	2	-1	7	3	-4	2	4	2
0	1	1	6	4	-2	9	6	-3	4	9	5
1	0	0	12	0	-12	1	4	3	0	3	3
1	0	1	13	5	-8	8	4	-4	12	7	-5
1	1	0	0	0	0	5	2	-3	9	1	-8
1	1	1	0	2	2	5	10	5	11	0	-11

Table 7.7: The differences in hand groups played by the best evolved counter-strategies against each static opponent dependent upon stack size (S), for similar seating position (P), opponents' prior bet (B), and tournament level (L) factors.

There is a slight majority of negative values in the difference columns, but the results are certainly not as conclusive as those shown for the other factors above. The proportion of negative values - 14 out of 24 - is not statistically significant. Therefore it appears that any strategic difference in the number of hand groups played based on stack size is less important in our experiments than those previously discussed.

7.2.5 The Gap Concept

As noted in Section 3.2.3, the Gap Concept states that the strength of hand needed to open a pot is less than that required in an otherwise identical situation if an opponent has already bet. The "gap" between the strength of the two hands is also said to increase as a tournament progresses.

We can look for the presence of the Gap Concept in the strategies of the three evolved players by comparing the number of hand groups played dependent upon both the opponents' prior actions and tournament level. If we compare situations of identical seating position and stack size factors we would expect the Gap Concept to dictate firstly that fewer hands are played after a prior bet than not, and also that this difference in the number of hand groups played with or without a prior bet is larger in late levels compared to early ones.

The results on the effect of a prior bet seen previously show that the first of these two conditions is met. Table 7.8 tabulates the alleles of the best evolved counter-strategies to assess whether the size of the gap increases with tournament progression.

	SB		**SI**		**KPR**	
	L=0	**L=1**	**L=0**	**L=1**	**L=0**	**L=1**
P=0, S=0						
B=0	9	12	6	13	13	13
B=1	3	6	7	9	2	4
Diff	-6	-6	1	-4	-11	-9
P=0, S=1						
B=0	10	7	12	13	8	11
B=1	2	4	3	6	4	9
Diff	-8	-3	-9	-7	-4	-2
P=1, S=0						
B=0	12	13	1	8	0	12
B=1	0	0	5	5	9	11
Diff	-12	-13	4	-3	9	-1
P=1, S=1						
B=0	0	5	4	4	3	7
B=1	0	2	2	10	1	0
Diff	0	-3	-2	6	-2	-7

Table 7.8: Seeking the presence of the Gap Concept. The differences in hand groups played by the best evolved counter-strategies against each static opponent dependent upon opponents' prior bet (B) and tournament level (L), for similar seating position (P), and stack size (S) factors.

The rows headed "Diff" show the same values previously seen in the prior bet analysis. To investigate whether the gap widens between early and late levels we compare the difference values between early (L=0) and late (L=1) levels, for each otherwise identical situation.

Of the twelve pairs of values the gap grows in six cases, whilst in five it contracts. Therefore from these results we find no significant evidence of the presence of the Gap Concept in its entirety. Whilst we do observe fewer hands being played after a bet, our results do not validate the assertion that the gap widens later in tournaments.

7.3 Convergence in the Evolutionary Algorithms

In the foregoing discussion on the make-up of the three best evolved counter-strategies we have looked at general strategic considerations regarding the four factors employed. We now seek to understand whether a player's actions in different situations have the same consequence on their performance. We first investigate this by analysing generational data from the evolutionary runs.

The initialization and parameterization of the EA described in Chapter 6 is in part designed to maintain diversity in the gene pool. The diversity should at least be continued for a number of generations sufficient to find suitable solutions. It is the nature of EAs, though, to assign a higher probability of survival to the strongest candidates. Over time the genes from these individuals will start to swamp the population, to such an extent that all candidate solutions start to resemble one another. This process is known as convergence.

We are able to examine convergence within our evolutionary runs since we have stored information on the individuals in each generation. This is one of the benefits of using EAs as problem solvers. Not only can we find a solution, but we can also gain an insight into how it is derived.

Here we wish to observe whether the rate of convergence is similar across all genes, or whether some are more inclined to home in on specific alleles more rapidly. Since the algorithm is designed to promote strong solutions we can infer that genes with a large convergence measure are in some way important contributors to the high scores achieved.

To analyse the convergence we retrieve the population data from each run's final generation[3]. From these we calculate the standard deviation of the alleles at each gene. A low standard deviation reflects a large degree of convergence, and vice versa.

[3] Analysis not presented here was performed which showed similar degrees of convergence within each population from around generation 20 onwards.

Table 7.9 gives the standard deviations of allele value in the best found counter-strategies to the three static opponents at generation 50.

(a)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
SB	8.6	8.8	10.9	7.0	2.0	2.4	5.5	3.9
SI	5.7	11.1	11.8	12.1	6.6	2.9	8.1	6.5
KPR	11.1	7.1	11.7	10.6	1.7	4.1	3.6	9.2

(b)

	9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
SB	11.5	1.3	12.2	4.4	0.8	0.3	1.8	2.3
SI	3.9	3.2	8.1	4.4	3.5	1.6	5.2	10.1
KPR	1.2	2.7	10.4	7.1	8.8	0.8	10.9	0.9

Table 7.9: Standard deviation of alleles in the EA final populations against the three static opponents.

Before commenting on these results we must remember that mutations from the previous generation can potentially skew some of the values calculated. Certain allele values exist within the population not because they have been found to be strong, but simply through the chance effect of mutation. However, we should also remember that that the average population fitnesses are relatively high by generation 50, in spite of any such effects.

From the table it is evident that certain genes exhibit convergence to a greater extent than others. So in the case of Sklansky Basic opposition, for example, we find that gene 10 shows a higher degree of convergence than its immediate neighbours. It is also clear that some genes have converged at different rates across the three opponents. Interestingly, though, gene 14 shows a large amount of convergence across all three opponents.

7.4 The Relative Importance of Genes

Of all the sixteen possible genes, that numbered 14 has a consistently low standard deviation of allele value in the final populations of all three evolutionary runs. With reference to Table 6.1, the scenario which this gene represents is given in Table 7.10, along with the corresponding alleles in the best evolved counter-strategies.

Gene	Convergence	P	B	L	S	SB	SI	KPR
14	High	Late	Bet	Early	Large	0	2	1

Table 7.10: The counter-strategies represented by gene 14, dependent upon seating position (P), opponents' prior bets (B), tournament level (L), and stack size (S). The three right hand columns show the alleles for each best evolved counter-strategies to the three static opponents.

This table shows that in a late seating position, after a prior bet, early in a tournament, and with a large stack, the evolutionary runs have centred on playing a very low number of hand groups.

To assess the contribution of this and the other genes on a strategy's resulting fitness, we perform a random search. This employs a uniform random sampling from the $[0, 14)$ range for each allele within a chromosome. We employ this method since it covers a wider range of the search space than the directed search of our EA. Taking individuals from all points within the space avoids biasing the sampling.

The random search samples 1,000 individuals and evaluates each over 200 tournaments. Before continuing with the gene analysis, we first assess whether the random search finds solutions as comparably strong as the EA. This is interesting as, if it does, it undermines the necessity for a sophisticated search. The random search compares with the EA of the previous chapter in the number of individuals sampled. An EA utilizing 20 individuals over 50 generations also samples 1,000 candidate solutions[4].

After the 1,000 samples are evaluated, the best counter-strategies discovered for each of the three opponents are played off over 5,000 tournaments for statistical testing. The proportion of tournaments won are shown in Table 7.11. For comparison purposes this table also includes the win proportions previously presented in Table 6.5 for the corresponding best counter-strategies found through the evolutionary method.

Note that in all cases the evolved counter-strategies score a higher proportion of tournament wins than their random search counterparts. This result emphasizes the difficulty in locating strong solutions within the fitness landscape without using a guided search, and stresses the successful realization of this task by evolutionary means.

To give statistical significance to the above results we use the Z-test for the com-

[4] However, it should be noted that not all of these samples are unique. Elitism means that two individuals pass through successive generations unaltered. Also, it is possible through the effects of crossover and mutation for an offspring solution to exactly match a previously sampled individual.

	SB	**SI**	**KPR**
Random Search	73.6%	11.9%	6.3%
Evolution	86.8%	23.3%	15.7%

Table 7.11: Highest proportion of tournament wins found by random search and evolutionary experiments against the three static opponents.

parison of proportions explained previously. Here we compare the percentage of tournaments won by the best counter-strategies found through random search to those produced by the EA. The calculated p-values are given in Table 7.12.

	SB	**SI**	**KPR**
Random vs Evolutionary	< 0.0001	< 0.0001	< 0.0001

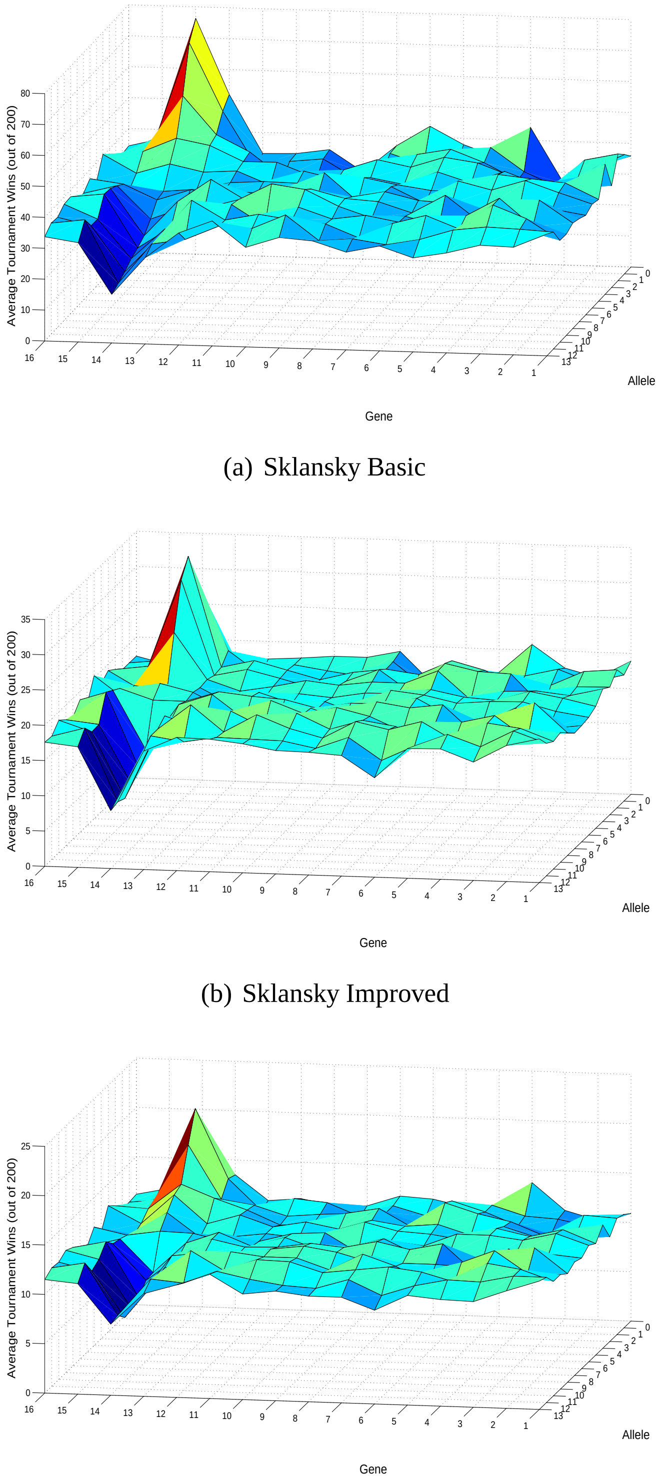
Table 7.12: P-values for the proportion comparison test between the best solutions found through random search and those of the EA against the three static opponents.

Given these values we reject the null hypotheses that the random search players score the same proportion of tournament wins as the evolved ones. The EA proves to be superior to a purely random search in discovering stronger players.

Returning to the assessment of the importance of genes, we now use the 1,000 randomly selected candidates per opponent and their associated fitnesses. We calculate the mean fitness for the strategies conditioned on every possible allele. Hence we average over all individuals with allele x in gene y, with $x \in \{0, 1, \ldots 13\}, y \in \{1, 2, \ldots 16\}$. The resulting plots for each of the three static opponents are shown in Figure 7.1.

The trends within each graph are striking. In all three cases we note that the average fitnesses of the random candidates are highly conditional upon the allele at gene 14. Against all opponents a low allele produces high average fitnesses, whilst a high allele results in low average fitnesses. It is also interesting to note that the three plots are otherwise relatively flat for all other alleles of all other genes.

This shows that a strategy's play in late position, after a prior bet, in an early tournament level, and with a large stack, has a profound effect on its expected performance. Playing a very tight game in such a situation is rewarded by a greater expected number of tournament wins, whilst being too loose has the opposite effect. This tactic is very simple to understand. When the prize money is far away and a player has a sufficiently



(a) Sklansky Basic

(b) Sklansky Improved

(c) Kill Phil Rookie

Figure 7.1: Average number of tournament wins by the randomly sampled counter-strategies, conditioned on each gene's allele.

large stack, there is no benefit to risking tournament elimination by betting after an opponent has entered the pot. As Ciaffone [31] puts it:

> The fold is an important weapon; don't be afraid to use it.

As the preceding convergence analysis shows, the EA clearly picks up on the importance of this situation. By ensuring that strategies within the population cluster around low values in gene 14, exploration can continue on the other alleles towards the goal of finding the global maximum.

7.5 Conclusions

In this chapter we have given further thought to the best counter-strategies found by the exhaustive simulation and evolutionary experiments. We have shown that there are common trends in the play of the best solutions dependent upon the states of the binary variables incorporating game-related information.

Through an analysis of the difference between the two sets of counter-strategies, we showed that there is no discernible "closeness" between the evolved counter-strategies and those of based on one particular game factor. Different tactics succeed against different opponents.

We returned to the poker professionals' recommendations for tournament poker strategy, and noted that the four factors have identical manifestations in the evolved players as we saw in those discovered by exhaustive simulations. We also observed that the effects of stack size were less pronounced, and that the Gap Concept was not totally discernible.

Convergence within the evolutionary runs was used to investigate whether the alleles on some genes are of greater importance than others. We observed that gene 14 in particular undergoes a high degree of convergence across all three static opponents, and that this gene has a major effect on the number of tournaments won within our framework. As part of this evaluation we noted that the counter-strategies found by our EA are statistically stronger than those found using a purely random search.

Chapter 8

Further Work

The preceding results and analysis validate the use of evolutionary algorithms for research into tournament poker strategies. Given that our investigations have been successful a plethora of potential extensions - both within and to the framework - are apparent. In this chapter we discuss the range of possibilities that this thesis opens up to future researchers.

8.1 Extending and Enhancing the Available Knowledge

We have seen that the inclusion of game-related information in the form of binary variables leads to tournament poker players which have better performance than those based on hand strength knowledge alone. The extra game factors employed have all had a statistically significant effect on the number of wins achieved, even though the resolution on the information contained was extremely low. Increasing the definition in this information is the simplest enhancement to the work undertaken.

Recall that with all of the four factors investigated, a player's betting can only be influenced by knowing whether each piece of information is in one of two states. We saw that the categorization of knowledge is limited to the following cases:

- Seating position: The player is in an early/late position with respect to the dealer.
- Opponents' prior actions: There are no bets/at least one bet prior to the player's action
- Tournament level: The decision is made early/late in a tournament
- Chip stack amount: The player has few/many chips

All of these criteria can be further expanded to increase the resolution on the information available. This augmentation would also proliferate the number of possible strategies available to the player, but we can now vouch for the evolutionary approach in searching the space.

Potential extensions to the information relating to seating position, tournament level, and chip stack size are obvious. Both seating position and the stage of the tournament could be modelled more accurately than the "early/late" scheme employed here. To increase the size of the strategy space gradually it may initially be desirable to move from a binary to a tertiary variable in these two cases. Simply adding a "middle" category with the appropriate reclassification is a sensible first step. Following on from this it would be possible to use each individual seating position and level as separate states, thus giving our test player full and complete knowledge of these two factors.

Chip stack amounts could also be reclassified. Again, a recommendable preparatory development would be to use a classification scheme of "small", "medium", and "large". To make the adjustment from the two-state case we could again take Harrington's M and use, for example, the gradations: $M < 3$, $3 \leq M < 7$, and $M \geq 7$.

Potentially the most interesting extension to the information available to the test players, though, is in a re-categorization of the opponents' prior actions. We have seen from the previous results that this is the single most important factor to consider, given that it has the greatest effect on the number of tournaments won. The current procedure is to act based solely upon whether at least one opponent has bet or not prior to our test player's action. Refinements to this suggest themselves immediately.

Within our all in or fold Texas hold'em variant it is apparent that a bet by anyone other than the chip leader puts at risk their tournament survival. We have seen a tendency, both in the opponents' strategies and our own, to only bet very strong hands once another has entered the pot. Therefore a player who acts after two people have moved all in has an even smaller chance of winning the pot. Not only is he up against more than one opponent, he is even more likely to be facing a better hand. The current framework treats any number of prior bets as similar cases. A scheme which incorporates the number of players to have bet into the pot should therefore increase our test player's ability to make good decisions.

Another consideration that would benefit our player is to know what the size of any prior bet is. At present our test player only knows *if* an opponent has bet, and not how much. The extra information is vital for at least two reasons. Firstly, it could be that knowing how many chips the opponent holds signifies the range of hands they are playing. The experiments incorporating stack size generated counter-strategies that played a greater range of hands if low on chips. If our test player has a mid-range hand they may not wish to fold if they believe their opponent only acted due to imminent elimination. Secondly, the present structure does not allow our test player to differentiate between the size of prior bets. This means that a bet of $1 is equivalent to one of $1,000. Comprehending the bet size as both an absolute amount and relative to one's own stack would likely be profitable.

The hand group classification used within the experiments was decidedly coarse. We took all 169 possible starting hands and classified them according to a simple segregation of thirteen hands into thirteen groups. We selected this number of groups arbitrarily. We could have chosen 169 groups, allocating one hand to each, but by choosing a smaller number we essentially blinkered our player into only recognizing thirteen distinct hands. The benefit of this simplification is that it reduces the size of the strategy space, and it made our investigations more tractable.

Even if we wished to maintain the same number of groups we could further improve the ability of test players by a better classification of the hands. It is well known by poker players and theoreticians that there is a wide dispersion in the potential profitability of different starting hands. Moreover, some hands in particular have a much greater profitability than others. The grouping scheme we used completely ignored this fact. Referring to Appendix A, we note that each starting hand in the Sklansky-Chubukov rankings has an associated X-value. The higher this value the better the hand. In our creation of thirteen groups of thirteen we ignored the information contained in these figures, and so we have hands of greatly varying profitability grouped together.

One way to resolve this issue whilst maintaining the same number of groups is to employ K-means clustering [18] on the X-values. This algorithm initially allocates each hand to one of k random sets, and then iteratively reassigns them such that the intra-cluster variance is minimized. An implementation of the algorithm was performed using $k = 13$, with the results shown in Table 8.1.

The group assignments differ greatly from those of our naive classification given in Table 4.3. We find that there are a greater number of smaller sized groups for the best hands, and a small number of large groups for those hands with low profit potential. The first new group consists solely of a pair of aces, whilst the last, conversely, contains fifty hands.

The use of a more realistic classification such as that proposed would enable the

Group	Starting Hands	#
1	AA	6
2	KK, AKs, QQ	16
3	AKo, JJ, AQs, TT	28
4	AQo, 99, AJs, 88, ATs, AJo, 77	50
5	66, ATo, A9s, 55, A8s, KQs, 44, A9o, A7s	58
6	KJs, A5s, A8o, A6s, A4s, 33, KTs, A7o, A3s, KQo, A2s, A5o, A6o	94
7	A4o, KJo, QJs, A3o, 22, K9s, A2o, KTo, QTs	78
8	K8s, K7s, JTs, K9o, K6s	28
9	QJo, Q9s, K5s, K8o, K4s, QTo, K7o, K3s	64
10	K2s, Q8s, K6o, J9s, K5o, Q9o, JTo, K4o, Q7s, T9s, Q6s, K3o, J8s, Q5s, K2o, Q8o	128
11	Q4s, J9o, Q3s, T8s, J7s, Q7o, Q2s, Q6o, 98s, Q5o, J8o, T9o, J6s, T7s, J5s, Q4o	120
12	J4s, J7o, Q3o, 97s, T8o, J3s, T6s, Q2o, J2s, 87s, J6o, 98o, T7o, 96s, J5o, T5s, T4s, 86s, J4o, T6o, 97o, T3s, 76s, 95s, J3o, T2s, 87o, 85s	216
13	96o, T5o, J2o, 75s, 94s, T4o, 65s, 86o, 93s, 84s, 95o, T3o, 76o, 92s, 74s, 54s, T2o, 85o, 64s, 83s, 94o, 75o, 82s, 73s, 93o, 65o, 53s, 63s, 84o, 92o, 43s, 74o, 72s, 54o, 64o, 52s, 62s, 83o, 42s, 82o, 73o, 53o, 63o, 32s, 43o, 72o, 52o, 62o, 42o, 32o	440

Table 8.1: The 13 groups of the 169 possible starting hands after K-means clustering, and the number of unique starting hands contained within each group (cf. Table 4.3).

test player to better discriminate between hands. An added bonus of the K-means approach is that the number of groups, and therefore the size of the strategy space, could potentially be reduced. Even in using a slightly smaller number of groups the similarity of starting hand potential within each class will be better than the naive approach.

Deceptive plays such as bluffing could potentially be included, with an alteration to the simplistic single threshold value used. A bluff is a bet with a hand that would not expect to win in a showdown. Therefore within our framework a bluff is a bet with a hand in a high numbered group.

Since we use only one threshold value to guide our players, we divide their possible actions into just two intervals. A player moves all in with hand groups numbered less than or equal to the threshold, and folds otherwise. However, we could potentially include a second threshold value greater than the first *above which* the player bets. Recall from our discussion in Section 2.4 that the game theoretic optimal strategy for the first player in von Neumann's poker [120] is to bet with both his very best *and* very worst hands. The use of two thresholds would allow for a similar scheme.

An example based solely on hand strength is illustrated in Table 8.2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Single threshold	B	B	B	B	F	F	F	F	F	F	F	F	F
Dual threshold	B	B	B	B	F	F	F	F	F	F	F	B	B

Table 8.2: Incorporating bluffing into strategies by the use of a second threshold. "B" and "F" refer to all in bets and folds respectively.

The single threshold strategy moves all in with hands in the top four groups, and folds otherwise. By incorporating a second threshold this player would also be able to move all in with hands in the bottom two groups. Therefore by employing two threshold values per scenario, we would be able to examine the effects of deceptive play and how its use alters dependent upon game-related factors.

It should finally be mentioned that the framework we have constructed is also amenable to the inclusion of additional dimensions. One such possibility involves incorporating opponent modelling into the structure. A very simple scheme would be to classify opponents as "loose" or "tight", depending upon observed tendencies in their betting frequencies. Maintaining separate thresholds for the two different classes of opponent might further improve tournament performance. This dual state classification could be further extended in a manner similar to those described for the other factors, allowing for better discrimination between the range of possible opponents.

8.2 Competing Against a Range of More Realistic Opponents

The experiments we performed in this body of work employed the use of three fairly basic strategies taken from the non-academic poker literature. Additionally, we only examined cases where our test players were seated at tables completely comprised of opponents from one of the three types. The use of a mixture of stronger, more realistic opposition would benefit the research into finding universally strong strategies.

Throughout this research we have shown by experiments on opponents of increasing complexity that several pieces of game-related information affect the counter-strategies of our test players in exactly the same way. That is, depending upon the state of a binary variable we have seen our test players play correspondingly more or fewer hands, regardless of their opponent. Whilst this gives us positive results on the importance and application of the game factors, we have stopped short of constructing an all-encompassing tournament poker strategy that would compete well against any style or mix of opponents. The reason for this is two-fold. Firstly, our remit was not to produce a strong tournament strategy *per se*, but to identify the elements within one and compare our findings to the suggestions of poker authors. Secondly, we recognize that a very important element of a complete strategy for poker is opponent modelling. Hence a truly “universal” tournament poker strategy should, at some level, incorporate a facet that we have chosen to neglect for reasons previously explained.

An important extension of this research, therefore, would be to compete our test players against mixtures of different opposing styles. Now that we have validated some of the elements of a successful tournament poker strategy we could look to forge strategies that are able to compete well against a multitude of opponents. This could start by using the simple addition of a binary variable which categorizes the style of one’s opponents as loose or tight as discussed above, then subsequently incorporate increased resolution with more elaborate statistical measures like those employed in research on ring game poker [39, 65, 116].

8.3 Changes to the Tournament Structure

Additional experiments could focus on making changes to the tournament structure employed. The set-up that we chose in our experiments is one of many possibilities. Obvious components to change are the number of entrants, payoff structure, hands per level, and players per table.

The quest to program a potentially world championship winning poker player will eventually have to tackle the issue of multi-table play. To explain, a tournament can potentially have an unlimited number of entrants, with all seated at tables of approximately ten players. As competitors are knocked out, the remaining tables are consolidated such that empty seats are filled and the total number of tables in play reduces. This procedure culminates in a single table. Our experiments have solely focused on play at the final table.

In all prior stages of a real tournament players may have to base their betting decisions on factors away from their own table. For example, a player who has the largest stack on his table may in fact be small stacked in comparison to all competitors. Our framework could be expanded to include this complete knowledge to develop successful strategies for multi-table play.

Similarly changes to the number of players per table and hands per level is likely to have an effect on best play. Experiments could be performed to illustrate the strategic consequences of a change in either of these tournament components, keeping all other constituents unchanged.

Payoff structure is one aspect of our experimental design which is uncommon in real tournaments. It is far more usual for a ten-player competition to use the percentage payoffs of 50%, 30%, and 20% for the top three finishers. It would be an interesting exercise to examine the effect of a change from our winner-takes-all approach, and see how the resulting strategies differ. These experiments could be easily performed within our evolutionary framework, with the only change necessary being to the fitness function.

8.4 Removal of the All In or Fold Betting Restriction

In the attempt to investigate tournament poker strategies that are able to compete against humans at the highest level, there are clearly two features of our experimental investigations that need to be improved. We have already alluded to the necessity for opponent modelling. The other discrepancy between our analysis and real poker is our imposition of the all in or fold betting restriction.

Recall that we introduced the binary betting option to vastly reduce the size of the strategy space. In making the problem of understanding tournament poker strategy more tractable we negated the use of the final three betting rounds, and lost the bet sizing finesse found in the real game. The most challenging augmentation to our experimental scheme would be to remove this limitation and research the strategic elements found in a true Texas hold'em tournament.

Two ramifications of using all in or fold are that we do not have to consider how much to bet, nor do we face the three post-flop betting rounds. The use of genuine Texas hold'em in the experiments would require significant amplifications to the strategy representations. Not only would our test player have to be able to respond to different sized bet amounts, it would also have to be able to act in kind. Given that there is a continuum of bet sizes possible in no-limit play, the two-way consideration and implementation of this factor presents significant problems.

A similarly tricky problem arises if we wish to embrace post-flop play. The difficulties arise in several forms. We first have the increase in strategy space due to the partial hand effects of community cards. By this we mean that whereas in the preflop game there are 1,326 unique hands to consider, the inclusion of post-flop play increases this number dramatically. In selecting five cards from fifty-two there are a total of 2,598,960 possible poker hands.

Opponent modelling would become far more awkward with the inclusion of multiple betting rounds. The first issue concerns maintaining an opponent model based on how they play the individual betting rounds. For this there would probably have to be separate models for each round and each opponent. In addition to this is the challenge of linking actions - both for the opponent and oneself - across the betting rounds within a hand. So, for example, when humans play the final betting round of a poker hand they are able to use knowledge of the play up to that point, and how their opponent has played similar situations in the past. Connecting information across these two bridges could prove to be particularly troublesome.

8.5 Summary

This chapter has illustrated some of the many possible future adjuncts to this piece of research. We have shown that our framework could be extended to further refine and understand the results of the previous chapters. By increasing the number of different possible states for the game factors we have seen how more detailed strategies could be discovered in the enlarged strategy space. We have also mentioned how a reclassification of the hand groups could prove to be beneficial in finding better performing strategies whilst keeping the size of the search space constant.

Following this we remarked upon the possibility of playing tournaments against a wider range of more complex opponents. This would be necessary to further confirm our findings, as it is important to discover whether counter-examples to the recommended strategies can be contrived.

To give the experiments a slightly more realistic edge, we have talked about how changes in the payoff structure could easily be implemented through an alteration to the evolutionary algorithm's fitness function. We also discussed how the move towards larger tournaments with a greater number of competitors requires additions to our framework.

The removal of the all in or fold betting restriction is the most apposite direction for future researchers aiming to develop an authentic Texas hold'em tournament program. We mentioned how the lifting of this limitation would impact the construction of strategies, and it remains an open question whether the framework we have fashioned would be sufficient to handle the consequential explosion in the size of the strategy space. An EA could be used and would clearly be preferable to an exhaustive approach, but run-times on the enlarged problem may still be too prohibitive with current computing technology.

Chapter 9

Summary and Conclusions

Much investigation into computer games players has been undertaken, but the techniques which apply to games of perfect information are not applicable to poker. Modern efforts have tackled the game with a variety of techniques, but all have focused on the ring game format with limit betting. The most commonly played poker variant, Texas hold'em, is the subject of increasing amounts of research, but this thesis is the first study on the game using no-limit betting within a tournament structure.

Many professional poker players have authored books detailing their recommendations for good poker strategy in the non-academic literature. However, their assertions are based primarily on experience and lack a scientific basis. By creating a framework utilizing a slightly reduced form of Texas hold'em, we have been able to isolate and test the most important messages in their writings.

The all in or fold form of Texas hold'em used simplifies the task of investigating different poker strategies. We saw that a program encoded with one such system fared relatively well in an organized poker tournament. This and two similar all in or fold strategies were then employed as opponents to test different strategies against.

The first suite of experiments sought first to determine how well a strategy based solely on a player's hand strength fared against the different opponents. These results then formed a baseline from which to make comparisons. We next incorporated items of game-related information into our test player's knowledge domain. We showed by exhaustive simulations that utilizing factors relating to seating position, opponents' prior actions, tournament level, and stack size all had a statistically significant effect on the test player's strength.

In wishing to combine all four game factors, we highlighted the difficulties of extending the exhaustive simulation approach, and turned instead to evolutionary algorithms. These were seen to have been successfully applied to a number of game players, including some for poker. We also showed that an EA was able to find mathematically optimal strategies in a toy poker variant.

Returning to our all in or fold pre-flop Texas hold'em test bed, we discussed the issues of EA implementation before performing evolutionary runs on players which could act based on all four game factors. The resulting strategies were then shown to score a significantly larger number of tournament wins than any previously found, both confirming that the amalgamation of knowledge is beneficial in forming stronger strategies and that an EA is able to resolve conflicts between contradictory inputs.

In a further analysis of our evolved players we compared their strategies with the poker professionals' guidance. We found evidence of the presence of some of these effects, but not others. By examining convergence within the evolutionary runs we noted that some genes seemed to converge particularly rapidly. One gene in particular was further noted to have a large effect on the resulting number of tournament wins.

We concluded our study with a discussion on the ways in which this research could be extended by future researchers in the field of computer poker. We noted that several augmentations present themselves immediately, and that supplementary efforts into such topics as bluffing could be attempted to advance this work.

The challenge of developing a tournament poker program can be divided into two parts. The first task is to gain an understanding of general tournament strategy, whilst the second is to be able to adapt to specific opponents in real time.

This research has addressed the first of these issues. All of our experiments have been performed offline; we have developed players that are able to counter particular opponents after several iterations of simulation. Since the solutions found are opponent-specific, we have been careful to term them "counter-strategies" and have avoided making claims about their potential strength against different competitors. However, we have seen that certain strategic trends, such as playing a wider range of hands later in a tournament, are common across all adversaries. So whilst we cannot go as far as to claim that the high-level strategies discussed are the correct way to play against any opponent, our observation of similar tactics leading to the highest scores against three increasingly complex adversaries does lend weight to this argument.

Taking this work forward will require further investigation of the strategic effects found within tournaments. Knowing exactly when and how to adjust for different blind and payoff structures, chip stack sizes, and opponents will require a large amount of future effort. Whilst these researches may be undertaken with a variety of computational intelligence techniques, we have shown that evolutionary algorithms can be used to positive effect in this regard.

This thesis contributes to the growing interest in computer poker by taking the first scientific steps towards understanding tournament poker strategy. In so doing we have been able to empirically validate many of the factors cited by professional poker players as being important considerations in a player's decision making. The stringent use of hypothesis testing for this purpose is novel, and its results substantiate the authors' claims.

The ultimate test of any poker player - human or computer - is to adapt to previously unseen opponents in real time. Once the elements of strong tournament strategy have been refined it is interesting to ponder how these will be combined with opponent modelling to create strong players. It can be argued that flexible opponent modelling is the most difficult problem to overcome in the development of a world championship winning program. This aspect would be hard enough to implement for many games, but the pervasive deception found in poker makes the issue even more difficult.

Given the progress made in developing limit ring game Texas hold'em computer programs and opponent modelling by the GAMES Group, it may well be that directly converting their players will be the quickest route to success in no-limit tournament play. Incremental additions and the honing of tournament-directed features in both general strategy and opponent modelling could well prove triumphant. This thesis has been the first attempt to bridge the gap between limit ring game and no-limit tournament play, but we recognize that many further studies are required to refine the ideas contained herein.

The day when a computer poker program wins the world championship may be some time away, but continued investigations towards this goal will not only make it inevitable, but will further our understanding of the more general topic of reasoning under conditions of uncertainty.

Appendix A

Texas Hold'em Starting Hands

Given that card ordering is unimportant and that it employs the standard 52-card deck there are

$$^{52}_{\ 2}C = 1,326$$

unique starting hands for a player in Texas hold'em.

This figure can further be reduced due to the equivalence of suits. So for example, the hand $A\clubsuit 8\diamondsuit$ has exactly the same value as $A\spadesuit 8\heartsuit$. One must take care, however, to separate suited and offsuit hands. The hand $A\heartsuit 8\heartsuit$ should not be treated as commensurate with the two aforementioned, since both of its cards are of the same suit. This differentiation is important due to the latter hand's increased *flush*-making potential. A flush is a complete poker hand comprising five cards of the same suit.

One way to visualize the number of different starting hands, accounting for suit equivalence, is shown in Table A.1.

Since there are 13 distinct card ranks, there are

$$13^2 = 169$$

different starting hands within this matrix. The cells along the top left to bottom right diagonal represent the paired cards. Without loss of generality we can allocate suited hands to the cells above this line, and offsuit hands to those below. Returning to our example above, therefore, both $A\clubsuit 8\diamondsuit$ and $A\spadesuit 8\heartsuit$ occupy the cell $(A,8)$, whereas $A\heartsuit 8\heartsuit$ occupies $(8,A)$.

The entries within each cell show the number of unique starting hands of a particular two-card combination, taking account of suit equivalence. Starting with the pairs,

	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**T**	**J**	**Q**	**K**	**A**
2	6	4	4	4	4	4	4	4	4	4	4	4	4
3	12	6	4	4	4	4	4	4	4	4	4	4	4
4	12	12	6	4	4	4	4	4	4	4	4	4	4
5	12	12	12	6	4	4	4	4	4	4	4	4	4
6	12	12	12	12	6	4	4	4	4	4	4	4	4
7	12	12	12	12	12	6	4	4	4	4	4	4	4
8	12	12	12	12	12	12	6	4	4	4	4	4	4
9	12	12	12	12	12	12	12	6	4	4	4	4	4
T	12	12	12	12	12	12	12	12	6	4	4	4	4
J	12	12	12	12	12	12	12	12	12	6	4	4	4
Q	12	12	12	12	12	12	12	12	12	12	6	4	4
K	12	12	12	12	12	12	12	12	12	12	12	6	4
A	12	12	12	12	12	12	12	12	12	12	12	12	6

Table A.1: A matrix illustrating the reduction of the 1,326 unique two-card starting hands in Texas hold'em to 169 different groups. The row and column headers each signify the rank of a single card. Suited hands are allocated above the top left to bottom right diagonal, with offsuit hands below. Entries within the matrix show the number of unique hands within each group.

there are ${}^4_2C = 6$ ways of choosing two cards of the same rank from four suits. For non-pair hands there are $4^2 = 16$ combinations, of which 4 are suited and the remaining 12 offsuit.

In total, therefore, there are 78 pairs, 312 suited and 936 offsuit hands. These figures sum to 1,326.

Appendix B

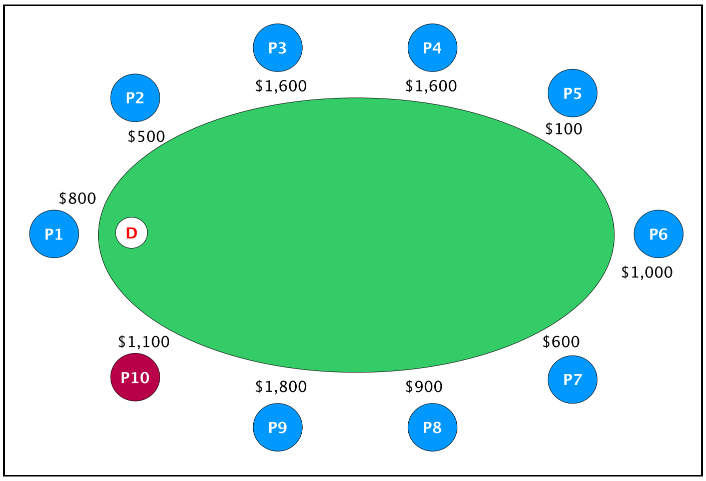
Texas Hold’em Example and Five-Card Hand Rankings

This appendix first details a step-by-step example of the play of a hand within a no-limit Texas hold’em tournament. We then give the five-card hand rankings used in determining the winner of a showdown.

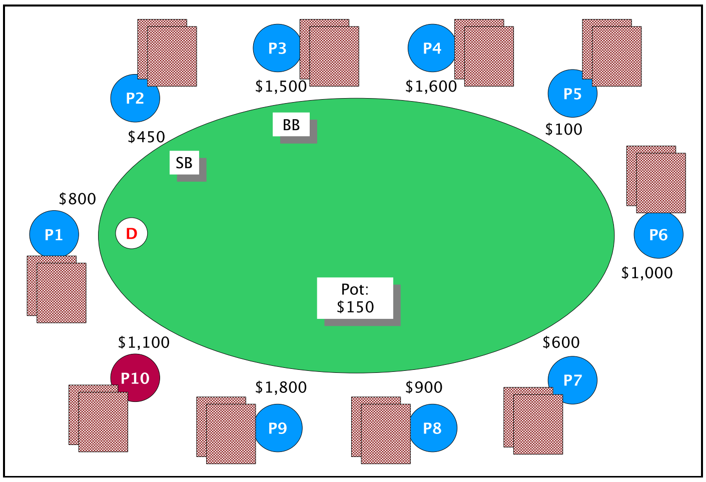
B.1 Example Hand

This example is included to illustrate the explanation of Texas hold’em given in Section 2.2.

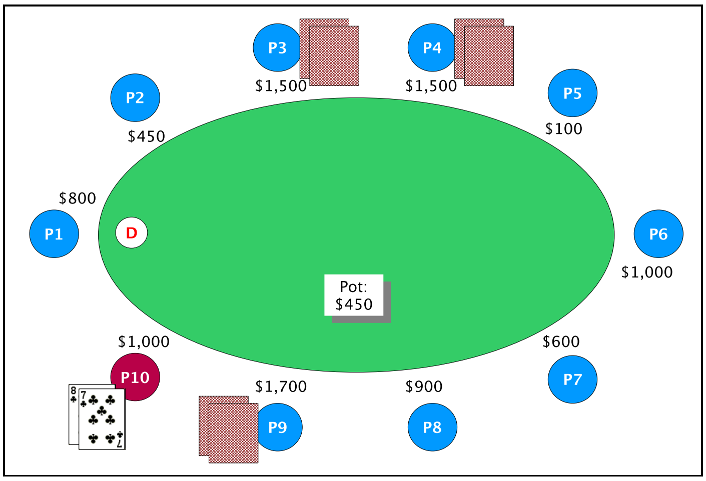
Ten players are competing in a single table tournament and each started with $1,000 in chips. Our player is labelled “P10”, with her opponents labelled “P1” up to “P9”. The players’ seating positions and present chip stacks are:



The dealer for this hand is Player 1, shown by the *dealer button* on the table in front of him. The two players to the dealer's left post the small blind (SB) and big blind (BB) respectively, which for this hand are \$50 and \$100. Thus the pot starts at \$150. All players now receive their hole cards. The game state after the deal is:

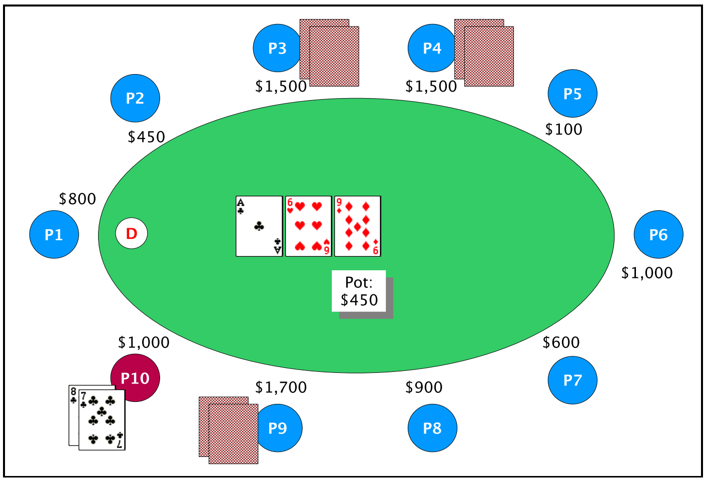


First the pre-flop round of betting takes place. Starting to the left of the big blind with Player 4, the competitors must in turn either call the existing bet amount, raise it, or fold. Player 4 chooses to call the \$100 big blind. The next four players fold, and take no further part in this hand. Player 9 then calls for \$100. Having been dealt 8♣7♣ our player also calls. The dealer and small blind fold, and the big blind checks. This he can do since he has already entered \$100 into the pot with his forced bet. Now, since all active players have paid exactly the same amount into the pot, the betting round is over with the pot standing at \$450. The following diagram shows the hand after the pre-flop betting round[1].

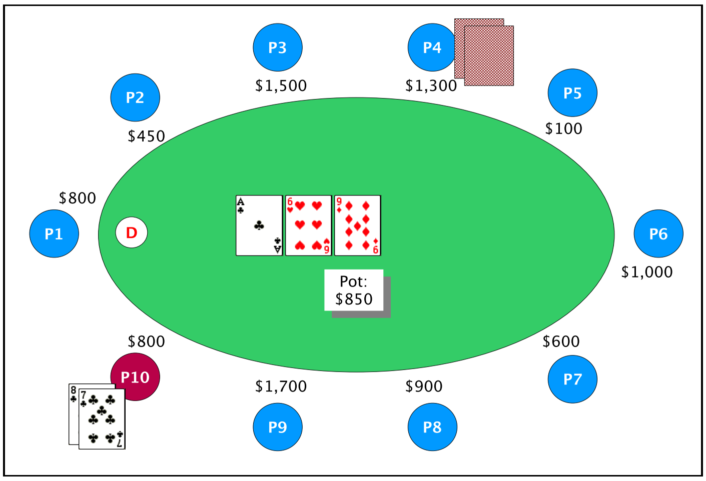


[1] We show our player's starting hand face-up for illustrative purposes. Note that during the actual game these cards are only visible to her.

The dealer now deals three cards, known as the flop, face up into the middle of the table. In this game the flop is *A*♣6♡9♢. The game state after the flop and before the next round of betting is shown below.



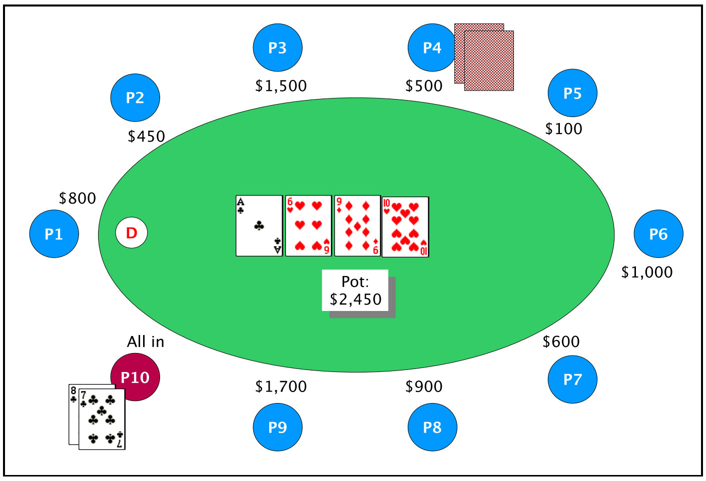
Play now proceeds with the second round of betting. All post-flop betting rounds start with the first active player to the left of the dealer. The options available to Player 3 are either to bet, check[2], or fold[3]. He checks, and play passes to Player 4. This player now has exactly the same options of bet, check or fold. He chooses to bet the relatively small amount of $200. The next to act, Player 9, must now either call $200, raise, or fold. He folds. Our player, with the possibility of making a straight, calls. Following this Player 3 folds. Since the active players (Players 4 and 10) have now entered the same amount into the pot, the betting round ends with the game state shown.



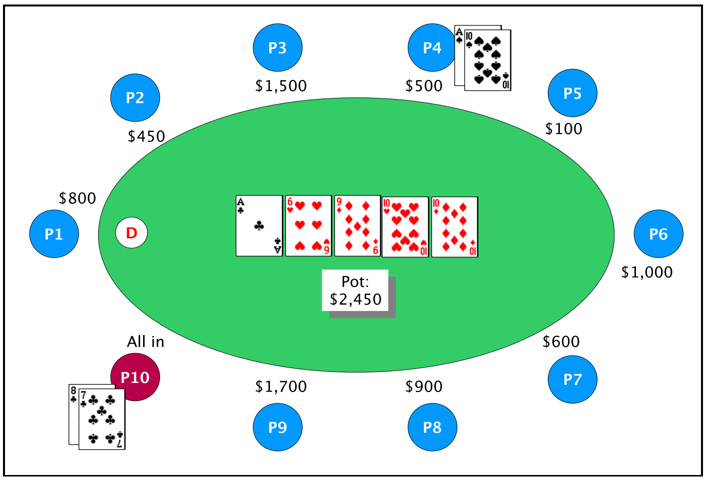
[2] Essentially a check can be thought of as calling a bet of zero in all post-flop rounds. Hence it is only available when there has yet to be a bet.

[3] Although a permissible action with no prior bet in a given round, folding is strictly dominated by checking. It costs a player nothing to check, and he still retains participation in the hand.

The turn card dealt onto the table is $T\heartsuit$. This is a very good card for our player since it completes the straight. Betting starts with Player 4, who checks, then moves on to Player 10. Our player bets \$200, whereupon Player 3 raises another \$600 to \$800. To call this raise costs our player \$600, which is exactly the amount left in her stack[4] In so doing she declares herself to be all in: all of her chips are now in the pot. With this the betting round is completed.



The hand is not over yet. The dealer now places the $T\diamondsuit$ river card onto the table, after which there would normally be a final round of betting. Here, however, there is only one active player with chips in his stack. Therefore the final betting round is foregone and play proceeds directly to the showdown. Players 4 and 10 now show their hole cards, and declare their best possible five-card poker hand using any combination of their own two private cards and the five community cards.



[4] Note that had Player 3 raised by a larger amount, Player 10 would still be able to call. The *table stakes* rule means that Player 10 could call with her final \$600 and announce herself all in. Any excess above this amount in the raise of Player 3 would immediately be returned him.

The best hand our player can make is the straight consisting of $6\heartsuit 7\clubsuit 8\clubsuit 9\diamondsuit T\heartsuit$[5]. The opponent uses his hole cards to make the full house $A\clubsuit A\spadesuit T\spadesuit T\heartsuit T\diamondsuit$.

Since a full house ranks higher than a straight, Player 4 wins and collects the $2,450 pot. Player 10, on the other hand, no longer possess any chips and is eliminated from the tournament in 10th place.

B.2 Five-Card Poker Hand Rankings

Cards rank from ace high down to two, with no discrimination between different suits. The winner of a Texas hold'em showdown is determined by reference to a complete ranking of all possible five-card poker hands, given in Table B.1. This table also details how ties within a hand ranking are broken.

Note that although the ranking constituent of some hands uses less than all five cards (for example, three of a kind), the seemingly extraneous cards are required in case of ties.

[5] Either available ten could be used to produce an equally ranked hand.

Hand	Description	Example	Tiebreaker
Straight flush	Five consecutive cards of the same suit	$4\spadesuit 5\spadesuit 6\spadesuit 7\spadesuit 8\spadesuit$	The holder of the highest ranked card within their straight flush wins. Two similar straight flushes of different suits split the pot.
Four of a kind	Four equally ranked cards	$3\heartsuit 3\clubsuit 3\diamondsuit 3\spadesuit Q\heartsuit$	The highest ranked four-card set wins.
Full house	Three cards of one rank and two of another	$2\heartsuit 2\spadesuit 2\diamondsuit 7\clubsuit 7\spadesuit$	The highest ranked three-card set wins.
Flush	Five cards of the same suit	$K\diamondsuit J\diamondsuit 8\diamondsuit 4\diamondsuit 2\diamondsuit$	The holder of the highest ranked card within their flush wins. If these are identical it is decided upon the second highest, and so on. Identical flushes of different suits split the pot.
Straight	Five consecutively ranked cards	$8\diamondsuit 9\diamondsuit T\spadesuit J\clubsuit Q\heartsuit$	The holder of the highest ranked card within their straight wins. Two similar straights split the pot.
Three of a kind	Three equally ranked cards	$J\heartsuit J\clubsuit J\spadesuit A\heartsuit 3\clubsuit$	The highest ranked three-card set wins.
Two pairs	Two cards of one rank and two of another	$Q\heartsuit Q\clubsuit 4\diamondsuit 4\clubsuit 6\spadesuit$	The highest ranked pair wins. If these are identical, the next highest ranked pair wins. If these are the same, the fifth cards are compared to determine the winner. If all comparisons are equivalent, the pot is split.
Pair	Two cards of equal rank	$K\clubsuit K\diamondsuit A\heartsuit Q\heartsuit 3\diamondsuit$	The highest ranked pair wins. If identical, the highest ranked non-paired cards are compared, the the next highest, and so on.
High card	None of the above	$A\spadesuit T\spadesuit 8\diamondsuit 7\clubsuit 4\clubsuit$	The highest ranked cards are compared, then the next highest, and so on.

Table B.1: The ranking of all five-card poker hands, plus the rules by which ties between hands of similar rank are broken. The hands are listed in order of decreasing strength.

Appendix C

Sklansky-Chubukov Hand Rankings

In 2005 a problem appeared on the forums of Two Plus Two publishing [96] which sought to value the worth of every possible starting hand in the following scenario. Suppose that there are two players in a no-limit Texas hold'em game, with blinds of $1 and $2. The small blind accidentally flips over his cards so that his opponent in the big blind sees them. The small blind decides that he will either go all in or fold his hand. If the small blind has $X in his stack after posting the dollar blind, for what values of X is it better to go all in than fold with? The problem assumes that the big blind has an infinite stack, and that he will call an all in bet if he has a positive expectation on the hand.

To calculate the values of X for each starting hand it is necessary to enumerate over all the big blind's possible holdings. With two cards accounted for and ignoring suit similarities there are 1225 combinations of two cards from the remaining fifty. In deciding whether to call the opponent weighs his expectation on the hand versus the odds that he is getting from the pot. For example, if the small blind bets $6 then the big blind is getting odds of 3-to-2 on a call. This is because he stands to win $9 for a bet of $6. If the probability of winning the hand is greater than the 40% implied by these odds[1] he should call.

There are four possible scenarios which can occur in determining the winner for this problem:

- Small blind bets and big blind folds $\Rightarrow$ small blind wins
- Small blind bets and big blind calls with hand which loses a showdown $\Rightarrow$ small blind wins

[1] To convert odds of *a*-to-*b* to a probability, one divides *b* by the sum of *a* and *b*

- Small blind bets and big blind calls with hand which wins a showdown ⇒ big blind wins
- Small blind folds ⇒ big blind wins

Given a starting hand for the small blind, a system of equations can be constructed over all possible hands for the big blind which equates the value of a fold and an all in move at a specific value of $X. Below this value it is correct for the small blind to move all in, and with a stack larger than $X it is correct for him to fold. These decisions maximize the player's expected winnings.

Ordering the starting hands in decreasing value of $X within this problem gives rise to the Sklansky-Chubukov rankings. These are named after the problem initiator, David Sklansky, and its solver Victor Chubukov.

The first and last few starting hands are given in Table C.1. The complete list can be found online at the homepage of Victor Chubukov [30].

Hand	**$X-value**
AA	∞
KK	954.00
AKs	554.51
QQ	478.01
AKo	331.89
JJ	319.21
...	...
72o	2.24
52o	2.18
62o	2.14
42o	1.98
32o	1.83

Table C.1: Best and worst starting hands in Texas hold'em by Sklansky-Chubukov ranking.

An alternative approach is to play all combinations of hands off against each other with all possible community cards and note the percentage of victories for each starting hand. Accurate estimates can be found by Monte-Carlo simulation over a large number

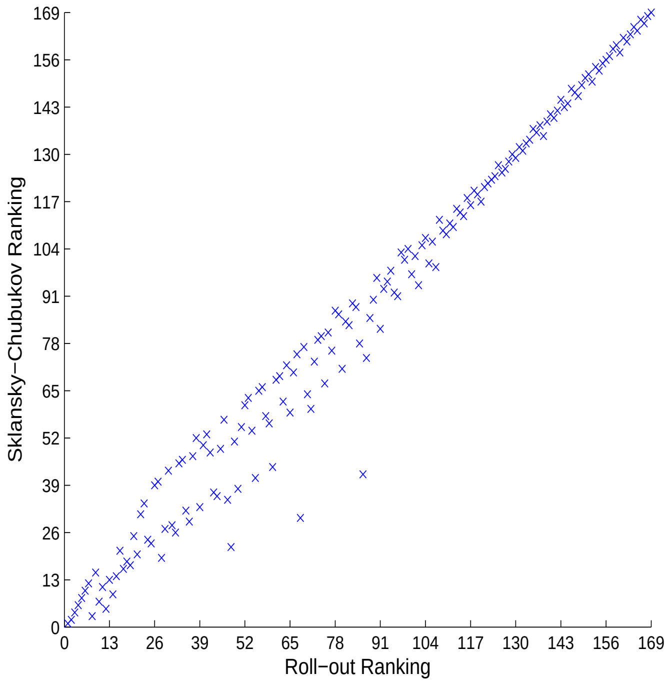


Figure C.1: A comparison of the hand rankings of the 169 different starting hands using roll-outs and the Sklansky-Chubukov $X-value.

of hands. The results yielded are known as "roll-out" rankings. An example of roll-out rankings constructed by simulation can be found online [21].

Figure C.1 gives a scatterplot of all 169 starting hands based on the two different methods.

It is clear by observation that there is a great similarity in the relative orderings yielded by the two approaches. The correlation between the two sets of rankings is 0.989. Three seemingly anomalous values occur towards the bottom of the plot, representing the lowest pairs: 44, 33, and 22. This shows that these hands are graded more highly in the Sklansky-Chubukov rankings than the roll-outs.

The Sklansky-Chubukov rankings contain an advantage over those from roll-outs since the former procedure incorporates blinds, and hence pot odds, into its formulation. With this the Sklansky-Chubukov $X-value yields a better absolute distinction between the strength of hands than the simpler percentage of hands won.

Appendix D

Published CIG 2007 Paper

This appendix contains a re-print of the paper accepted into the IEEE Symposium on Computational Intelligence and Games, Hawaii 2007 (CIG 2007). The paper condenses much of the work contained in this thesis, and was presented at the conference by the author.

An Investigation into Tournament Poker Strategy using Evolutionary Algorithms

Richard G. Carter
Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh, Scotland
r.g.carter@sms.ed.ac.uk

John Levine
Strathclyde Planning Group
Dept. of Computer and Information Sciences
University of Strathclyde, Scotland
john.levine@cis.strath.ac.uk

Abstract—**In this paper we assess the hypothesis that a strategy including information related to game-specific factors in a poker tournament performs better than one founded on hand strength knowledge alone. Specifically, we demonstrate that the use of information pertaining to opponents' prior actions, the stage of the tournament, one's chip stack size and seating position all contribute towards a statistically significant improvement in the number of tournaments won.**

Additionally, we test the hypothesis that a strategy which combines information from all the aforementioned factors performs better than one which employs only a single factor. We show that an evolutionary algorithm is successfully able to resolve conflicting signals from the specified factors, and that the resulting strategies are statistically stronger.

Keywords: evolutionary algorithms, game playing, tournament poker

I. Introduction

Researchers in artificial intelligence have long been interested in developing programs for games which are able to compete with and ultimately beat human opposition. Recent successes in the games of checkers, chess and backgammon have encouraged efforts on more complex games such as go, bridge and poker. Card games such as those mentioned present a particular challenge for researchers due the lack of complete information of the game state at any point in time. Opponents' hands must be modelled probabilistically, with inferential conclusions reached by computing the likelihood of each players' holding given their actions in the current and previous games. The development of poker players is especially exacting since the nature of the game demands that each competitor tries to deceive their opponents as to which cards they hold. Research into games of imperfect information such as poker has the potential to be extremely valuable, since reasoning under conditions of uncertainty is typical of many real-world problems.

Early game theoretic investigations of poker have been superseded by significant contemporary contributions. The GAMES Group at the University of Alberta has led the way in combining a strong analytical understanding of the game together with innovative approaches in opponent modelling. Their work over the last ten years has been so successful that a particular poker variant - two-player limit ring game Texas Hold'em - is now practically solved in the game theoretic sense, with continual improvements in the opponent modelling leading to increasingly strong players.

However, many hurdles have yet to be cleared before we reach the point of having a poker World Champion. Three difficulties in particular have to be overcome to take the current state of the art to such a level. Firstly, poker is almost always played between several competing players, rather than two. This means that multiple opponent models must be maintained. Secondly, the main event of the World Series of Poker held annually in Las Vegas uses a form of betting known as no limit. Where limit betting sets pre-specified increments to each player's bet and raise, no limit allows the competitors to bet any amount up to their current chip stack size. The third challenge is that the World Championships play Texas Hold'em in a tournament, rather than ring game format. A tournament setting adds extra complexity to the decision-making process compared to the same situation in a ring game. The nature of tournament play dictates that each player must balance accumulating and protecting their chips to ensure survival. This paper marks the first attempt to understand some of the issues involved in developing a player for Texas Hold'em played within a tournament, so that strong ring game programs may be more easily adapted to the tournament format.

Noted poker authors consistently state in the non-academic poker literature that factors contained within the tournament are important and should be included in the decision-making process before selecting a betting action. Taking hand strength knowledge as a given, the first hypothesis that we test in this research is that the inclusion of information relating to:

- opponents' prior betting actions
- the stage of the tournament
- one's own chip stack amount
- seating position

improves a player's tournament performance against three different static opponents. Through the use of Monte Carlo simulation on a slightly simplified form of no limit Texas Hold'em we find that all four of these factors are statistically significant in their contribution.

The strategies we find determine which hands should be played dependent upon the additional game information. To create a strategy using all the available information, however, we cannot simply combine the suggested betting actions from the single factor results. This is because the single factor strategies are often in conflict in any given situation. Therefore the second hypothesis that we test is that an evolutionary algorithm is able to resolve such disagreements, and that the resulting strategy is stronger than any achieved by employing only one of the factors in isolation. Results contained within this work show this to be true.

II. Background

A. *Texas Hold'em*

Texas Hold'em (also known simply as Hold'em) is the most widely played poker variant. It is this form which is used as the main event in the annual World Series of Poker. A game of Texas Hold'em can be played with up to 22 players, although it is more usual to see between two and ten players at a single table. A comprehensive set of rules for the game of Texas Hold'em is available online [1].

Texas Hold'em typically employs one of three different forms of betting structure: limit, pot limit, or no limit. In limit Hold'em, the size of the bets are fixed amounts. In the first two betting rounds, each bet or raise is a set amount. In the final two rounds the fixed bet size doubles. In pot limit Hold'em a player may wager any amount up to the size of the pot. No-limit Hold'em removes this restriction by allowing each player to bet any amount up to their stack size.

As with all poker games, Hold'em can be played either as a ring game (also known as cash game) or in a tournament. In a ring game the players contest pots with real money and no predetermined end time. A poker tournament, on the other hand, is played with tournament chips and ends once the game has been reduced to a single player. In a ring game players may continually enter and exit the table, and players who lose all their chips are able to purchase more to continue in the game. By contrast, the players in a tournament usually buy a set number of tournament chips before the game and are eliminated from the competition if their stack size reaches zero. The major differences between ring game and tournament play are summarised in Table I.

TABLE I
STRUCTURAL DIFFERENCES BETWEEN RING GAME AND TOURNAMENT POKER

Difference	**Ring Game**	**Tournament**
Entry fee	Variable	Tournament cost
Chips	Money replacement	Game tokens
Blinds	Fixed	Rising schedule
Number of players	Limited to a table	Unlimited
Game exit	Player discretion	Zero chips
Profit and loss	On each hand	Based on finish

B. *Previous Poker Research*

The first academic investigations into the game of poker were undertaken in the mid-twentieth century. Early pioneers in the field of game theory, such as von Neumann and Morgenstern [2], employed greatly simplified poker variants to formulate a framework for strategy selection in non-cooperative environments. The toy pokers that were examined were typically only two- or three-player games, and used pared decks of cards to reduce the space of possible strategies.

More recent investigations into poker have focused on developing computer programs which are able to play more realistic variants to a high standard. The most advanced and successful work on computer poker play to date has been produced by the Game-playing, Analytical methods, Minimax search, and Empirical Studies (GAMES) Group at the University of Alberta, led by Jonathan Schaeffer.

The GAMES Groups' poker research started with the development of programs designed to play ten-player limit Texas Hold'em in a ring game, and employed a combination of statistical measures and expert rules to effect decision-making [3]. Billings *et al.* [4] then turned to two-player limit Hold'em, for which they were able to derive "pseudo-optimal" strategies. Recent efforts by the GAMES Group target the addition of opponent modelling to the game theoretic foundation, with greatly improved results [5].

Outside of academia, many books have been written on strategy for play in Texas Hold'em. Whilst poker literature for ring game play abounds, relatively little has been written on Texas Hold'em tournament strategy. Those that exist discuss the differences between ring game and tournament play, and expound on how certain factors affect strategic considerations.

The most common strategic messages in the non-academic writings relate to how one's range of playable starting hands should increase throughout a tournament, the importance of stack size in betting decisions, and how the payoff structure of the tournament determines correct strategy. The purpose of this research is to empirically validate some of these assertions.

C. *Evolutionary Algorithms Applied to Poker*

Evolutionary algorithms have been commonly applied to search for strong strategies in a wide variety of different games. One notable example of this is the development of the checkers player *Blondie24* by Kumar Chellapilla and David Fogel [6].

The first, and to date most extensive attempts to apply evolutionary computation to poker have been performed by Luigi Barone and Lyndon While [7], [8]. The authors develop poker players that are able to adapt strategically given inputs from their environment, such as their own hand strength, seating position, bet size, and a measure of their opponents' playing styles. The experiments performed in their research use Texas Hold'em with limit betting, and are employed within a ring game format.

Graham Kendall and Mark Willdig [9] employ evolutionary methods to learn to play a simplified draw poker game. Candidates from the population are played at tables containing opponents of different styles, and the adaptive players are seen to adjust their strategy appropriately to each situation.

Texas Hold'em has also been used in research by Jason Noble [10]. Rather than play against static opposition, his work seeks to develop strong poker players through self-play. However, the primary focus of these studies is on the comparison of different co-evolutionary methods, and poker is simply used as the test bed for the experiments.

The programs developed by the GAMES Group have been shown to perform well against professional poker players, but no such achievements were reported in the work on evolutionary poker programs. Whilst it presently seems that the approach of marrying game theoretic understanding with opponent modelling has the greatest promise in achieving the strongest poker programs, results from evolutionary methods have highlighted some of the strategic problems faced in playing poker games.

III. Specification and Implementation

All the experiments performed focus on ten-player winner-takes-all all in or fold pre-flop Texas Hold'em. This format makes one simplification to an authentic poker tournament. The restriction of the players' betting actions to all in or fold reduces the strategy space to a tractable size. Substituting the more flexible betting choice with a binary decision shifts the focus to the more general and important strategic question of when to bet, rather than how much. Since players will have either bet all of their chips or folded before the flop, the resulting betting rounds are redundant. All in or fold pre-flop betting strategies have been proposed for use in real tournaments, and we use three of these taken from the poker literature as the opponents within our experiments.

Whilst it is more usual for ten-player tournaments to employ a percentage payout structure, the "shootout" format with a single winner can also be employed. The winner-takes-all design only credits a player for finishing first, and hence second place is equivalent to finishing last. Establishing the competitions in this way ensures that we assess each strategy's ability to win tournaments, and not just their capacity for tournament survival.

All players start with $1,000 in tournament chips. The tournaments use eleven levels, with the blinds increasing as the tournament progresses. Each level consists of ten hands, except for the final level which is used until the tournament concludes.

There were a total of three different opponents used across all the simulations. These encoded the Sklansky Basic strategy, Sklansky's Improved strategy, and the Kill Phil Rookie strategy, and are explained below.

The first two strategies are taken from the book "Tournament Poker for Advanced Players" [11]. The Sklansky Basic strategy is highly restrictive, and will only bet the very best starting hands once another player has already bet into the pot. If no other player has yet bet, this strategy will move all in with a slightly larger subset of hands. The Sklansky Improved strategy is similarly restrained in the hands it will play if an opponent has entered the pot. Where this strategy differs from the first is in its use of the ratio of the player's stack to the total amount of the blinds to determine playable hands when no other player has yet made a bet.

The third opponent employed is based on one from the book "Kill Phil" [12]. This book contains several strategies of increasing complexity, and these experiments use the simplest, so-called "Rookie", strategy. Similar to the two Sklansky strategies, the Kill Phil Rookie strategy contains instructions on which hands are playable depending upon whether or not an opponent has yet bet into the pot. The major difference in this strategy is that the classification of playable hands is determined by the number of players remaining and the tournament level.

For all the experiments, we seat our test player at a table against nine similar opponents from either of the three mentioned above.

Note that we do not incorporate any opponent modelling within these experiments. Whilst we acknowledge that this is an essential element of strong poker strategy in any setting, here we seek more general results regarding the dynamics of tournament play.

By discriminating between "suited" and "offsuit" hands there are 169 different starting hands in Texas Hold'em. Using the hand strength ordering of Sklansky and Chubukov [13], we create thirteen groups of thirteen hands: the strongest in Group 1 down to the weakest in Group 13. Note that the classes are not all of precisely the same size due to the varying frequencies of pairs, suited, and unsuited hands. Whilst the classification employed here is extremely coarse, it is sufficient to show the dynamics of tournament strategy that we seek.

The first suite of experiments use Monte Carlo simulation to assess strategies which employ either hand strength knowledge alone, or this in conjunction with another factor believed to influence decision-making. All strategies take the form of a threshold value between zero and thirteen. In measuring the performance of players using hand strength knowledge alone, a strategy (x) represents a player who will move all in with cards in the groups higher than or equal to x, and fold otherwise. When we incorporate one extra game factor the representation becomes two-dimensional, (x, y). The x-value is used if the binary variable is true, and the y-value is used if the binary variable is false. The binary variables used are summarised in Table II.

TABLE II

Binary variables used to assess decision factors

Factor	Binary Variable
Opponents' actions	No prior bet in the current hand
Tournament stage	Tournament level ≤ 6
Chip stack amount	M ≤ 5
Seating position	Early position

The variable M used to assess chip stack size is taken from Harrington [14] and is simply the ratio of one's stack to the total of the blinds at the current level. Early and late position are determined relative to the dealer, such that a player in the first half of those required to act is deemed to be in early position.

The simulations sequentially test every possible strategy over a series of 200 tournaments. A player scores one point for winning a tournament, and zero otherwise.

To extend the aforementioned representation into multiple dimensions for each possible binary variable presents problems due to the computational time required to cover all of the strategies. Instead, we employ an evolutionary algorithm which is able to encode an action for any possible combination of the four factors.

A strategy within the evolutionary algorithm is encoded as a chromosome of sixteen real numbers, each value taken from the interval $[0, 14)$. A single gene determines the hands played in one of the 2^4 scenario combinations of the four binary variables. When required to act the evolutionary player refers to the relevant gene, and moves all in if the floor of the value is less than or equal to the group containing their hand.

The evolutions each commence with a population of twenty randomly generated strategies. This comparatively small population was found to be sufficient to produce significant results. Individuals within the population reproduce according to tournament selection, with two elites passing through to the next generation unaltered. Reproduction occurs at a rate of 70%, and employs uniform crossover with equal weighting between the parents. Mutation applies a Gaussian shock to a randomly selected allele with a standard deviation of 2, and reflection occurs at the upper and lower bounds. Many of these shocks do not result in a change in an individual's strategy due to the representation used. In these experiments we use a relatively high mutation rate of 20%, although smaller values were found to produce similar results.

IV. Experimental Results

A. Monte Carlo Simulation Results

Results from the first set of experiments, in which the test players act according to hand strength alone, are plotted in Figure 1.

The graph clearly shows that betting with either too many hand groups or too few results in reduced performance. Against all three opposing strategies the highest number of tournament wins arise when the player moves all in with around the top four groups of starting hands.

There is, unfortunately, a large amount of noise seen in this plot. This can be reduced by simulating each strategy over a larger number of tournaments than the 200 used here. Further simulations were performed with the players each playing 1,000 tournaments, and more clearly defined peaks were observed. We present the results over 200 tournaments for comparison with those from the inclusion of the binary variables which follow. In these latter cases, simulating all 196 (x, y) strategy pairs over 1,000 tournaments would be extremely time-consuming.

The next suite of experiments expands the strategy representation from one to two threshold values. Each value corresponds to playable hand groups depending on the state of the additional binary variable. Here again each possible (x, y) strategy, $(x, y \in \{0, 1, ..., 13\})$ is played in 200 tournaments against a table of each of the three adversarial strategies. Figures 2, 3, 4, and 5 show the results achieved against the Sklansky Basic strategy. Plots showing scores against the Sklansky Improved and Kill Phil Rookie strategies can be found at the authors' website [15], and all show similar surfaces.

There are two interesting aspects to these graphs. Firstly, all are asymmetric in the line $y = x$. This shows that the information contained within the binary variable is of consequence to tournament performance. For comparison purposes a control experiment was performed in which the binary variable was chosen to be the result of a coin toss. The plot of this experiment in Figure 6 shows much greater symmetry, subject to noise.

The second important aspect of these results concerns the number of tournaments won by incorporating the additional information. The peaks of those graphs representing the inclusion of prior bet knowledge, tournament level, chip stack size, and seating position are far larger than that achieved through hand strength knowledge alone. The peak from the control experiment utilising a coin toss is comparable to hand strength only, as expected.

The strategies leading to the best results against the Sklansky Basic (SB), Sklansky Improved (SI), and Kill Phil Rookie (KPR) strategies are summarised in Table III.

TABLE III
Best strategies found by Monte Carlo simulations against various opponents

Game Knowledge	**SB**	**SI**	**KPR**
Hand strength only	(4)	(3) & (5)	(4)
Hand and prior bet	$(9, 0)$	$(13, 3)$	$(12, 3)$
Hand and level	$(1, 12)$	$(1, 11)$	$(3, 12)$
Hand and stack	$(5, 0)$	$(12, 2)$	$(8, 2)$
Hand and position	$(8, 2)$	$(13, 2)$	$(11, 1)$

It is evident from these results that the state of a binary variable polarises the test player's best strategy in all cases against all opponents. The nature of these strategies also bear favourable comparison to suggested tournament play in the non-academic poker literature.

If no opponent has yet bet the best strategy is to play many hands. If the pot has already been opened, however, a lesser number of hands should be played. Poker authors state that a player should be much tighter if an opponent has already bet into the pot.

Knowledge of tournament level affects the best play by being tighter in the early levels and looser in the late ones. These strategies also follow poker author's recommendations. They state that at the start of a tournament one should be more concerned about being eliminated and so should play to protect chips. Then in the later levels one should play more hands to try to steal the increased blinds.

A player with a small number of chips is seen to move all in with more hand groups than one with a relatively large stack. The escalating blinds used in tournaments mean that

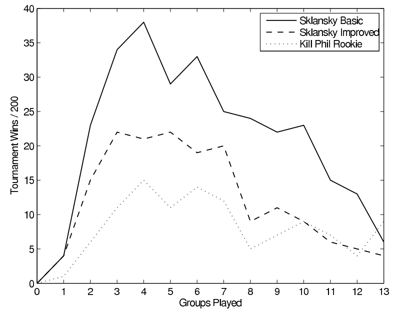


Fig. 1. Tournament wins against various opponents using hand strength knowledge only

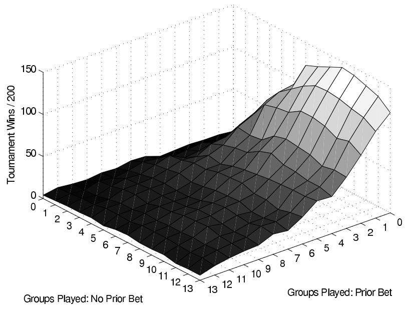


Fig. 2. Tournament wins against Sklansky Basic opponents for strategies incorporating opponents' prior bet knowledge

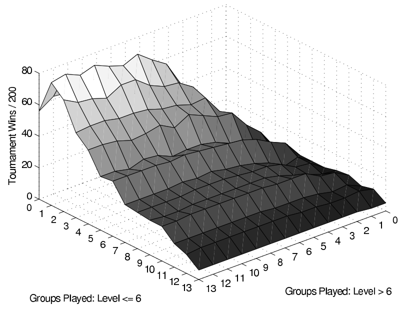


Fig. 3. Tournament wins against Sklansky Basic opponents for strategies incorporating tournament level knowledge

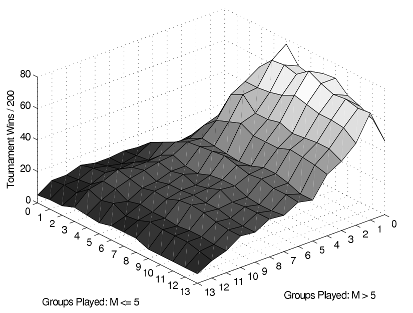


Fig. 4. Tournament wins against Sklansky Basic opponents for strategies incorporating chip stack knowledge

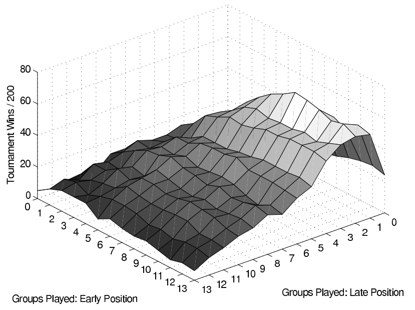


Fig. 5. Tournament wins against Sklansky Basic opponents for strategies incorporating seating position knowledge

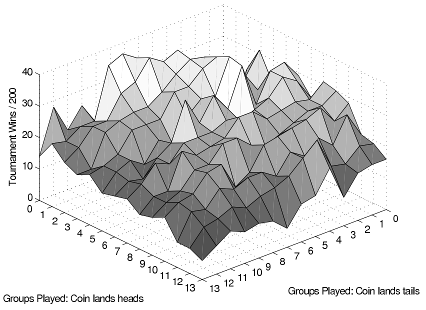


Fig. 6. Tournament wins against Sklansky Basic opponents for strategies incorporating coin toss knowledge (control experiment)

persistently waiting for strong hands leads a player to being anted away. Again the best strategies found here conform to the suggested style.

The only results here that are in conflict with poker authors concern those from the inclusion of seating position knowledge. Here we see that a player in early position bets more hands than one in a later seat. It is more commonly suggested that correct strategy is the opposite of this. A late seat means that there are less players to act after oneself, and so specifically in an unopened pot a bet with a slightly worse hand than normal can prove to be profitable. The reason for the disagreement in our experiments is likely to be two-fold. Firstly, all the opposing strategies play a more restricted set of hands once a player has entered the pot. When our test player is in early position it seems to be able to take advantage of this by betting weaker hands and essentially "scaring" the opposition. The second likely cause is that we are not able to distinguish between those occasions where the test player is first into a pot. It could still be the case that in these scenarios our player is playing more aggressively.

It is this desire to see all factors in combination that leads to the next set of experiments. The results previously seen in Table III contain potential conflicts for a strategy seeking to incorporate all game-related factors. For example, suppose we are playing the Sklansky Basic opponents and are in the latter half of a tournament with a large stack size. The best strategies found would suggest playing any hand in the top twelve groups due to the tournament level, but contradictorily that we should fold all hands since we have a large stack.

The simulation-based approach is expedient in the case of only one decision factor, but as we include more the time taken to cover the strategy space grows exponentially. For this reason we turn towards an evolutionary approach.

B. Evolutionary Algorithm Results

Candidates within a population encode strategies that dictate which hand groups to play in any of the 16 possible scenarios within the strategy hypercube. The guided stochastic search then moves the population in the direction of stronger solutions.

A randomly seeded population of twenty candidate solutions were each played in 200 tournaments per generation against tables consisting of each of the three opponents. The average population fitness per generation, and the incremental global best solutions are shown for the evolutions against Sklansky Basic, Sklansky Improved, and Kill Phil Rookie opposition in Figures 7, 8, and 9 respectively.

All of these plots show that the average population fitnesses are quick to increase in the first ten generations of the evolution. Strong solutions are more likely to be maintained, whilst weaker strategies fail to reproduce. In the early generations the rate of finding new global best strategies is at its fastest. These results are in keeping with generally observed trends in the use of evolutionary algorithms.

Improvement in the global best solution against Sklansky Basic opponents noticeably levels off by generation 25. The evolutions against the other two opponents were still finding

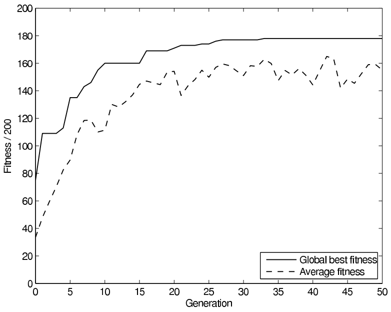


Fig. 7. Evolution against Sklansky Basic opponents

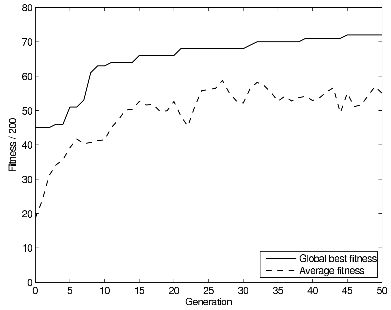


Fig. 8. Evolution against Sklansky Improved opponents

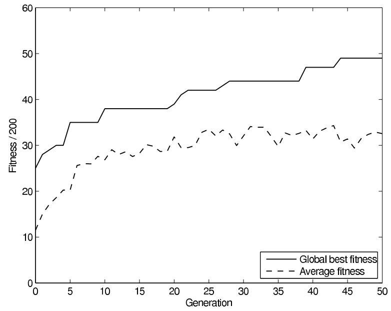


Fig. 9. Evolution against Kill Phil Rookie opponents

new global bests approximately every five generations when the runs were terminated. It is expected that even stronger strategies would have been uncovered had the number of generations been increased.

The most remarkable aspect of these graphs is in observing the number of tournament wins achieved by the best found strategies. At the termination of the run, the best solution against Sklansky Basic had won 178 out of 200 tournaments, a rate of almost 90%. This figure is a marked improvement on the comparable 38 tournament wins gained using hand strength alone shown previously in Figure 1. The evolutions against Sklansky Improved and Kill Phil Rookie opposition also show marked improvements, with both more than tripling the top score found based only on hand strength.

Next, we took all the best strategies found using the different representations and played them off against their respective opponents over 5,000 further tournaments. Performing these experiments over a larger number of tournaments allows us to gain a greater level of statistical surety in the scores attained. The number of tournament wins in these enlarged experiments are shown in Table IV.

TABLE IV
TOURNAMENT WINS (OUT OF 5,000) OF BEST FOUND STRATEGIES AGAINST VARIOUS OPPONENTS

Game Knowledge	SB	SI	KPR
Hand strength only	1,004	565	319
Hand and prior bet	3,562	1,095	595
Hand and level	1,997	746	574
Hand and stack	1,637	819	534
Hand and position	1,417	715	460
Hand and all factors	4,340	1,165	786

For all three opponents the knowledge of whether or not another player has bet into the pot shows the largest gains. Intuitively, we would expect knowledge of one's opponents actions to be the most important factor since all players are in competition with one another for the money in the pot. Least improvement comes from seating position knowledge. The lesser importance of this is most probably due to the all in or fold nature of the tournaments. Poker authors have stated the much of the positional advantage is reduced by such strategies.

Statistical analysis was performed on these results using the Z-test for the equality of two proportions. Firstly, we tested the null hypothesis that the proportion of tournaments won by the strategies incorporating a single binary variable was the same as those using hand strength alone against each respective opponent. All were found to be rejected as is shown in Table V. From this we conclude that the inclusion of any of these extra pieces of game information significantly improves tournament performance.

TABLE V
P-VALUES FOR THE PROPORTION OF TOURNAMENTS WON WITH THE INCLUSION OF A GAME FACTOR COMPARED TO HAND STRENGTH ALONE

Factor	SB	SI	KPR
Prior bet	< 0.00001	< 0.00001	< 0.00001
Level	< 0.00001	< 0.00001	< 0.00001
Stack	< 0.00001	< 0.00001	< 0.00001
Position	< 0.00001	< 0.00001	< 0.00001

Following this we tested the null hypothesis that the strategies found by the evolutionary algorithms won the same proportion of tournaments as those achieved in the best case of hand strength and one other factor (in each case this factor is knowledge of an opponent's prior bet). Table VI reveals that all null hypotheses were rejected at the 95% confidence level. Hence we conclude that the incorporation of knowledge from all four factors has produced strategies with a markedly higher win rate compared to those utilising only a single factor.

TABLE VI
P-VALUES FOR THE PROPORTION OF TOURNAMENTS WON WITH THE INCLUSION OF ALL GAME FACTORS COMPARED TO HAND STRENGTH AND PRIOR BET KNOWLEDGE

	SB	SI	KPR
Evolved player	< 0.00001	< 0.05	< 0.00001

It should be remembered that the opposition strategies are able to act on exact knowledge of their hand, stack size, and tournament level. Our test players are severely limited in the granularity of the information they can base decisions on due to the coarse classifications employed. The results given are perhaps even more impressive in the light of this.

To analyse the nature of the best evolutionary strategies found we calculated the average hand group played for each possible state of the four binary variables. These values are presented in Table VII.

TABLE VII
AVERAGE ALLELE VALUES IN GLOBAL BEST STRATEGIES AGAINST EACH OPPONENT

Scenario	SB	SI	KPR
No prior bet	8.5	7.6	8.4
Prior bet	2.1	5.9	5.0
Level ≤ 6	4.5	5.0	5.0
Level > 6	6.1	8.5	8.4
M ≤ 5	6.9	6.8	8.0
M > 5	3.8	6.8	5.4
Early position	6.6	8.6	8.0
Late position	4.0	4.9	5.4

An inspection of these results shows the same trends as those previously found in Table III. The strategies typically play less hands in the following situations: after an opponent has bet, in the early levels of a tournament, with a large stack size, and in late position.

It is interesting to note that these best strategies have managed to resolve conflicts in the signals given by each separate game factor. The strategies found by the evolutionary algorithm manage to implicitly weigh the importance of each piece of information in formulating a betting action depending upon the situation.

V. Further Work

The framework used in the evolutionary experiments is extremely amenable to expansion. For example, any other factor thought to influence decision-making could be readily incorporated into the representation.

One possible enhancement would be to increase the resolution on the factors already employed. An enlarged number of classifications of stack size, tournament level, and seating position would further refine the strategy space.

The hand groupings used in these experiments are too coarse to be of direct benefit in a real poker game. Splitting out the upper groups into better defined classes, and condensing the bad hands into a smaller number of larger groups would better reflect the differences in starting hand potential.

An interesting topic for future investigation is to better understand the relative importance of each factor dependent upon tournament level. For example, some poker authors state that stack size becomes more important than hand strength towards the end of a tournament.

Clearly the most important continuation of this research is the removal of the all in or fold betting restriction. As well as being able to bet fractional amounts of one's stack, the strategies called for would also have to encode for post-flop play. It remains an open question whether an unrestricted Texas Hold'em tournament strategy could be found within such a framework.

VI. Conclusions

In this research we set out to show that information available to players related to their opponents' prior actions, the stage of the tournament, chip stack size, and seating position are all important elements in the strategy of a Texas Hold'em tournament player. By comparing the performance of players who use knowledge of their hand strength alone to those who incorporate each of the above factors we have shown that there is a statistically significant improvement in the number of tournaments won by the latter.

We then demonstrated that an evolutionary algorithm is able to resolve conflicting signals from the decision-making factors. The strategies which combine all the available information are seen to perform to a statistically higher standard than those which use only one piece of knowledge in conjunction with their hand strength.

The "strategies" we have derived in this research should perhaps technically be termed "counter-strategies", since each is specific to their own particular opponent. However, the interpretations of the counter-strategies found (i.e. play more hands if no opponent has bet, if one has a small stack, late in the tournament etc.) are the same against all three opponents of increasing complexity. It is compelling to suggest that these tactics, which mirror the guidance given in the non-academic poker literature, should underpin the strategy of a competent player against any given opponent(s). Whilst we cannot go as far as to claim this outright, our results do lend weight to that argument.

At present the approach of game theoretic understanding and opponent modelling has been seen to yield stronger poker programs than any that have been found by evolutionary algorithms alone. However, transitioning these limit ring game strategies to a no limit tournament setting requires additional understanding of the complexities of tournament play. The results presented in this research show that an evolutionary approach is well suited to the task of analysing tournament strategy, and that it can be used to complement other forms of computer poker research.

Acknowledgments

The authors would like to thank the University of Alberta's GAMES Group for the publicly available Texas Hold'em Java code [16] which was integral to this research.

References

[1] R. Ciaffone, "Robert's rules of poker," WWW Address: "http://www.worldseriesofpoker.com".

[2] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behaviour*. Princeton, NJ: Princeton University Press, 1953.

[3] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron, "The challenge of poker," *Artificial Intelligence Journal*, vol. 134, no. 1, pp. 201–240, 2001.

[4] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron, "Approximating game-theoretic optimal strategies for full-scale poker," in *Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003, pp. 899–905.

[5] F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and C. Rayner, "Bayes' bluff: Opponent modelling in poker," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, 2005, pp. 550–558.

[6] K. Chellapilla and D. B. Fogel, "Evolving an expert checkers playing program without using human expertise," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 4, pp. 422–428, 2001.

[7] L. Barone and L. While, "Evolving computer opponents to play a game of simplified poker," in *Proceedings of the 1998 International Conference on Evolutionary Computation*, Anchorage, AK, 1998, pp. 108–113.

[8] ——, "An adaptive learning model for simplified poker using evolutionary algorithms," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, 1999, pp. 153–160.

[9] G. Kendall and M. Willdig, "An investigation of an adaptive poker player," in *14th Australian Joint Conference on Artificial Intelligence*, Adelaide, Australia, 2001, pp. 189–200.

[10] J. Noble, "Finding robust Texas Hold'em poker strategies using pareto coevolution and deterministic crowding," in *Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA 02)*, Las Vegas, NV, 2002, pp. 233–239.

[11] D. Sklansky, *Tournament Poker for Advanced Players*. Las Vegas, NV: Two Plus Two Publishing, 2004.

[12] B. Rodman and L. Nelson, *Kill Phil - The Fast Track to Success in No-Limit Hold'em Poker Tournaments*. Las Vegas, NV: Huntintdon Press, 2005.

[13] V. Chubukov, "Sklansky-Chubukov hand rankings," 2006, WWW Address: "http://www.decf.berkeley.edu/ chubukov/rankings.html".

[14] D. Harrington and B. Robertie, *Harrington on Hold'em: Expert Strategy for No-Limit Tournaments; Volume II: The Endgame*. Las Vegas, NV: Two Plus Two Publishing, 2005.

[15] R. Carter, "Monte Carlo and evolutionary algorithm results for CIG'07," 2006, WWW Address: "http://homepages.inf.ed.ac.uk/s0343921/".

[16] U. of Alberta GAMES Group, "Computer poker (Texas Hold'em) Java source code," 2000, WWW Address: "http://spaz.ca/aaron/poker/src/".

Bibliography

[1] Betfair poker. [Internet]. WWW address: “http://poker.betfair.com” [Accessed 29 June 2007].

[2] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, NY, 1985.

[3] Robert Axelrod. Evolving new strategies: The evolution of strategies in the iterated prisoners dilemma. *The Complexity of Cooperation*, 1997.

[4] Luigi Barone and Lyndon While. Evolving computer opponents to play a game of simplified poker. In *8th University of Western Australia Computer Science Research Conference*, Seoul, Korea, 1997.

[5] Luigi Barone and Lyndon While. An adaptive learning model for simplified poker using evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 153–160, 1999.

[6] David Beaumont. Java implementation of Mersenne Twister. [Internet]. WWW address: “http://www.math.sci.hiroshima-u.ac.jp/∼m-mat/mt/versions/java/MTRandom.java” [Accessed 29 June 2007].

[7] Hans-George Beyer. Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanical Engineering*, 186:239–269, 2000.

[8] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *Eighteenth International Joint Conference on Artificial Intelligence*, pages 899–905, Acapulco, Mexico, 2003.

[9] Darse Billings. The First International RoShamBo Programming Competition Results. [Internet]. WWW address: “http://www.cs.ualberta.ca/∼darse/rsb-results1.html” [Accessed 29 June 2007].

[10] Darse Billings. Vexbot wins poker tournament. *International Computer Games Association Journal*, 26:281, 2003.

[11] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence Journal*, 134(1):201–240, 2001.

[12] Darse Billings, Aaron Davidson, Terence Schauenberg, Neil Burch, Michael Bowling, Robert C. Holte, Jonathan Schaeffer, and Duane Szafron. Game-tree search with adaptation in stochastic imperfect-information games. In *Computers and Games*, pages 21–34, Ramat-Gan, Israel, 2004.

[13] Darse Billings, Denis Papp, Lourdes Pena, Jonathan Schaeffer, and Duane Szafron. Using selective-sampling simulations in poker. In *Proceedings of the AAAI Spring Symposium*, pages 13–18, Stanford, CA, 1999.

[14] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Opponent modeling in poker. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference*, pages 493–499, Madison, WI, 1998.

[15] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Poker as a testbed for machine intelligence research. In *12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI ’98)*, pages 228–238, Vancouver, Canada, 1998.

[16] Darse Billings, Lourdes Pena, Jonathan Schaeffer, and Duane Szafron. Using probabilistic knowledge and simulation to play poker. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Innovation Applications of AI Conference*, pages 697–703, Orlando, FL, 1999.

[17] Ken Binmore. *Fun and Games: A Text on Game Theory*. Houghton Mifflin, Boston, MA, 1991.

[18] Christopher Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, England, 1995.

[19] Emile Borel. *Applications aux Jeux des Hazard*. Gautier-Villars, Paris, France, 1938.

[20] Jürgen Branke, Christian Schmidt, and Hartmut Schmeck. Efficient fitness estimation in noisy environments. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 243–250, 2001.

[21] Steve Brecher. Percentage of pots won with hold'em hands. [Internet]. WWW address: "http://www.gocee.com/poker/HE_Value.htm" [Accessed 29 June 2007].

[22] D. Brunson, B. Baldwin, M. Caro, J. Hawthorne, D. Reese, and D. Sklansky. *Doyle Brunson's Super System*. Cardoza Publishing, New York, NY, 2002.

[23] Seth Bullock. Are artificial mutation biases unnatural. In *Fifth European Conference on Artificial Life*, pages 64–73, Lausanne, Switzerland, 1999.

[24] Ken Buntjer. *The Secret to Winning Big in Tournament Poker*. Red Rose Publishing, 1995.

[25] Murray Campbell, A. Joseph Hoane Jr., and Feng hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1-2):57–83, 2002.

[26] Gary Carson. *The Complete Book of Hold'em Poker*. Kensington Publishing Corp., New York, NY, 2001.

[27] Kumar Chellapilla and David B. Fogel. Evolving neural networks to play checkers without expert knowledge. *IEEE Transactions on Neural Networks*, 10(6):1382–1391, 1999.

[28] Kumar Chellapilla and David B. Fogel. Evolving an expert checkers playing program without using human expertise. *IEEE Transactions on Evolutionary Computation*, 5(4):422–428, 2001.

[29] Chessbase. A short history of computer chess. [Internet]. WWW address: "http://www.chessbase.com/columns/column.asp?pid=102" [Accessed 29 June 2007].

[30] Victor Chubukov. Sklansky-Chubukov hand rankings, 2006. [Internet]. WWW address: "http://www.decf.berkeley.edu/∼chubukov/rankings.html" [Accessed 29 June 2007].

[31] Bob Ciaffone and Jim Brier. *Middle Limit Holdem Poker*. Bob Ciaffone, 2001.

[32] D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K.V. Price. *New ideas in optimization*. McGraw-Hill, Maidenhead, England, 1999.

[33] Nichael L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 183–187, Mahwah, NJ, 1985. Lawrence Erlbaum Associates, Inc.

[34] W.H. Cutler. An optimal strategy for pot-limit poker. *The American Mathematical Monthly*, 82(4):368–376, 1975.

[35] F.A. Dahl. A reinforcement learning algorithm applied to simplified two-player texas holdem poker. In *Proceedings of the 12th European Conference on Machine Learning*, pages 85–96, Freiburg, Germany, 2001.

[36] Charles Darwin. *The Origin of Species*. John Murray, London, England, 1859.

[37] Aaron Davidson. Using artificial neural networks to model opponents in Texas hold'em, 1999. [Internet]. WWW address: "http://spaz.ca/aaron/poker/poker.html" [Accessed 29 June 2007].

[38] Aaron Davidson. Opponent modeling in poker: Learning and acting in a hostile and uncertain environment. Master's thesis, University of Alberta, Edmonton, Canada, 2002.

[39] Aaron Davidson, Darse Billings, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI 2000)*, pages 1467–1473, Las Vegas, NV, 2000.

[40] JE Davis and G. Kendall. An investigation, using co-evolution, to evolve an awari player. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 2, pages 1408–1413, 2002.

[41] Kenneth A. de Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

[42] Dan Egnor. Iocaine powder explained. [Internet]. WWW address: "http://www.ofb.net/ egnor/iocaine.html" [Accessed 29 June 2007].

[43] Susan L. Epstein. Game playing: The next moves. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, FL, 1999.

[44] Chris Ferguson and Thomas S. Ferguson. On the Borel and von Neumann poker models. *Game Theory and Applications*, 9:17–32, 2004.

[45] Chris Ferguson and Tom Ferguson. Uniform (0, 1) two-person poker models. In *Proceedings of Eleventh International Symposium on Dynamic Games and Applications*, 2004.

[46] Nicholas V. Findler. Studies in machine cognition using the game of poker. *Communications of the ACM*, 20(4):230–245, 1977.

[47] Nicholas V. Findler, Heinz Klein, R. Channing Johnson, Alexander Kowal, Zachary Levine, and John Menig. Heuristic programmers and their gambling machines. In *Proceedings of the ACM National Conference*, pages 28–37, San Diego, CA, 1974.

[48] David B. Fogel. *Blondie 24: Playing At The Edge of AI*. Morgan Kaufmann, San Francisco, CA, 2001.

[49] Lawrence J. Fogel, Alvin J. Owens, and Michael J. Walsh. *Artificial Intelligence through Simulated Evolution*. Wiley, New York, NY, 1966.

[50] Ian Frank, David Basin, and Alan Bundy. Combining knowledge and search to solve single-suit bridge. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 195–200, 2000.

[51] Colin M. Frayn. An evolutionary approach to strategies for the game of Monopoly. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, pages 66–72, Colchester, England, 2005.

[52] Robert Gibbons. *A Primer in Game Theory*. Prentice Hall, Upper Saddle River, NJ, 1992.

[53] D. B. Gillies, J. P. Mayberry, and J. von Neumann. *Two variants of poker*, volume 2, pages 13–50. Princeton University Press, Princeton, NJ, 1953.

[54] Matthew L. Ginsberg. GIB: Steps toward an expert-level bridge-playing program. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 584–589, 1999.

[55] Matthew L. Ginsberg. GIB: Imperfect information in a computationally challenging game. *Journal of Artificial Intelligence Research*, 14:313–368, 2002.

[56] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Boston, MA, 1989.

[57] D.E. Goldberg, K. Deb, and J.H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6(4):333–362, 1992.

[58] John J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.

[59] Harrah's. World Series of Poker. [Internet]. WWW address: "http://www.worldseriesofpoker.com" [Accessed 29 June 2007].

[60] Dan Harrington and Bill Robertie. *Harrington on Hold'em: Expert Strategy for No-Limit Tournaments; Volume I: Strategic Play*. Two Plus Two Publishing, Las Vegas, NV, 2004.

[61] Dan Harrington and Bill Robertie. *Harrington on Hold'em: Expert Strategy for No-Limit Tournaments; Volume II: The Endgame*. Two Plus Two Publishing, Las Vegas, NV, 2005.

[62] Dan Harrington and Bill Robertie. *Harrington on Hold'em: Expert Strategy for No-Limit Tournaments; Volume III: The Workbook*. Two Plus Two Publishing, Las Vegas, NV, 2006.

[63] Phil Hellmuth. *Play Poker Like the Pros*. Harper Collins Publishers, New York, NY, 2003.

[64] Matthew Hilger. *Internet Texas Hold'em*. Dimat Enterprises Inc., 2003.

[65] Bret Hoehn, Finnegan Southey, Robert C. Holte, and Valeriy Bulitko. Effective short-term opponent exploitation in simplified poker, 2005.

[66] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, England, 1998.

[67] John Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[68] Lee Jones. *Winning Low Limit Hold'em*. ConJelCo, Pittsburgh, PA, 2001.

[69] Gopal Kanji. *100 Statistical Tests*. SAGE Publications, London, England, 1999.

[70] Samuel Karlin and Rodrigo Restrepo. Multistage poker models. *Contributions to the Theory of Games III*, pages 337–363, 1957.

[71] Graham Kendall and Stephen Shaw. Investigation of an adaptive cribbage player. In *Computers and Games, Third International Conference*, pages 29–41, Edmonton, Canada, 2002.

[72] Graham Kendall and Craig Smith. The evolution of blackjack strategies. In *Proceedings of Congress on Evolutionary Computation*, volume 4, pages 2474–2481, Canberra, Australia, 2003.

[73] Graham Kendall and Glenn Whitwell. An evolutionary approach for the tuning of a chess evaluation function using population dynamics. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pages 995–1002, Seoul, Korea, 2001.

[74] Graham Kendall and Mark Willdig. An investigation of an adaptive poker player. In *14th Australian Joint Conference on Artificial Intelligence*, pages 189–200, Adelaide, Australia, 2001.

[75] Daphne Koller and Avi J. Pfeffer. Generating and solving imperfect information games. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1185–1192, Montreal, Canada, 1995.

[76] Daphne Koller and Avi J. Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215, 1997.

[77] Kevin B. Korb, Ann E. Nicholson, and Nathalie Jitnah. Bayesian poker. *Uncertainty in Artificial Intelligence*, pages 343–350, 1999.

[78] Koza. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Bradford Books, Cambridge, MA, 1992.

[79] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.

[80] Harold W. Kuhn. *A Simplified Two-Person Poker*, volume 1, pages 97–103. Princeton University Press, Princeton, NJ, 1950.

[81] Tom McEvoy. *Tournament Poker*. Cardoza Publishing, New York, NY, 2004.

[82] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, Germany, 1999.

[83] Ed Miller, David Sklansky, and Mason Malmuth. *Small Stakes Hold'em - Winning Big with Expert Play*. Two Plus Two Publishing, Las Vegas, NV, 2004.

[84] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 2002.

[85] David E. Moriarty and Risto Miikkulainen. Discovering complex Othello strategies through evolutionary neural networks. *Connection Science*, 7(3):195–209, 1995.

[86] Martin Müller. *Computer Go as a Sum of Local Games: An Application of Combinatorial Game Theory*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 1995.

[87] Martin Muller. Computer go 1984-2000. In T. Marsland and I. Frank, editors, *Computers and Games, Second International Conference*. Springer-Verlag, 2000.

[88] Martin Müller. Computer Go. *Artificial Intelligence*, 134(1):145–179, 2002.

[89] John F. Nash and Lloyd S. Shapley. *A Simple Three-Person Poker Game*, volume 1, pages 105–116. Princeton University Press, Princeton, NJ, 1950.

[90] Jason Noble. Finding robust texas hold'em poker strategies using pareto coevolution and deterministic crowding. In *Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA 02)*, pages 233–239, Las Vegas, NV, 2002.

[91] Jason Noble and Richard A. Watson. Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for pareto selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 493–500, San Francisco, CA, 2001.

[92] University of Alberta GAMES Group. Computer poker (Texas hold'em) Java source code, 2000. [Internet]. WWW address: "http://spaz.ca/aaron/poker/src/" [Accessed 27 March 2006].

[93] University of Pittsburg. International Conference on Cognitive Modelling, 2004. [Internet]. WWW address: “http://simon.lrdc.pitt.edu/ iccm/pokerbot.html” [Accessed 29 June 2007].

[94] Denis Papp. Dealing with imperfect information in poker. Master’s thesis, University of Alberta, Edmonton, Canada, 1998.

[95] Patrick Corr Paul Donnelly and Danny Crookes. Evolving Go playing strategy in neural networks. In *AISB Workshop on Evolutionary Computing*, Leeds, England, 1994.

[96] Two Plus Two Publishing. [Internet]. WWW address: “http://www.twoplustwo.com” [Accessed 29 June 2007].

[97] S. Rana, L.D. Whitley, and R. Cogswell. Searching in the presence of noise. *Parallel Problem Solving from Nature (Lecture Notes in Computer Science 1141)*, pages 198–207, 1996.

[98] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fromman-Holzboog, Stuttgart, Germany, 1973.

[99] Stewart Reuben and Bob Ciaffone. *Pot-Limit and No-Limit Poker*. Bob Ciaffone, 1997.

[100] Norman Richards, David E. Moriarty, and Risto Miikkulainen. Evolving neural networks to play Go. *Applied Intelligence*, 8:85–96, 1998.

[101] Blair Rodman and Lee Nelson. *Kill Phil - The Fast Track to Success in No-Limit Hold’em Poker Tournaments*. Huntington Press, Las Vegas, NV, 2005.

[102] Stuart Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 1995.

[103] Jonathan Schaeffer. *One Jump Ahead*. Springer-Verlag, New York, MA, 1997.

[104] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhaeuser, Stuttgart, Germany, 1977.

[105] New Scientist. *Raise you 50...*, volume 20/27 December 2003 / 3 January 2004, pages 64–67. Reed Business Information Ltd, London, England, 2003.

[106] Alex Selby. Optimal heads-up preflop holdem, 1999. [Internet]. WWW address: “http://www.archduke.demon.co.uk/simplex/index.html” [Accessed 29 June 2007].

[107] Jiefu Shi and Michael L. Littman. Abstraction models for game theoretic poker. In *Computers and Games, Second International Conference, CG 2000,* pages 333–345, Hamamatsu, Japan, 2000. Springer-Verlag.

[108] David Sklansky. *Hold’em Poker*. Two Plus Two Publishing, Las Vegas, NV, 2001.

[109] David Sklansky. *The Theory of Poker*. Two Plus Two Publishing, Las Vegas, NV, 2001.

[110] David Sklansky. *Tournament Poker for Advanced Players, Second Edition*. Two Plus Two Publishing, Las Vegas, NV, 2002.

[111] David Sklansky and Mason Malmuth. *Hold’em Poker For Advanced Players*. Two Plus Two Publishing, Las Vegas, NV, 2001.

[112] David Sklansky and Ed Miller. *No Limit Hold’em: Theory and Practice*. Two Plus Two Publishing, Las Vegas, NV, 2006.

[113] John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, England, 1982.

[114] Shane Smith. *Poker Tournament Tips from the Pros*. Cardoza Publishing, New York, NY, 2003.

[115] Steven F. Smith. *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, 1980.

[116] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes’ bluff: Opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 550–558, Edinburgh, Scotland, 2005.

[117] Sylvester Suzuki. *Poker Tournament Strategies*. Two Plus Two Publishing, Las Vegas, NV, 1998.

[118] Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[119] Steven Vajda. *Introduction to Linear Programming and the Theory of Games*. Wiley, New York, NY, 1960.

[120] John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, NJ, 1953.

[121] Ken Warren. *Winner's Guide to Texas Hold'em Poker*. Cardoza Publishing, New York, NY, 1996.

[122] Wikipedia. Poker. [Internet]. WWW address: "http://en.wikipedia.org/wiki/Poker" [Accessed 29 June 2007].

[123] Wikipedia. Statistical hypothesis testing. [Internet]. WWW address: "http://en.wikipedia.org/wiki/Statistical_hypothesis_testing" [Accessed 29 June 2007].