KNOWLEDGE VS. POWER IN THE GAME OF RIKKEN

Viktor Vorsteveld

Master Thesis MICC-IKAT 07-02

Thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science of Knowledge Engineering
at the Faculty of Humanities and Sciences
of Universiteit Maastricht

Thesis committee:

Prof. dr. H.J. van den Herik
Prof. dr. E.O. Postma
Dr. ir. J.W.H.M. Uiterwijk
Dr. H.H.L.M. Donkers

Universiteit Maastricht
Maastricht ICT Competence Centre
Institute for Knowledge and Agent Technology
Maastricht, The Netherlands
April 2007

Preface

In my Master Thesis I present the research performed at MICC-IKAT of the Faculty of Humanities and Sciences at the Universiteit Maastricht. The goal of this research is to discover which AI technique we should use to create the best playing agent for the game Rikken. To test several AI techniques I created the program RikMaster. What was the reason for me to choose a card game in this field of research?

When I was just a little boy my father taught me how to play Klaverjassen and when we were at my grandparents' place, we would always play the game together with my uncle and grandfather. We started playing after diner and stopped somewhere past midnight. When I grew older they taught me how to play Pandoeren, which is quite similar to Rikken, and we played that game for hours at a time. So, you might say that playing card games has always been part of my education. What I did not know back then, is that my interest in card games would result into this research and would be instrumental to make the final step towards my graduation.

I wish to thank several people for helping me performing this research and writing this thesis. First, I would like to thank prof.dr. H.J. van den Herik for finding the time to read and correct my thesis on such a short notice. Second, I would like to thank my daily advisor dr.ir. J.W.H.M. Uiterwijk for all the ideas, advice, and feedback he gave me during this research. Both also helped to arouse my interest for computer-game playing in their course Intelligent Search Techniques. The assignment to create an AI player for the game Fanorona inspired me to do this research. The other committee members are also recognised for their time and effort in reading this thesis. Futhermore, I would like to thank all of my fellow students, friends and family for their love and support.

Viktor Vorsteveld
Maastricht, April 2007

Abstract

In this thesis we will describe research that has been done in the field of the game Rikken. The goal of this research is to find an answer to the question if we can determine whether rule-based techniques are more suitable than Monte-Carlo techniques to create an effectively and efficiently playing AI player in the game of Rikken.

Rikken is a card game that exists of 2 phases, the bidding round and the playing round. In the bidding round an auction is held to decide which game type will be played during the playing round. There are four game types which each have their unique goal. The goal is always to win a certain (minimum) amount of tricks.

In order to test the techniques we created a simulation test environment, RikMaster. In this environment we created AI players that make use of rule-based or Monte-Carlo techniques and tested their performances on efficiency and strength. Each of the players needed a strategy for the bidding round and one for the playing round.

Four players were created, a random player, two rule-based AI players, and a Monte-Carlo player. The goal of the random player is to be a good comparison for the other players and is used by the Monte-Carlo player in simulations. Two rule-based players were created, the basic AI player and the advanced AI player. The goal of both players is to test rule-based AI but with different approaches. Each of them has a different bidding method that they use to evaluate their hands in the bidding round. The basic AI player uses basic rules that create a strategy based on the cards of the basic AI player and the cards already played in the current trick. The strategy of the advanced AI player is based on (hidden) information that becomes available during the playing round combined with the knowledge of his own cards. The advanced AI player keeps track of the cards that other players play and which suits they possess or not possess. The basic AI player is designed like a reactive player and the advanced AI player like a knowledge-based player. The fourth player is the Monte-Carlo player. This player uses simulations to decide what game type to choose during the bidding round and which cards to play during the playing round.

The experiments that we ran have three goals. The goal of the first series of experiments was to find out how accurate the bidding methods could predict the number of tricks that were won during the real play. The goal of the second series of experiments was to compare the strength of the players during the playing round. The goal of the third series of experiments was to investigate the overall performance of the four players in a tournament combining their behaviour in the bidding and the playing round.

From the experiments we learned that the advanced AI player, driven by rule-based techniques, had the best bidding method, the strongest play, and won the tournament. The Monte-Carlo player had a better performance in Rik and Solo 8, but the advanced AI player is the strongest player overall. Therefore we may conclude that the rule-based techniques, used by the advanced AI player, are more suitable to create an effectively and efficiently playing AI player in the game of Rikken.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis deals with comparing rule-based techniques with Monte-Carlo techniques in the game 'Rikken'. Four artificial intelligent players are developed and tested on their performance. In Section 1.1 an introduction to artificial intelligence (AI) in games is given. Section 1.2 gives an introduction to the game 'Rikken'. The problem statement and research questions are formulated in Section 1.3 and the outline of this thesis can be found in Section 1.4.

1.1 Game AI

With the development of the computer, halfway the 20th century, a new medium for games was born. All sorts of games that were played on a board or with a deck of cards could be translated to the computer screen. But instead of using the computer only as an interface to play a game, the computer could also participate in the game as a player. To act as a player, the computer required a certain behaviour. The first traces of such a behaviour can be found in the arcade hall. Arcade games like Pong and Pacman used certain techniques to create games challenging enough for people to play and pay for (Rabin, 2002). These techniques existed of simple rules combined with an amount of randomness to create nonpredictable behaviour. This was the beginning of a new domain called game AI. One could define game AI as follows.

> *Game AI refers to techniques used in computer and video games to produce the illusion of intelligence in the behavior of non-player characters. The techniques used typically draw upon existing methods from the academic field of Artificial Intelligence (AI). However, the term game AI is often used to refer to a broad set of algorithms that also includes techniques from control theory, robotics, computer graphics and computer science in general* (Wikipedia, 2007).

Not all techniques obtained from game AI are useful for all types of games. Games can be categorized into (1) perfect/imperfect-information games and (2) stochastic/deterministic games. A player has perfect information if (1) he[1] knows all the actions available to each player and (2) all the potential outcomes for each player. If the player does not comply to these conditions, he has imperfect information (Gibbons, 1992). Perfect information is equivalent to knowing enough to construct the entire game tree. Perfect-information games are not common in the real world, and are usually used only as approximations of the actual game being played. Stochastic games are games with an amount of randomness, like throwing dice or the drawing of cards. Chess is an example of a deterministic, perfect-information game, Backgammon is a game with stochastic properties and the card game Bridge is an example of a deterministic, imperfect-information game.

The techniques developed in game AI are often transferable to other domains. It can for example be used in military (Urlings, 2006), management or economic simulations (Biethahn and Nissen, 1994) to test human-like intelligence or have educational value in software for children (Bonnet *et al.*, 1990).

[1] In this thesis we use, for readability, 'he' and 'his' as an equivalent for 'he or she' and 'his or her'.

1.2 The game Rikken

Rikken is a card game of which the origin can be found in Brabant, the Netherlands. It is slightly related to 'Klaverjassen' (Sport, 2004) and 'Bridge' (Sint and SchipperHeyn, 1994) and is still popular among the people in Brabant. The game can be characterized as a deterministic, imperfect-information game. The only stochastic element is the drawing of the cards which makes the game an imperfect-information game. Rikken can be seen as a collection of games which follow the same rules but with different goals and characteristics. These games will be referred to as game types. The goal of each of the game types is to win a certain (sometimes minimal) number of tricks. Other properties of game types include the involvement of trump, playing with or without a partner and the reward for winning a game with that game type. The game exist of a bidding round and a playing round. The bidding round is to determine the game type that will be played in the playing round. After the playing round, the players are rewarded for their performance based on the properties of the game type. The goal of the game is maximizing the total reward. For the complete description of the game, we refer to Section 2.1.

1.3 Problem statement and research questions

In this research the game of Rikken is implemented in order to develop AI players. These AI players make use of two different AI techniques to make actions in the game. Rule-based AI techniques are used in multiple domains (Ligeza, 2006), which include the game AI domain (Cavazza, 2000). AI players with rule-based techniques make moves or actions based on a set of rules. The rules used to decide which move or action will be played depend on the situation the player is in. The Monte-Carlo technique is also used in different domains (Andrieu, 2003) including the game AI domain. This technique uses statistics to give a prediction of the most successful move or action. Monte-Carlo is a commonly used technique when dealing with imperfect information (Ginsberg, 1999). A problem statement is formulated to give structure to the research:

> *Can we determine whether rule-based techniques are more suitable than Monte-Carlo techniques to create an effectively and efficiently playing AI player in the game of Rikken?*

In order to answer the problem statement, four research questions are formulated:

> *RQ 1: How can rule-based techniques be compared to Monte-Carlo techniques in the game of Rikken?*

The game exists of two phases where the players can have influence to the outcome of the game. During the first phase, the bidding round, players decide on the game type that will be played in the second phase, the playing round.

The second research question is:

> *RQ 2: How do the performances of the techniques in the bidding phase compare?*

The third research question is:

> *RQ 3: How do the performances of the techniques in the playing phase compare?*

The winner of the game is the player who won the most chips at the end of the game. To become the winner you need a strong combination of bidding and playing.

The last research question is:

> *RQ 4: How do the performances of the techniques in both phases combined compare?*

The goal of this research is to compare the developed players in order to find out which technique was most successful in the game of Rikken.

1.4 Outline of the thesis

The outline of this thesis is as follows:

- Chapter 1 contains a general introduction to computer games followed by a short description of the game Rikken. Furthermore a problem statement and four research questions are formulated.
- Chapter 2 explains the rules and options of Rikken. In Section 2.1 the two phases of the game, the bidding round and the playing round, are described. Section 2.2 describes in which way the cards are divided among the players.
- Chapter 3 describes the four AI players developed for the game. The development of the four players constitutes an answer to *RQ 1*. Each player has his own bidding method and tactics for the different game types.
- Chapter 4 describes the set-up (Section 4.1) and results of the experiments done with the different AI players (Section 4.2). In Subsections 4.1.1 and 4.2.1 the bidding methods are analyzed in order to answer *RQ 2*. In Subsections 4.1.2 and 4.2.2 the players' strength with regard to the tactics in the playing round is examined. The goal of these experiments is to find an anser to *RQ 3*. In order to give answer to *RQ 4* a tournament is held which can be found in Subsections 4.1.3 and 4.2.3. In the tournament we determine the players' overall strength by combining the bidding round and the playing round.
- Chapter 5 gives an interpretation to the results of the experiments and it gives the final answers to *RQ 2*, *RQ 3*, and *RQ 4*.
- Chapter 6 contains the conclusions of this research. Section 6.1 answers the problem statement and the four research questions formulated in Chapter 1. In Section 6.2 we describe possibilities for future research.

Chapter 2

Rikken

In this chapter, the game of Rikken is explained in detail. In Section 2.1 we describe the possibilities in each phase of the game. The differences between the various game types and the way a game type is chosen is further explained.

In Section 2.2 we describe the shuffling of the cards before dividing them among the players. A way to generate interesting decks for the game is discussed.

2.1 Rules and options

In this section the course of the game is explained. This course exists of four stages. The first stage concerns the players that participate in the game and is described in Subsection 2.1.1. The second stage, Subsection 2.1.2, describes the distribution of the cards among the players. The third stage, Subsection 2.1.3, describes the bidding round. In this round an auction is held to decide which game type will be played in the playing round. The different goals and characteristics of the game types are explained in detail. The last stage, the playing round, is described in Subsection 2.1.4. The rules that come into play during the playing round are explained.

2.1.1 Players

Rikken is normally played by four players. Although players form temporary partnerships, one against three or two against two, they ultimately all play for themselves. The player that placed the highest bid during the auction is the winner of the bidding round. This player is referred to as the declarer. For some game types, the declarer is required to play with a partner and form a partnership. This partnership is referred to as the declaring side. The other players are referred to as the defenders or the defending side. The position of the players in a partnership can play a role in the outcome of the game. When for example the first and the last player are partners, this could be an advantage. When the first player is to throw the first card, the last player is his back-up because he has the last saying over that round.

2.1.2 Dealing

Before the game begins, the cards need to be dealt among the players. In the first game, the cards are shuffled, cut by the player to the dealer's right, and divided. Every player first receives seven cards followed by six cards to complete the hand. After every game the cards are collected in the order of play. This deck is then cut and dealt in the same way as above, first 7 followed by 6 cards. Cards must not be shuffled again! This dealing method assures interesting decks because during the play, suits get together and certain combinations are formed. After every game, the player left from the dealer becomes the new dealer.

2.1.3 Bidding

In the bidding round the game type is decided upon, and, if applicable, trump and partner are decided too. To decide which game type will be played during the playing round, an auction is held. Each game type represents a bid and the highest bid will be played. The bidding order can be found in Table 2.1.[1]

The research in this thesis is restricted to four of the game types viz. Rik, Solo 8, Piek, and Misère. These game types can be seen as the main game types and the other game types are more or less extensions of the main game types. Although the other game types would give the players more choice during the bidding round, they require the same tactics and stategies as the main game types during the playing round. Including al the game types would also enlarge the research and therefore we chose not the include these game types.

Table 2.1 also contains the number of tricks required to win that game type (goal), whether you play with or without trump and or partner, and the reward in chips when winning. For example when playing Solo 8, you need 8 tricks to win the game type, you play with trump but without a partner, and if you win 8 tricks or more, you recieve 1 chip from every player. When you lose you have to pay every player 1 chip, so the losers always pay the winners.

The Bid	Number of tricks	Partner	Trump	Reward in chips
Pas (fold)	—	—	—	0
Rik	8	with	with	1
Rik beter (Rik better)	8	with	with(hearts)	1
8 alleen (Solo 8)	8	without	with	1
Troela	8	with	with	2
Piek (Piccolo)	1	without	without	3
9 alleen (Solo 9)	9	without	with	4
Misère	0	without	without	5
10 alleen (Solo 10)	10	without	with	6
Open Piek [2]	1	without	without	6
11 alleen (Solo 11)	11	without	with	8
Open Piek+[2][3]	1	without	without	10
Open Misère[2]	0	without	without	10
12 alleen (Solo 12)	12	without	with	12
Open Misère+[2][3]	0	without	without	15
Solo	13	without	with	22
Schoppen Mie[4]	—	without	without	2

Table 2.1: The game types of the game.

The player left to the dealer makes the first bid. Each player may pass or bid a contract. If a player bids, the subsequent players can either pass or bid on a higher contract. A player who has passed is not allowed to bid again in the auction. The bidding continues around the table as many times as necessary for all players who have not yet passed. The contract is settled when three players have passed. This makes the player with the last and highest bid the declarer.

After the auction, depending on the outcome of the game type, the declarer has to decide on trump and partner. Any of the four suits can be chosen to be trump suit except for 'Rik beter' in which trump is always hearts.

The partnerships are determined by the declarer calling an ace, the holder of which becomes declarer's partner. For example the declarer says 'hearts are trumps and the ace of spades is my partner'. The declarer can pick any ace that is not trump suit or in his possession. The called ace must if possible be

[1] More details about the different game types of the game can be found in Subsection 2.1.4.
[2] "Open" means the declarer needs to reveal his cards in some point of the game. See Section 2.2
[3] + = "met praatje" and means players are allowed to discuss in order to find a winning strategy.
[4] Schoppen Mie can only be played when the first 3 players have folded.

in a suit of which the declarer holds at least one card. If the declarer has no suit without the ace, then the declarer can call the ace of a void suit, but must announce that the ace is being called 'blind'. If a player is in possession of all of the Aces, a King is called instead of an ace.

2.1.4 Playing

The player left to the dealer always leads to the first trick, except for the game types Open Piek (with discussion), Open Misère with discussion and Solo. In the latter three cases the declarer starts the game. Any card may be led to a trick but the other three players must follow suit. Players unable to follow suit may play any card. When trump is played, it is allowed to play a lower trump. The sequence of the cards ranking from high to low: Ace - King - Queen - Jack - 10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2. This includes the sequence for trump suit (in other card games this sequence is often different). If any trumps are played to a trick, it is won by the highest trump played. If there are no trumps in a trick, it is won by the highest card played in the suit that was led. The winner of a trick leads to the next trick.

The holder of the called ace is not obligated to play his ace when the suit of the called ace is led.

If the requirements for the game type are met at the end of the game, the declarer gets paid by the defenders. If the declarer had a partner during the game then his partner will also be paid by the defenders. If the requirements are not met, the defenders get paid by the declaring side. The player with the most chips at the end of the whole game is the winner.

An overview of the game types is given in Table 2.2.

Pas (fold)	These cards are not good enough to make a higher bid than the current.
Rik	Goal is to win at least 8 tricks. Declarer decides on trump and calls an ace to determine on the partnership.
Rik beter(rik better)	Same goal and options that apply to Rik, except that trump is always hearts.
8 alleen (solo 8)	Goal is to win at least 8 tricks without a partner. Declarer decides on trump.
Troela	Goal is to win at least 8 tricks. This game type is only possible when the declarer is in possession of three aces. The player in possession of the remaining ace becomes his partner. The partner decides on trump under the condition that trump suit is not the same as his ace's suit.[4]
Piek (piccolo)	Goal is to win one and only one trick without trump.
9 alleen (solo 9)	Goal is to win at least 9 tricks without a partner. Declarer decides on trump.
Misère	Goal is to refrain from winning any trick. No trump involved.
10 alleen (solo 10)	Goal is to win at least 10 tricks without a partner. Declarer decides on trump.
Open piek	Same goal and options that apply to Piek but after the first card in the second round the declarer has to reveal his cards by putting them on the table.
11 alleen (solo 11)	Goal is to win at least 11 tricks without a partner. Declarer decides on trump.
Open piek +	Same goal and options that apply to Open Piek but the players are allowed to talk about their strategy.
Open misère	Same goals and options that apply to Misère but after the first card in the second round the declarer has to reveal his cards by putting them on the table.
12 alleen (solo 12)	Goal is to win at least 12 tricks without a partner. Declarer decides on trump.
Open misère +	Same goal and options that apply to Open Misère but the players are allowed to talk about their strategy.
Solo:	Goal is to win all tricks without partner. Declarer decides on trump.
Schoppen Mie (Queen of Spades)	Goal is to not win the queen of spades nor the last trick. Queen of spades has to be played immediately when his owner cannot follow suit. It is not allowed to play spades until the queen of spades has been played unless the player only got spades left.

Table 2.2: The rules of the game types.

[4] In some versions of the game, the rules obligate the players to play Troela when the game type is available.

2.2 Card shuffling

In this Section two techniques to shuffle and divide a deck of cards are described. Subsection 2.2.1 describes the technique to generate random decks with equal chance. Subsection 2.2.2 describes a technique to generate decks that are interesting for the game Rikken.

2.2.1 Random decks

Before the cards can be evaluated by the bidding program, the cards need to be divided among the players. The set of divided cards is called a game instance. The problem that arises is to create each game instance with equal chance. The algorithm used to meet this requirement is known as the Knuth shuffle (Knuth, 1998). The algorithm moves from top to bottom in a pack of cards, each time swapping the current card with a random position from the pack that has not been passed through, including the current position. This algorithm produces the $n!$ permutations on a pack of n cards in a way that each game instance is created with equal chance. If for instance the algorithm would swap a card with any card from the deck, this would result in n^n possibilities. $n^n > n!$ so some game instances would be picked more than others resulting in unequal chances.

2.2.2 Rik decks

The previous subsection has learned how to create random game instances. This is however not the same way that the game is originally played. The first game is indeed random divided as described above. When playing, suits get together and certain combinations are formed. After each game the cards are collected in the order of play. The deck is then cut and divided, first 7 cards for each player followed by 6 each. This dealing method creates interesting new hands with a higher possibility of decks that for example contain 7 cards of the same suit or decks without a certain suit.

The dealing method works fine for human players but it is a different story for a computer player. When a deck is cut, the order of cards remains the same no matter how often the deck is cut. The only thing that changes is the beginning of the order. A computer player that knows in which order the cards are played in the previous game, can easily calculate the other players' cards in the next game. The cards remain in the same order after the cut and when the computer player receives his own cards, he can then calculate the position in the order. When the computer player also knows in what order the cards are dealt among the players, all the players' hands can be calculated, making it a perfect-information game. In order to prevent this from happening, a slightly different method is used. Instead of collecting all the cards one by one, the cards are grouped by their round creating 13 groups of 4 cards. These groups are randomly shuffled, in the same way as explained in Subsection 2.2.1. Hereafter the same method is used, so cutting the deck and dividing the cards. So, the combinations formed during the play remain, but the possibility to determine the players' cards by calculation is minimized.

Chapter 3

AI Players

In this chapter we describe the four players that were developed during this research and we answer the first research question (*RQ 1*). In Section 3.1 we introduce the 'Random Player', who plays the game in a random fashion. Sections 3.2 and 3.3 cover the 'Basic AI Player' and the 'Advanced AI Player' respectively. Both players use rule-based techniques in order to play the game. The basic AI player makes an action based on the current situation whereas the advanced AI player also incorporates information that becomes available during gameplay. In Section 3.4 we introduce the 'Monte-Carlo Player' who uses Monte-Carlo techniques to play the game. For each of the players we describe their behaviour during the bidding and playing round.

3.1 Random player

The AI player with the most basic strategy is that of the random player. This player is able to play the game by the rules but only in a random fashion. The goal for this player in this research is twofold. First, the performance of the player is used as a measure for the other players. Second, the Monte-Carlo player uses random players for his simulations (Section 3.4).

3.1.1 Bidding

In the bidding round, an auction is held to decide on the game type that will be played. The random player chooses a contract randomly from all possible contracts of the auction. If this contract is not higher than the current bid, the player folds.

The random player becomes the declarer if his bid was the highest contract. Depending on the game type, the random player needs to decide on trump and/or ace. The random player picks a suit randomly to become trump. The ace is called randomly as well but from the remaining legal possibilities.

3.1.2 Playing

The strategy of the random player is the same for every game type. First it is decided which of the cards in his possession are legal to play according to the rules. It is for example obligated to follow suit when possible. From the selection of possible cards, one card is chosen randomly and played.

3.2 Basic AI player

The basic AI player is a rule-based player. Rule-based systems are often referred to as expert systems, because they cover a specific limited domain. An example of a general rule-based system is Cyc.[1] The Cyc knowledge base is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. In our case an expert system in the domain of Rikken is desirable.

[1] www.cyc.com

3.2.1 Bidding

The bidding round is played to determine which game type will be played. In order to make a reasonable bid, an evaluation of the cards is necessary. We developed a method, the 'Trick Procent Count', to determine the strength of a hand of cards. The idea behind the method is to determine a chance between zero and one for every card in the hand. These chances represent the chance of winning a trick and when added up, they give a prediction of the maximum number of tricks this hand can make.

When you possess the highest card of a suit, the ace, and trump is not involved, then this card will win the trick when that suit is led to the trick. If you possess the second highest card, the king, but you do not possess the ace, then the ace has to be eliminated before the king is sure to win a trick. You need at least one other card besides the king to cover the king when the ace is played. The same goes for the rest of the cards in the suit. Winning a trick with a particular card depends on 3 factors, the height of that card (Ace, King, etc), the total number of cards in that suit in your possession and the number of cards in that suit not in your possession with a higher value than the card. The chance of winning a trick is given by the following formula:

$$\frac{\#cardsinsuit - \#cardshigher}{\#cardsinsuit}$$

All chances are between zero and one so all negative chances are set to zero. An example of the calculation of a hand of cards is given in Table 3.1.

Cards	Percentage
♠ A	$(4-0) \div 4 = 1.00$
♠ K	$(4-0) \div 4 = 1.00$
♠ J	$(4-1) \div 4 = 0.75$
♠ 2	$(4-9) \div 4 = 0.00$
♡ Q	$(1-2) \div 1 = 0.00$
♢ K	$(3-1) \div 3 = 0.66$
♢ J	$(3-2) \div 3 = 0.33$
♢ 4	$(3-8) \div 3 = 0.00$
♣ Q	$(5-2) \div 5 = 0.60$
♣ J	$(5-2) \div 5 = 0.60$
♣ 10	$(5-2) \div 5 = 0.60$
♣ 8	$(5-3) \div 5 = 0.40$
♣ 6	$(5-4) \div 5 = 0.20$
Total # tricks	6.15

Table 3.1: An example of the calculation of a hand of cards using TPC.

The conditions for the basic AI player to bid on a game type will be (1) based on the conditions to win that game type, see Subsection 2.1.3 and (2) the results of the experiments to determine the accuracy of the bid method, see Subsection 4.2.1. For example the minimal condition to win Solo 8 is to win 8 tricks. The initial condition for the basic AI player will be the prediction of 8 winning cards produced by the TPC method. When the experiments indicate that on average the prediction over- or underestimates the number of tricks that are made in the real game, the condition is adjusted to correct the estimation error. Each bidding method will have its own conditions for choosing a game type during the bidding round.

The procedure to decide on trump (and ace) is based on a point system. The strength of a suit is determined by the points of the cards plus the length of the suit. Each of the cards ranging from 2 to 10 is worth 1 point, Jack is 2, Queen is 3, King is 4 and Ace is 5 points. The suit with the maximum number of points will be trump suit. An example of the calculation is given in Table 3.2. The suit to call the ace is the one that has the maximum number of points after removing trump suit and suits which contain aces.

cards	points	length	
♠ A	5		
♠ K	4		
♠ J	2		
♠ 2	1		
Total	12 +	4 =	16

Table 3.2: The calculation needed to determine on trump or Ace (partner).

3.2.2 Playing

The strategy of the basic AI player is defined by certain rules. These rules depend on the current position in the playing round and the game type.

The rules that apply when the game type is Rik or Solo 8:

1. When you are first to play a card, the following rules apply:

   if you possess one or more cards that are the current highest of their suit **then**
   play a randomly chosen highest card
   else
   play a randomly chosen card
   end if

2. If you are second or third to play a card, the following rules apply:

   if you possess (a) card(s) of the suit asked **then**
   if you possess the highest card of the suit asked **then**
   play the highest card of the suit asked
   else
   if you possess a card of the suit asked that is higher than the cards already played **then**
   play a card randomly from the suit asked
   else
   play the lowest card of the suit asked
   end if
   end if
   else
   if you possess (a) card(s) of trump suit **then**
   play the lowest card of your trump suit
   else
   collect the lowest card of every suit in your possession and pick a random card from that selection. If the selected card is the current highest card of that suit in the game pick another card from the selection that is not the current highest card of the game. If either all the cards from the selection are currently the highest or the selection exist of a single card play a randomly chosen card from the selection
   end if
   end if

3. If you are last to play a card, the following rules apply:

   if you possess (a) card(s) of the suit asked **then**
   if you can win the trick with a card of the suit asked **then**
   play the lowest card of the suit asked that wins the trick
   else
   play the lowest card of the suit asked
   end if

else
  if you possess the trump suit and are able to win the trick with a trump card **then**
    play the lowest trump that wins the trick
  else
    collect the lowest card of every non-trump suit in your possession and pick a random card from that selection. If the selected card is the current highest card of that suit in the game pick another card from the selection that is not the current highest card of the game. If either all the cards from the selection are currently the highest or the selection exist of a single card play a randomly chosen card from the selection
  end if
end if

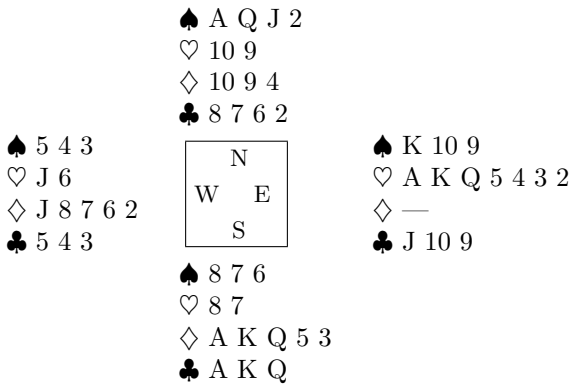


Figure 3.1: Explanation of rule 2 of Rik/Solo 8.

As an explanation of rule 2 above, consider the example of Figure 3.1. Player North is the declarer, his partner is player South, the game type is Rik, player East is a basic AI player, and North plays the Queen of spades to lead to the trick. East now has the choice to play the King of spades but this is a dangerous move because he knows the highest of spades (Ace) has not yet been played. By playing a random spade, East and partner sometimes win the trick: play King when the Ace is in possession of North or West, or play a lower spade when West is in possession of the Ace. They will also sometimes lose: when South is in possession of the Ace or when a lower spade than the Queen is played when West is not in possession of the Ace. When the King is not played and somebody plays the Ace in the same trick, East is sure to have the highest card of spades. This is the idea behind the second rule of the basic AI player applied to Rik and Solo 8.

The rules that apply when the game type is Piek:

if you am declarer **then**
  if you won zero tricks so far **then**
    play the highest card of the suit asked or when you have to lead to the trick, pick the highest card of every suit in your possession and pick one card randomly from that selection
  else
    play by the rules which apply to the game type of Misère
  end if
else
  if the declarer won zero tricks so far **then**
    make a selection of the legal possible cards and chose a card randomly from that selection
  else
    play by the rules which apply to the game type of Misère
  end if

end if

The rules that apply when the game type is Misère:

1. When you are first to play: play your lowest card of a random suit.

2. If you are not first to play:

if you am the declarer **then**
  if you possess cards of the suit asked **then**
    if you can play a card with a lower value than the highest card of the current trick **then**
      play the highest card that has a lower value than the highest card of the current trick
    else
      play the lowest card of that suit in your possession
    end if
  else
    play the highest card of a random suit
  end if
else
  if you can follow suit **then**
    if the declarer played a card in this trick **then**
      if the declarer currently wins the trick **then**
        if you can play a card with a lower value than the highest card of the current trick **then**
          play the highest card that has a lower value than the highest card of the current trick
        else
          play the lowest card of that suit in your possession
        end if
      else
        play the highest card of the suit asked
      end if
    else
      if you have a card with a lower value than the highest card of the current trick **then**
        play the highest card with a lower value than the highest card of the current trick
      else
        play the lowest card of that suit in your possession
      end if
    end if
  else
    make a selection of the highest cards of each suit in your possession and pick a card randomly from that selection
  end if
end if

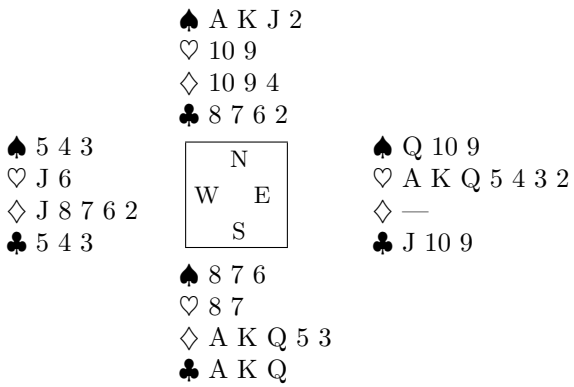
3.3 Advanced AI player

The advanced AI player is also a rule-based player. This player differs from the basic AI player in (1) the bidding method, (2) obtaining information, and (3) playing rules. The bidding method that is used by the advanced AI player to evaluate the strength of a hand of cards is described in Subsection 3.3.1. The playing rules are explained in Subsection 3.3.2. These rules make use of (hidden) information that is obtained by logic reasoning. For each of the players the advanced AI player keeps track of which suits they do or do not possess. By combining this information with the knowledge about the cards already played the advanced AI player can extract hidden information from certain situations and update his information set about the players. This is possible in the following situations,

1. You possess x cards of a certain suit and y cards of that suit are already played and $x + y = 13$. Then you know nobody has that suit except you.
2. Three players cannot follow suit. x cards are played, thus the player that followed suit has got $13 - x$ cards left of that suit. All the players are able to obtain this information. When the number of remaining rounds is also $13 - x$, all the players know that the player with that suit only got that suit left.
3. Two players cannot follow suit, x cards of that suit are already played and you possess y cards of that suit. The other player possesses $13 - x - y$ cards of that suit. Only you and the other player with suit know this information. Other players only know you may share that suit.
4. All 13 cards of a suit are played. None of the players possess this suit anymore. Everybody knows.

These situations will be clarified in the following examples:

Situation: 1st round, player North starts the round, no real game type but the goal is to maximise tricks, without trump.



Round	Player North	Player East	Player South	Player West	Winner round
1	♠ A	♠ 9	♠ 6	♠ 3	North
2	♠ K	♠ 10	♠ 7	♠ 4	North
3	♠ J	♠ Q	♠ 8	♠ 5	East

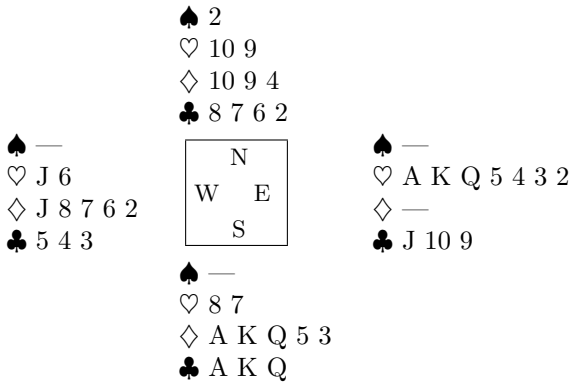
Figure 3.2: Hidden information: example of situation 1.

In Figure 3.2 three rounds where spades led to the trick were played and all the players were able to follow suit. In total twelve spades were played, so only one remains. The one player with the last spade (in this case North) knows the other players cannot follow spades. The other players only know one of the players has a spade left but not who. When North plays the last spade, the other players will know nobody has spades left.

In Figure 3.3 the game continues and East played three rounds of hearts. The first two rounds everybody followed suit but the third round nobody was able to follow suit. Now every player knows that the remaining four cards of hearts are also in possession of East. (In the case this happens when only four rounds are left to play, all the players automatically know East only possesses hearts.)

Figure 3.4 shows that East plays clubs and the trick goes to South. Next South plays three rounds of diamonds. After the first round, only East was not able to follow suit. The second round gives the same information but in the third round, only west is able to follow suit. In this case South and West share the remaining diamonds. Only South and West know they share the two diamonds. North and East only know two diamonds are left but are unaware of the distribution over South and West.

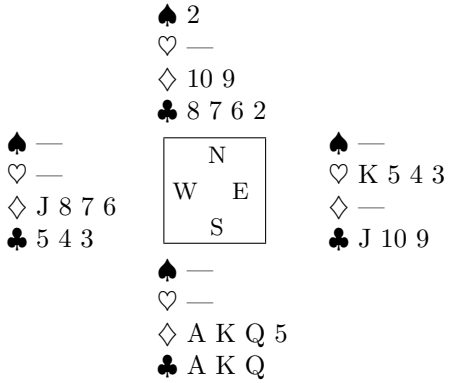
Situation: 4th round, player East starts the round.



Round	Player North	Player East	Player South	Player West	Winner round
4	♡ 9	♡ A	♡ 7	♡ 6	East
5	♡ 10	♡ Q	♡ 8	♡ J	East
6	♢ 4	♡ 2	♢ 3	♢ 2	East

Figure 3.3: Hidden information: example of situation 2.

Situation: 7th round, player East starts the round.

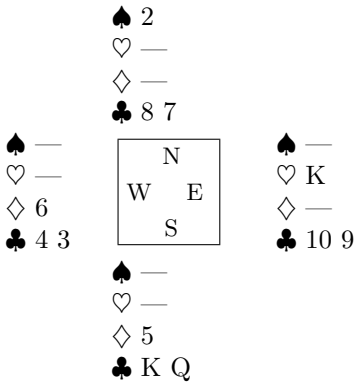


Round	Player North	Player East	Player South	Player West	Winner round
7	♣ 2	♣ J	♣ A	♣ 3	South
8	♢ 9	♡ 3	♢ A	♢ 6	South
9	♢ 10	♡ 4	♢ K	♢ 7	South
10	♣ 6	♡ 5	♢ Q	♢ 8	South

Figure 3.4: Hidden information: example of situation 3.

Figure 3.5 shows that after round twelve all thirteen cards of clubs are played. All the players know that none of the players can possess a club anymore. The rest of the game will not provide interesting situations.

Situation: 11th round, player South starts the round.



Round	Player North	Player East	Player South	Player West	Winner round
11	♣ 7	♣ 9	♣ K	♣ 3	South
12	♣ 8	♣ 10	♣ Q	♣ 4	South

Figure 3.5: Hidden information: example of situation 4.

The rules that determine the strategy make use of the information that becomes available during the play. We will discuss those rules in Subsection 3.3.2. First we will discuss the bidding method of the advanced AI player in Subsection 3.3.1.

3.3.1 Bidding

The advanced AI player combines two techniques to determine the best bid. The first technique is based on the Losing Trick Count technique (Cowan, 1987). The goal of LTC is to determine the maximum number of winning tricks of a hand of cards and is used to evaluate hands for Rik and Solo 8. The idea is to count the number of cards that will most likely lose a trick. Every suit can have at most three losers, an Ace will never be a loser, nor will a King in a suit of two cards or more, nor a Queen in a suit of three cards or more and the x represents the cards other than already mentioned in the following rules.

- No cards in suit: 0 losing tricks
- One card in suit: A = 0 losing tricks, x = 1 losing trick
- Two cards in suit: AK = 0 losing tricks, Ax, Kx = 1 losing trick, xx = 2 losing tricks.
- Three cards in suit: AKQ = 0 losing tricks, AKx, AQx, KQx = 1 losing trick, Axx, Kxx, Qxx = 2 losing tricks, xxx = 3 losing tricks
- Suits with more than three cards are judged according to their three highest cards since no suit can have more than three losing tricks

Table 3.3 gives an example of the evaluation of a hand of cards using the LTC technique.

The maximum number of losers is twelve (three for each suit). By determining the number of losers an indication is made of the number of cards that will not win a trick. We need a prediction of the number of tricks a hand will make. When we subtract the number of losers from the maximum number of losers possible (always twelve), we get this prediction. The prediction of the example in Table 3.3 will be $12-6=6$.

cards	losers
♠ A K J 2	1
♡ Q 10 9	2
♢ 10 9 4	3
♣ A K Q	0
Total	6 losers

Table 3.3: Example of Losing Trick Count.

The conditions for the advanced AI player to bid on a game type will be (1) based on the conditions to win that game type (Subsection 2.1.3) and (2) the results of the experiments to determine the accuracy of the bid method (Subsection 4.2.1).

The second technique is designed to determine the minimum number of winning tricks. It is used to determine the bids for Piek and Misère and counts the number of "winners". The following procedure is repeated for every suit. First the cards are divided into categories which each have their own value, see Table 3.4.

cards	2 3 4	5 6 7	8 9 10	J Q K	A
points	1	2	3	4	5

Table 3.4: The categories used in LTC for Piek and Misère.

For every card in the suit we add up the points corresponding to their categories. The number of points and the number of cards in the suit determine the number of winners in that suit. The parameter x represents the number of points:

- No cards in suit: 0 winning tricks.
- One card in suit: $x \leq 1 : 0$ winning tricks, $x \geq 2 : 1$ winning trick.
- Two cards in suit: $x \leq 3 : 0$ winning tricks, $x \geq 4 \wedge x \leq 6 : 1$ winning trick, $x \geq 7 : 2$ winning tricks.
- Three cards in suit: $x \leq 5 : 0$ winning tricks, $x \geq 6 \wedge x \leq 7 : 1$ winning trick, $x \geq 8 \wedge x \leq 10 : 2$ winning tricks, $x \geq 11 : 3$ winning tricks.
- Four cards in suit: $x \leq 7 : 0$ winning tricks, $x \geq 8 \wedge x \leq 9 : 1$ winning trick, $x \geq 10 \wedge x \leq 12 : 2$ winning tricks, $x \geq 13 \wedge x \leq 16 : 3$ winning tricks,$x \geq 17 : 4$ winning tricks.
- Suits with more than four cards are judged according to their four lowest cards.

For an example, see Table 3.5.

cards	points	winners
♠ A K J 2	$5 + 4 + 4 + 1 = 14$	3
♡ Q 10 9	$4 + 3 + 3 = 10$	2
♢ 10 9 4	$3 + 3 + 1 = 7$	1
♣ 5 3 2	$2 + 1 + 1 = 4$	3
Total		9

Table 3.5: Example of LTC for Piek and Misère.

3.3.2 Playing

This section describes the strategies of the advanced AI player. The strategies make uses of a technique to determine which card to get rid of in a situation where the advanced AI player is not able to follow suit. Each card has its value ranging from 0 to 12 and corresponds with the 2 to Ace from each of the

suits. The number of points in a certain suit are summed and divided by the number of cards from that suit. These results are summed to produce the final score for a certain hand. For an example see Table 3.6.

Cards	Points
♠ A ♠ K ♠ J ♠ 2	$(12 + 11 + 9 + 0) \div 4 = 8$
♡ Q	$10 \div 1 = 10$
♢ K ♢ J ♢ 4	$(11 + 9 + 2) \div 3 = 7.33$
♣ Q ♣ J ♣ 10 ♣ 8 ♣ 6	$(10 + 9 + 8 + 6 + 4) \div 5 = 7.4$
Total score	$8 + 10 + 7.33 + 7.4 = 32.73$

Table 3.6: Point system for the advanced AI player to determine which card is best to play.

When a card is played from this hand, the total score can change. It can go up, down or stay the same. If the score goes up this means a card is played that is of little importance to the total score. Otherwise the score would have gone down. This card is relatively lower than the other cards. For example when ♠ 2 is played: $12 + 11 + 9 \div 3 = 10\frac{2}{3} > 8$. If the score goes down, this means a card is played which is relatively higher than other cards. For example when ♠ A is played: $11 + 9 + 0 \div 3 = 6\frac{2}{3} < 8$. In some situations clearing a suit with a queen can lower the total score more than playing an ace that is from a suit which you possess other cards from. For example when the ♡ Q is played the final score goes down with 10 points. When ♠ A is played the final score goes down with only: $8 - 6\frac{2}{3} = 1\frac{1}{3}$. This method is used for Misère to determine which card is best to get rid of when not being able to follow suit.

When looking at the game we can distinguish 2 sides, the offensive side and the defensive side. The offensive side contains the declarer (the player who won the auction) and, if appropriate, his partner. The defensive side contains the rest of the players. In some game types the offensive side exists only of the declarer. The two sides use different strategies but both use the obtained information to decide which card must be played to follow a certain strategy.

The strategy for the game type Rik or Solo 8

The strategy of the offensive side is to first try to eliminate the trump of the defensive side and second playing their high cards of other suits. Eliminating the trump of the defensive side prevents the high cards from other suits being taken by the defensive side by using trump cards.

Strategy of the player from the offensive side that is first to go:

1. Check if defensive side has trump.
   If true or possibly true: continue with 2.
   Else play the highest card of a non-trump suit. If this is not possible, play a random card of a non-trump suit.

2. Information:

   - Defensive side possesses trump.

   Check if player possesses trump.
   If true: continue with 3.
   Else play the highest card of a non-trump suit. If this is not possible, play a random card of a non-trump suit.

3. Information:

   - Defensive side possesses trump.
   - Player possesses trump.

Check if player possesses highest trump.
If true: play the highest of trump.
Else: play a random of trump.

Strategy of the player from the offensive side that is second to go:

1. Check if you can follow suit.
   If true: continue with 2.
   Else: continue with 10.

2. Information:

   - You have to follow suit.

   Check if trick is currently in possession of your side.
   If true: continue with 3.
   Else: continue with 6.

3. Information:

   - You have to follow suit.
   - Trick in possession of your side.

   Check if highest card is played.
   If true: play your lowest card.
   Else: continue with 4.

4. Information:

   - You have to follow suit.
   - Trick in possession of your side.
   - Highest of that suit not played.

   Check if you possess the highest card.
   If true: continue with 5.
   Else: play a random card of that suit.

5. Information:

   - You have to follow suit.
   - Trick in possession of your side.
   - Highest of that suit not played.
   - You possess highest card.

   Check if partner played second highest card.
   If true: play lowest card.
   Else: play highest card.

6. Information:

   - You have to follow suit.
   - Trick not in possession of your side.

   Check if you have the highest card.
   If true: continue with 7.
   Else: continue with 9.

7. Information:

   - You have to follow suit.

- Trick not in possession of your side.
- You have the highest card.

Check if other players are all in your side.
If true: play the smallest card that is higher than the first card.
Else: continue with 8

8. Information:

   - You have to follow suit.
   - Trick not in possession of your side.
   - You have the highest card.
   - Not all/none of the other players are in your side

   Check if these players can follow suit.
   If true: play highest card.
   Else: play the smallest card that is higher than the first card.

9. Information:

   - You have to follow suit.
   - Trick not in possession of your side.
   - You do not possess the highest card

   Check if you have a higher card than played.
   If true: play the smallest card that is higher than the first card.
   Else: play lowest card.

10. Information:

    - You cannot follow suit.

    Check if trick is currently in possession of your side.
    If true: play a small card, no trump.
    Else: play a small trump card if possible, else a small other card.

Strategy of the player from the offensive side that is third to go:

1. Check if the trick has been trumped.
   If true: continue with 2.
   Else: follow the rules as if the player were second in turn.

2. Information:

   - Trick has been trumped.

   Check if you can follow suit.
   If true: play your lowest card.
   Else: continue with 3.

3. Information:

   - Trick has been trumped.
   - You cannot follow suit

   Check if trick is currently in possession of your side.
   If true: play smallest of a different suit than trump.
   Else: continue with 4.

4. Information:

- Trick has been trumped.
- You cannot follow suit
- Trick not in possession of your side.

Check if you possess a higher trump than played.
If true: play the smallest trump higher than the one played.
Else: play smallest card, if possible no trump.

Strategy of the player from the offensive side that is last to go:

1. Check if trick is currently in possession of your side.
   If true: play smallest card, if possible of a different suit than trump.
   Else: continue with 2.
2. Information:

   - Trick not in possession of your side.

   Check if you have a card to win the trick.
   If true: play the smallest card to win the trick.
   Else: play the smallest card, if possible no trump.

Strategy of the players on the defensive side is the same strategy as the offensive side except for the first turn:

1. Play the highest card of a non-trump suit. If this is not possible, play a random card of a non-trump suit. If this is also not possible, play a random card.

The strategy for the game type Piek

In this strategy one card is selected to win the one trick necessary to win the game, called the cover card. The two procedures (LTC) that are explained in Section 3.2.1 to determine the probability for each card to win a trick and the probability for each card to lose the card are used to select the cover card. The probabilities for each card are summed and the card with the highest value becomes the cover card. This card has a high chance of winning a trick when necessary and is easy to lose when another card wins a trick.

Strategy of the offensive side that is first to go is the following:

1. Check if you won zero tricks so far.
   If true: Play the card that when played results in the highest total score (Table 3.6), meaning the card is relatively low.
   Else play the game according to the Misère rules.

Strategy of the offensive side that is second, third or fourth to go is the following:

1. Check if you won zero tricks so far.
   If true: continue with 2.
   Else play the game according to the Misère rules.

2. Information:

   - You won zero tricks so far.

   Check if you have to follow suit.
   If true: continue with 3
   Else: continue with 4.

3. Information:

   - You won zero tricks so far.
   - You have to follow suit.

   If the cover card is the only high card left, play it.
   Else play the highest card lower than the cards already played, or else the lowest card of that suit.

4. Information:

   - You won zero tricks so far.
   - You cannot follow suit.

   Play a high card of any suit excluding the cover card.

Strategy of the defensive side is the following:

1. Check if your opponent won zero tricks so far.
   If true: Play a random card.
   Else play the game according to the Misère rules.

The strategy for the game type Misère

Strategy of the offensive side that is first to go is the following:

1. Play the card that when played results in the highest total score (Table 3.6), meaning the card is relatively low.

Strategy of the offensive side that is second, third or fourth to go is the following:

1. Check if you have to follow suit.
   If true: continue with 2.
   Else play the card that when played results in the lowest total score (Table 3.6), meaning the card is relatively high.

2. Information:

   - You must follow suit.

   Check if you have a lower card than already played.
   If true: play the highest card that is lower than the card already played.
   Else: play the lowest card of the suit asked.

The strategy of the defensive side that is first to go:
The strategy of the defensive side that is first to go is the same as the strategy of the offensive with the exception that you cannot play a card from a suit that the offensive side does not possess. If this is not possible play the card that has the highest total score meaning the card is relatively low.

Strategy of the defensive side that is second, third or fourth to go is the following:

1. Check if you have to follow suit.
   If true: continue with 2. Else: play the card that when played results in the lowest total score (Table 3.6), meaning the card is relatively high.

2. Information:

   - You must follow suit.

Check if the offensive side already played.
If true: continue with 3.
Else: play the highest card that is lower than the card already played.

3. Information:

   - You must follow suit.
   - Offensive side has already played.

   Check if the trick currently goes to the offensive side.
   If true: continue with 4.
   Else: play your highest card of the suit asked.

4. Information:

   - You must follow suit.
   - Offensive side has already played.
   - Trick currently goes to offensive side.

   Check if have a card lower than the offensive side.
   If true: play the highest card that is lower than the card already played.
   Else: play your highest card of the suit asked.

3.4 Monte-Carlo player

In this section we discuss the functionality of the Monte-Carlo Player (MC player). In order to discuss this functionality we first need to explain the principle of Monte-Carlo simulation (MC simulation). The principle behind MC simulation is that the behaviour of a statistic in random samples can be assessed by the empirical process of actually drawing lots of random samples and observing this behaviour (Mooney, 1997). When translated to the field of games, this comes down to simulating lots of random moves or games to get an indication of the average outcome of a certain move or game.

The implementation of a simulation starts with putting the real game on hold. The MC player creates four random players who will represent the players of the real game. The players are seated in the same order as the players they represent. The random player who represents the MC player receives the same cards as the MC player. The remaining cards (the cards that were not yet played minus the cards that were already played minus the cards of the MC player) are divided randomly among the other players. All the players receive the same amount of cards as the player they represent. This completes the initialising of the simulation. The players will finish the simulation by playing the rest of the game randomly.

3.4.1 Bidding

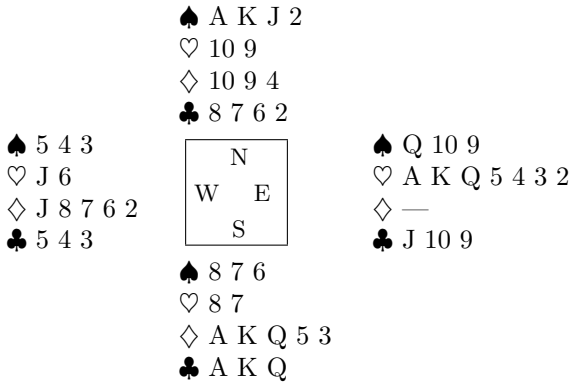
The MC player has to make a bid based on the hand of cards given. By simulating the game x number of times, the average outcome of the game can be determined for that particular hand of cards. The best number of simulations to choose the game type will be determined during the preliminary experiments in Subsection 3.4.3. For each card in the hand we save the number of games in which this card was played in the first round of the simulation and the total number of tricks that was won during those simulations. The total number of tricks is divided by the number of games to get the average trick value of every card. For the game types Rik and Solo 8 the conditions to win Rik or Solo 8 are compared to the highest trick value. The conditions to win Piek and Misère are compared to the lowest trick value. If multiple game types apply to the conditions, the game type with the highest payoff is chosen.

The MC player uses the same technique as the advanced AI player to decide on trump and ace, see Subsection 3.3.1.

3.4.2 Playing

Once the bidding round is finished, the playing round starts. When it is the turn of the MC player to play a card, he starts by putting the real game on hold. In order to decide which card the MC player will play, two procedures have to be executed. First the MC player will simulate the game y number of times. This number will be obtained during the preliminary experiments in Subsection 3.4.3. The second procedure is to pick the best card based on the results of the simulations. The first procedure will be explained by the example in Figure 3.6.

Situation: 1st round, player North starts the round, no real game type but the goal is to maximise tricks, without trump.



Round	Player North	Player East	Player South	Player West	Winner round
1	♠ A	♠ 9	—	—	—

Figure 3.6: Monte-Carlo player in action.

In the example represented in Figure 3.6, the South player represents the MC player and it is his turn. The MC player begins by creating 4 random players. The South player obtains the same cards as the MC player. The other players divide the remaining cards of the game randomly. This could result in the situation represented in Figure 3.7.

This game is played as a simulation of the original game. When the simulation is finished, the first card that was played by the South player and the total number of tricks made by the South player is saved. In

Simulation example, all players are random players.

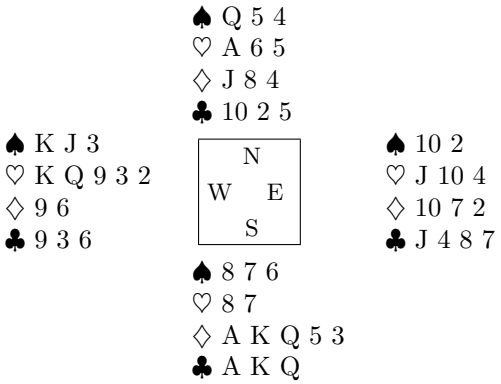


Figure 3.7: Monte-Carlo player simulating.

this case only three cards are possible to play, the 8, 7 and 6 of spades. This whole procedure is repeated y number of times.

The next procedure is to determine the average trick value of the cards. When all the simulations are finished, a vector with the first cards played and the number of tricks won in that simulation is obtained. For each first card, all the corresponding number of tricks are added up and divided by the total number of times that card has been played. For example when the 8 of spades is chosen in 5 out of the 10 simulations and the total number of tricks is 25, the average trick value of the 8 of spades will be $25 \div 5 = 5$.

The next step in the procedure is to pick the right card for the right game. When the objective is to maximise the number of tricks, the MC player picks the card with the highest average trick value. This is the case when the game type is Rik or Solo 8. The card with the lowest average trick value is played when the bid is Piek or Misère.

3.4.3 Preliminary experiments

The MC player uses simulations to decide on the game type in the bidding round and everytime when he has to play a card in the playing round. We performed two experiments to decide the number of simulations needed to produce consistent results. The first experiment is directed to the simulations in the bidding round. To test the consistency of the results of the simulations, we took 100 random hands. Each hand was then evaluated for the number of simulations we want to test. We take the average of the standard deviation of the 100 samples. The maximum average standard deviation that we would like is 0.1. In that case the bidding simulations are consistent up to 0.1 trick. This gives us the number of simulations that will be used when evaluating hands. The results can be found in Figure 3.8.

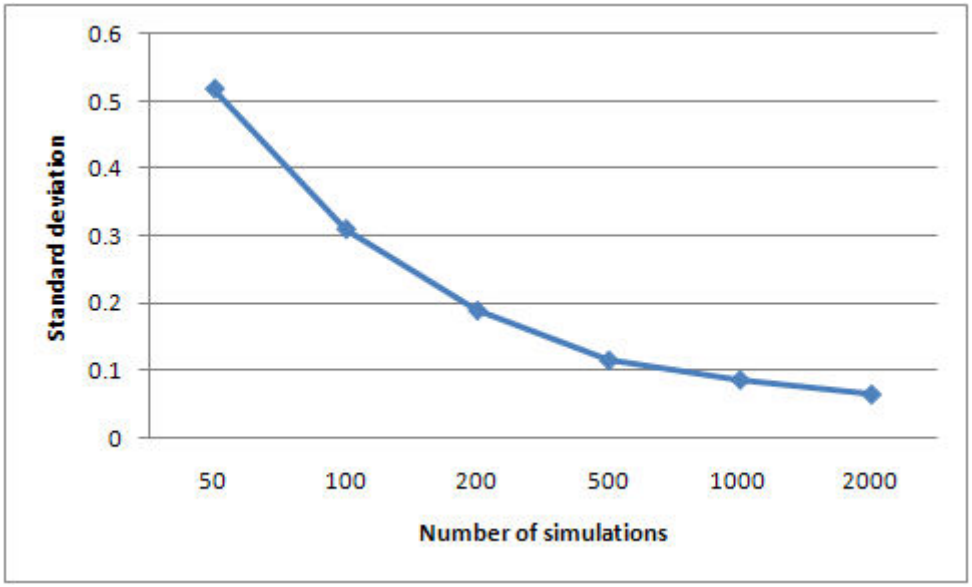


Figure 3.8: Monte-Carlo bidding consistency.

The second experiment is developed to find out how many simulations it takes to get consistent results when deciding which card is best to play. In order to measure the consistency we provided 100 times the same game situation to the MC player. The MC player then simulated that situation the number of times we wanted to test it. The result is a list of 100 cards that the MC player chose for the (same) 100 game situations. One would expect the same card for all the game situations if the simulations of the MC player are perfectly consistent. We would like at least 90% consistency for the first card of choice. This leaves 10% to play another card than the card that normally would be played. This makes the MC player more unpredictable for other players. The results of the experiments can be found in Figure 3.9.

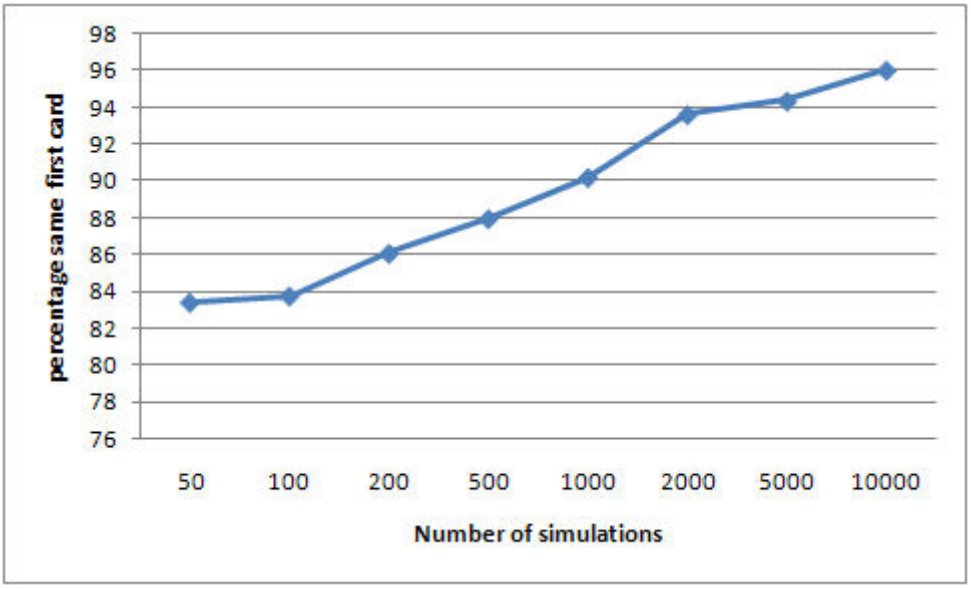


Figure 3.9: Monte-Carlo playing consistency.

The number of simulations we will use for the bidding round will be 1000. That is where the average standard deviation drops below 0.1, see Figure 3.8. The number of simulations during the playing round is also set on 1000. This is where the percentage of the same first card exceeds 90 %, see Figure 3.9.

3.5 Summary

We described the development of four AI players. Two of the players use rule-based techniques and one player uses Monte-Carlo techniques to make decisions and actions in the game. The developed players are the answer to the first research question *RQ 1*. Rule-based techniques can be compared to Monte-Carlo techniques by looking at the performance of the developed players in the game Rikken.

Chapter 4

Experiments and Results

In this chapter we present the experiments performed in this research. The experiments are designed to give answer to the last three research questions, *RQ 2*, *RQ 3*, and *RQ 4*. This chapter is divided in two parts, the experimental set-up (Section 4.1) and the results (Section 4.2).

4.1 Experimental set-up

This section is divided into three subsections. Section 4.1.1 describes the experiments to determine the performance of the bidding methods. Section 4.1.2 describes the experiments to determine the strength of the players in the playing round. Section 4.1.3 describes the tournament in which the overall strength of the players is determined.

4.1.1 Bidding methods analysed

In this section the bidding methods discussed in Section 3 are tested in order to answer the second research question *RQ 2*. Experiments will be performed to measure the accuracy of the bidding methods. For each bidding method, we take four equivalent players to play the game. We look at one particular player, player X, to measure the accuracy of the bidding method. Player X determines the number of tricks possible with his hands of cards using the bidding method and we save that number. We skip the rest of the bidding round and let player X be the winner of the bidding round. The four players play the game following the rules. We look at the number of tricks player X made during the game and also save this number. We now got the predicted number of tricks made by the bidding method of player X based on his hand of cards and the real number of tricks made by player X during the game. We will call these two numbers a datapoint. We repeat the whole process until enough datapoints are collected.

The number of tricks a player can make ranges from 0 to 13. We would like to continue testing until each of these possibilities is filled with 500 datapoints. Some possibilities, like winning 13 tricks, are very unlikely to occur. It would take too much computation time to get 500 datapoints for those possibilities; hence we decided to set the maximum number of experiments to be 100,000. The whole experiment is repeated for each of the players (except the random player) and for each of the game types. With three bidding methods, three types of players and four different game types, the total number of series experiments we have to run is 36.

4.1.2 Players' strength compared

The purpose of the experiments described in this section is to stipulate the strength of the players' tactics in the game types and answer the third research question *RQ 3*. To make the results more significant, the players have to face the same game situations. The number of ways to divide the four hands of cards of a game among four players equals $4! = 24$. Each player plays the same hand six times but each time with a different distribution of the remaining hands. To avoid the bidding system from having any influence on the results of the experiments the player that would start the bidding round automatically plays. The

distribution is arranged in such a way that the declarer always plays with the same hand of cards for the 24 games.

4.1.3 Tournament

In this section we describe the tournament experiments. The goal of the tournament is to determine the overall strength of the players and answer the fourth research question *RQ 4*. The overall strength of a player is a combination of the results in the bidding and playing round. Each of the players use their own bidding method. The conditions to bid on a certain game type are set to minimal conditions to win that game type. This means 5[1] or more tricks to play Rik, 8 or more to play Solo 8, between 1 and 0.5 to play Piek and 0.5 or less to play Misère. If multiple conditions apply, the game type with the highest payoff is chosen. In the first phase of the tournament all players participate and compete against each other in 96,000 games. The two best players proceed to play 50,000 games head-to-head to decide who is the best player. The best player is the one with the highest amount of chips at the end of the tournament.

4.2 Results

In this section the results of the experiments are presented. We first present the results of the bidding methods in Section 4.2.1. In Section 4.2.2 the results of the experiments to determine the strength of the players are presented. In Section 4.2.3 we present the results of the tournament.

4.2.1 Bidding methods results

In this section we present the results of the bidding methods. The experiments were done for the basic AI player, the advanced AI player, and the Monte-Carlo player and are ordered that way. The results are divided per game type for each of the players. The results shows to what extent the bidding methods on average approach the real number of tricks made. Each point in the graph represents the average of the 500 runs of experiments except for winning 0, 11, 12 and 13 tricks in the game types Rik and Solo 8. These averages are based on less than 500 runs but are not that interesting for those game types because the minimal conditions to play those game types are between 4 and 9 tricks. For each datapoint the average can over- or underestimate the real number of tricks. To determine which bidding method is the most accurate, we look at the total wrong estimation.

The results for the basic AI player shown in Figures 4.1 and 4.2 are very similar. All the bidding methods produce averages that are almost linear. The fluctuation at the beginning and the end is caused by the low number of experiments since it is very unlikely to win zero or thirteen tricks when playing Rik or Solo 8. The TPC method almost always underestimates the number of tricks made in the real game. The LTC method and the Monte-Carlo method predict almost the same but LTC underestimates more than it overestimates. The Monte-Carlo method overestimates about as much as it underestimates. The bidding method with the least number of wrong estimations is the LTP method, see Table 4.1.

The results of the basic AI player shown in Figures 4.3 and 4.4 show that all bidding methods heavily overestimate the real number of tricks made. The bidding method least bad at Piek was the LTC method and for Misère it was the Monte-Carlo method, see Table 4.1.

[1] According to (Sport, 2004), to win Rik you need around 5 winning tricks.

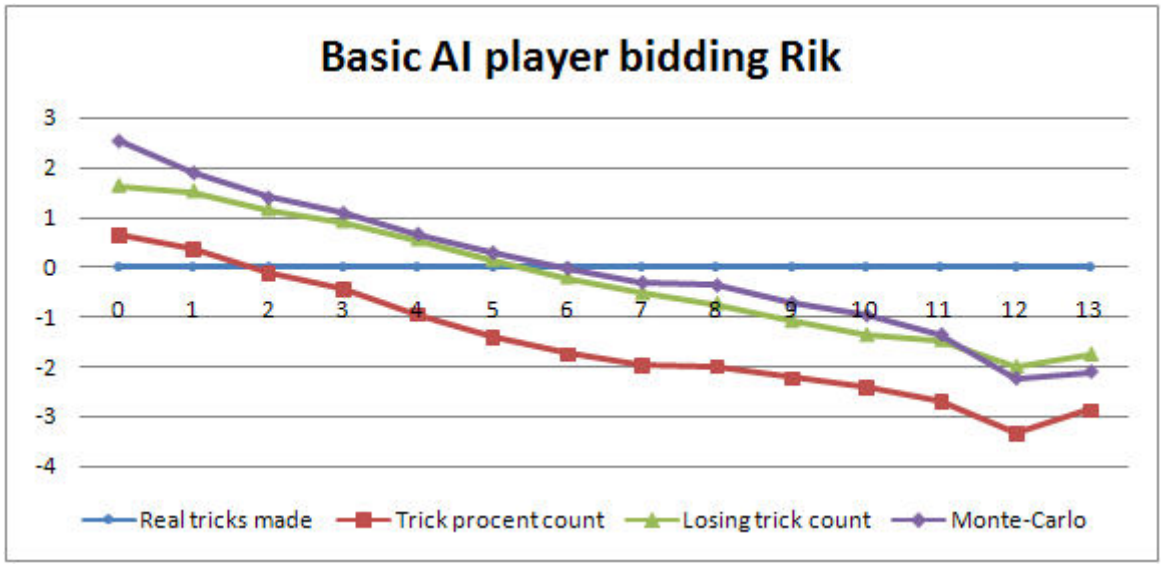


Figure 4.1: Results of the bidding methods with game type Rik for the basic AI players.

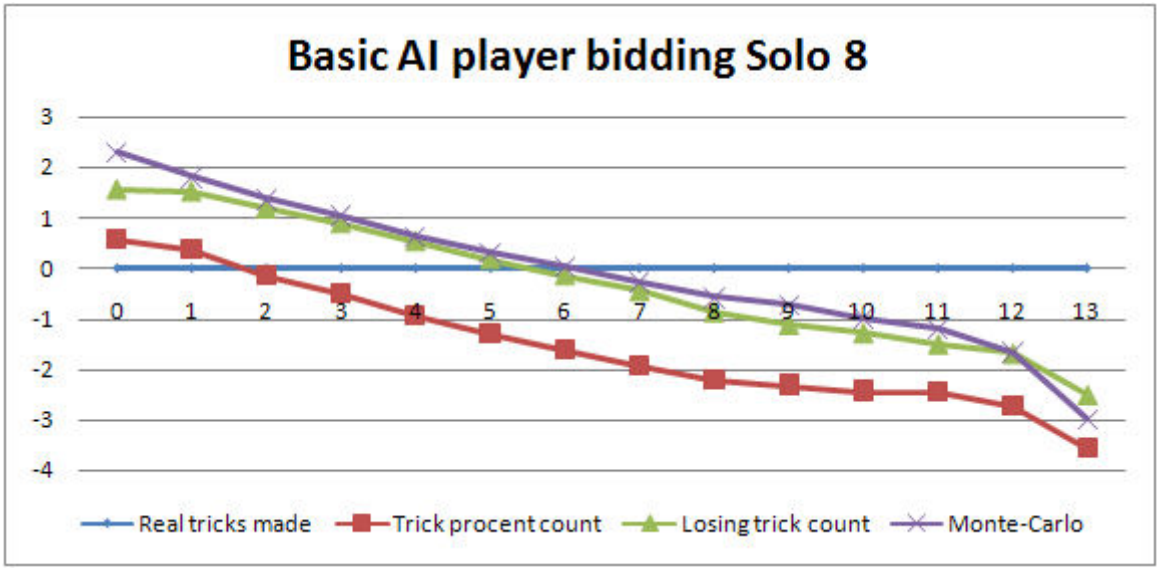


Figure 4.2: Results of the bidding methods with game type Solo 8 for the basic AI players.

	TPC				LTC				MC			
Estimate	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere
Over	1.02	0.93	5.79	5.02	5.95	5.87	4.63	4.39	7.88	7.52	4.85	4.36
Under	22.18	22.26	-	-	9.12	9.51	-	-	8.13	8.38	-	-
Total	23.20	23.19	5.79	5.02	15.07	15.38	4.63	4.39	16.01	15.91	4.85	4.36
Overall		57.20				39.47				41.13		

Table 4.1: Results of the bidding methods for basic AI players.

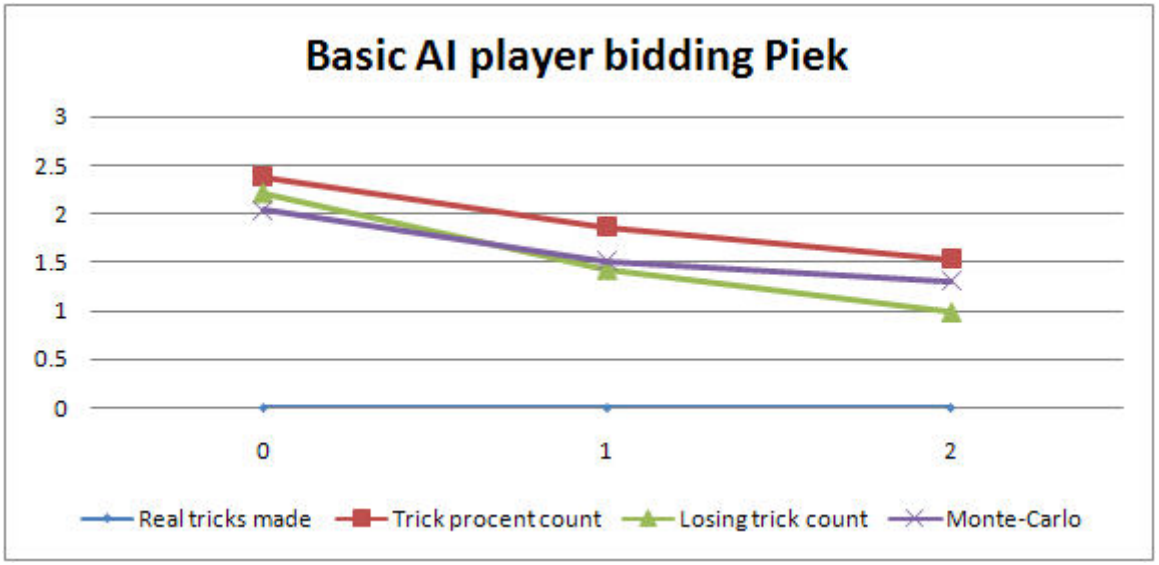


Figure 4.3: Results of the bidding methods with game type Piek for the basic AI players.

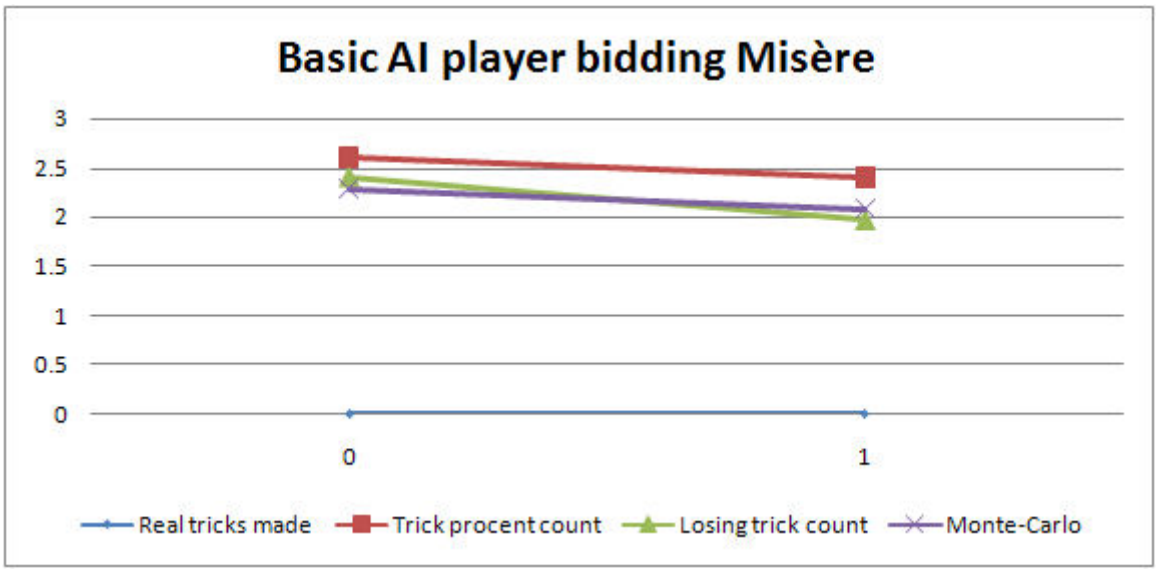


Figure 4.4: Results of the bidding methods with game type Misère for the basic AI players.

The results for the advanced AI player for Rik and Solo 8 are shown in Figures 4.5 and 4.6. They are in the same order as the results of the basic AI player. The TPC underestimates most of the time, LTC and Monte-Carlo are about the same. LTC underestimates more than it overestimates and the Monte-Carlo method overestimates at the same level as it underestimates. The best bidding method for both game types is the LTC method, see Table 4.2.

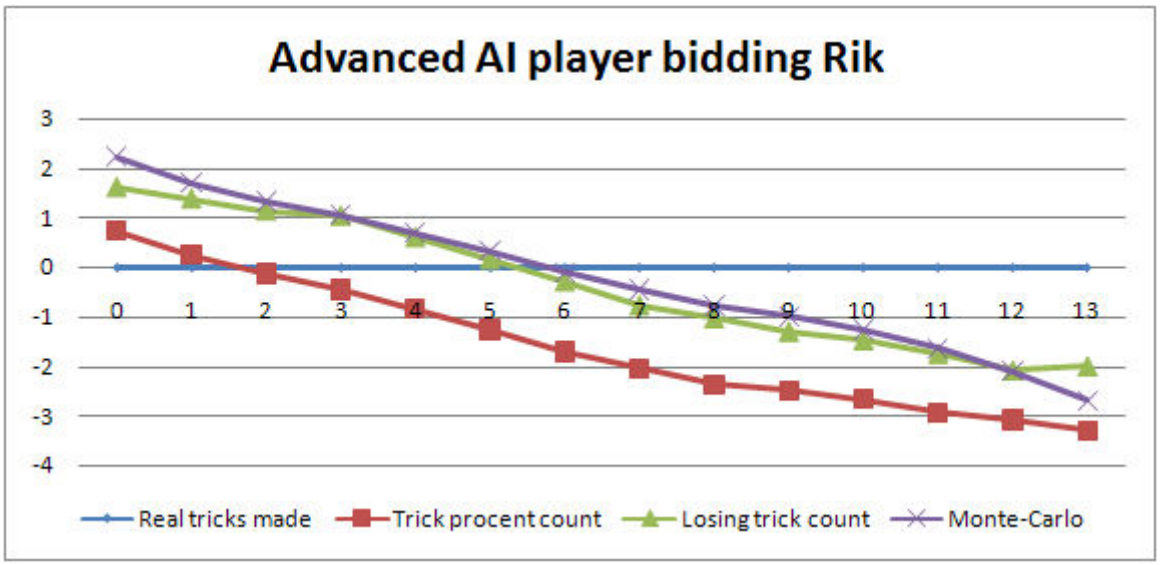


Figure 4.5: Results of the bidding methods with game type Rik for the advanced AI players.

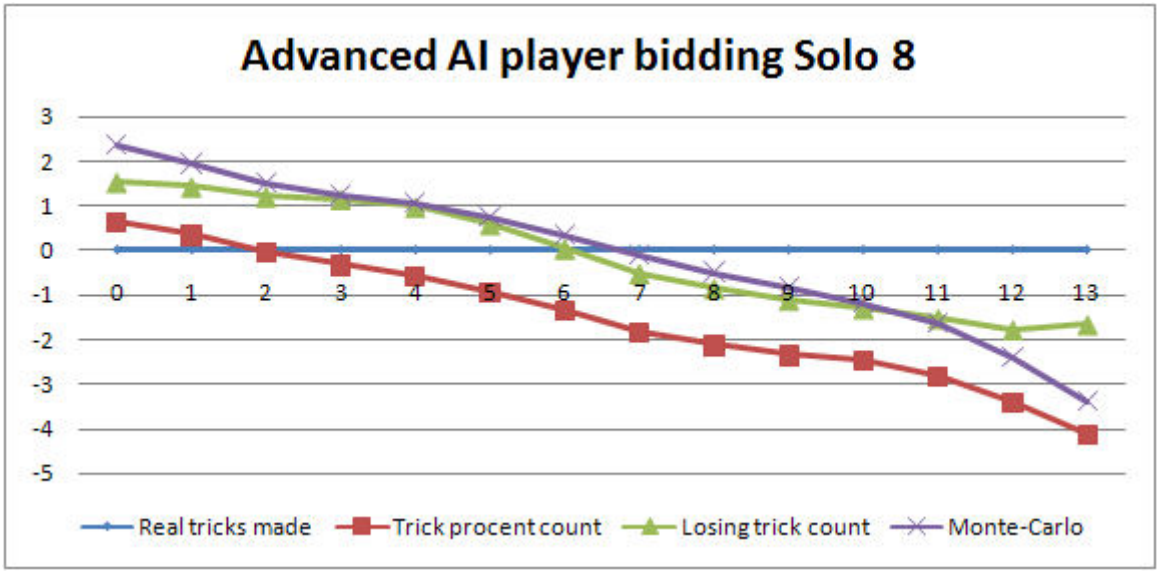


Figure 4.6: Results of the bidding methods with game type Solo 8 for the advanced AI players.

The results for the game types Piek and Misère for the advanced AI player can be found in Figures 4.7 and 4.8. They show that all methods overestimate the real number of tricks. The bidding method which is least bad at Piek and Misère is the LTC method, see Table 4.2.

The results for the Monte-Carlo player for Rik and Solo 8 are shown in Figures 4.9 and 4.10. The results

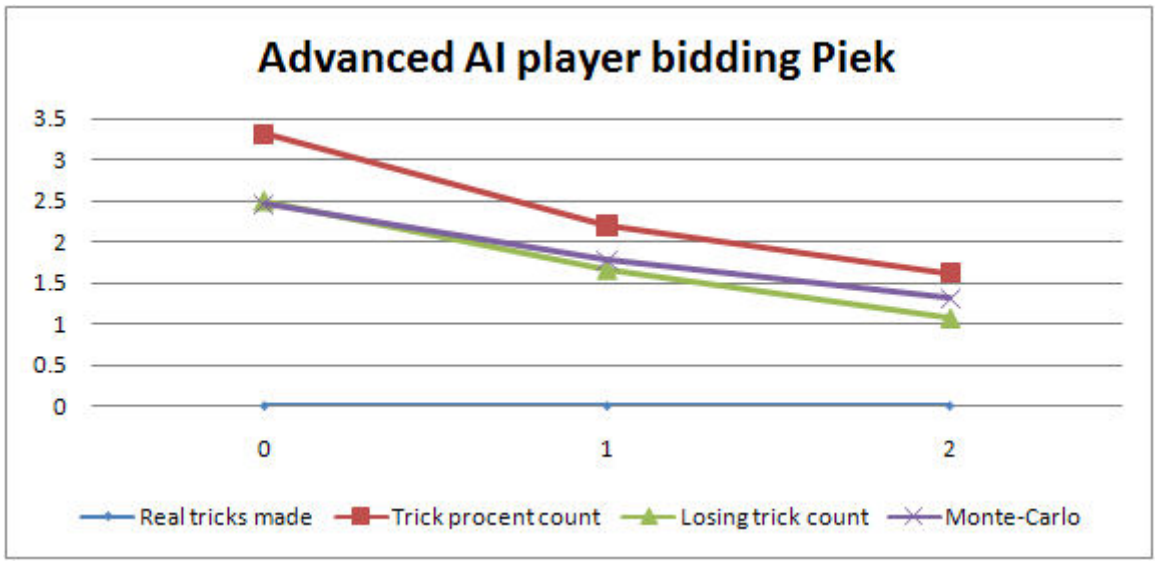


Figure 4.7: Results of the bidding methods with game type Piek for the advanced AI players.

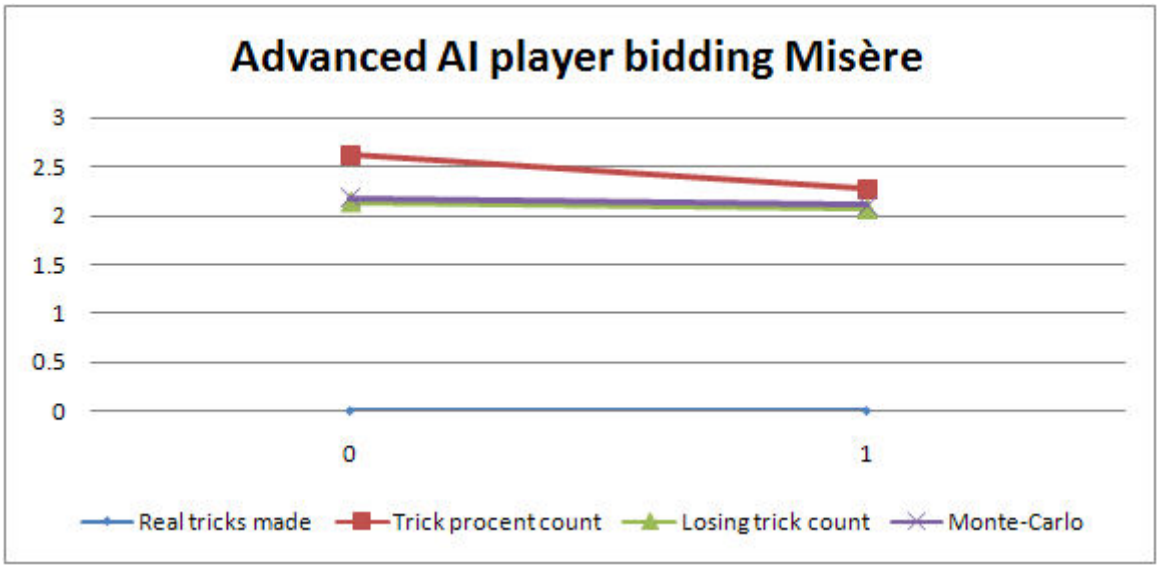


Figure 4.8: Results of the bidding methods with game type Misère for the advanced AI players.

	TPC				LTC				MC			
Estimate	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere
Over	0.96	0.94	7.13	4.88	5.94	6.94	5.23	4.2	7.36	9.13	5.55	4.28
Under	23.44	22.33	-	-	10.72	8.76	-	-	9.94	10.00	-	-
Total	24.40	23.28	7.13	4.88	16.66	15.70	5.23	4.2	17.29	19.13	5.55	4.28
Overall		59.69				41.79				46.25		

Table 4.2: Results of the bidding methods for advanced AI players.

for Rik show that the methods predict the real tricks quite good except for the last 3 points (winning 11 tricks and more). This can be explained by the low number of experiments where 11 and more tricks are actually won. The results for Solo 8 are similar to the results of the basic and advanced AI player. The best bidding method for both game types is the LTC method, see Table 4.3.

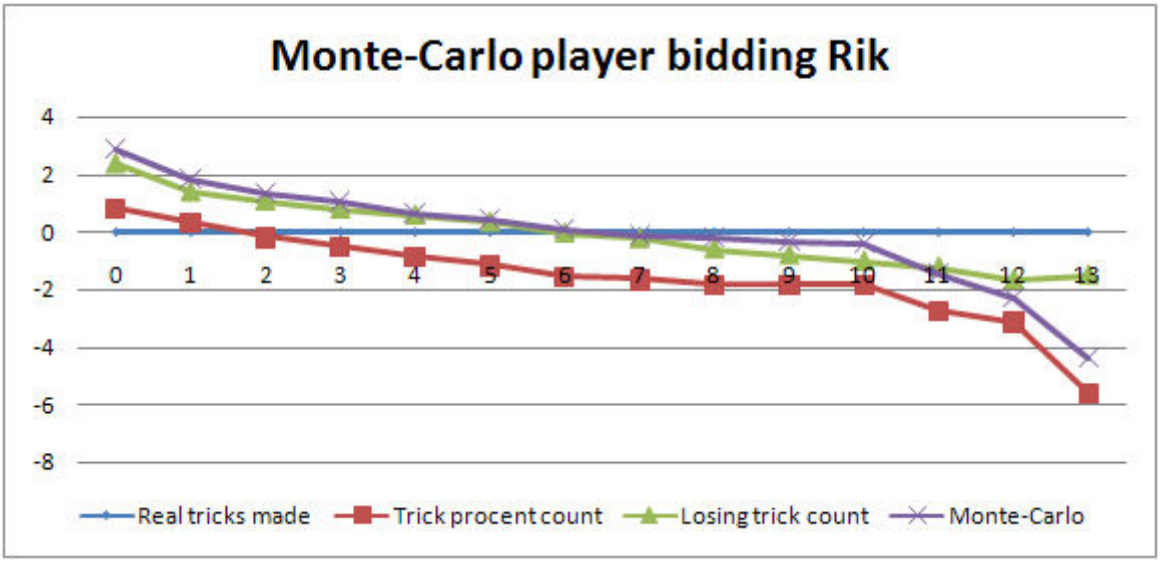


Figure 4.9: Results of the bidding methods with game type Rik for the Monte-Carlo players.

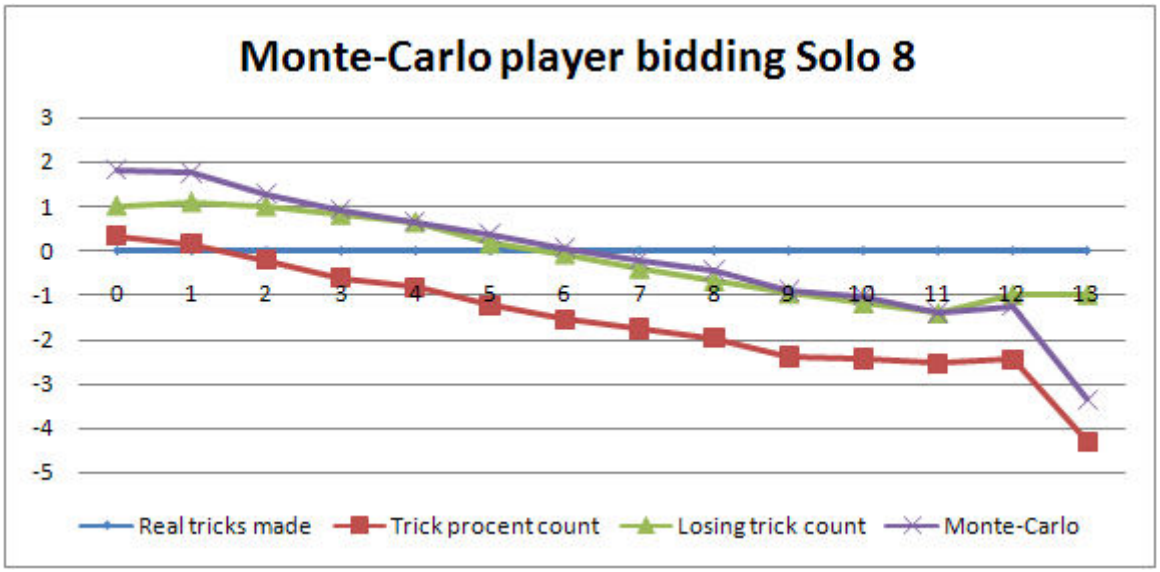


Figure 4.10: Results of the bidding methods with game type Solo 8 for the Monte-Carlo players.

The results for the game types Piek and Misère for the Monte-Carlo player can be found in Figures 4.11 and 4.12. They show that all methods overestimate the real number of tricks. The bidding method which is least bad at Piek and Misère is the LTC method, see Table 4.3.

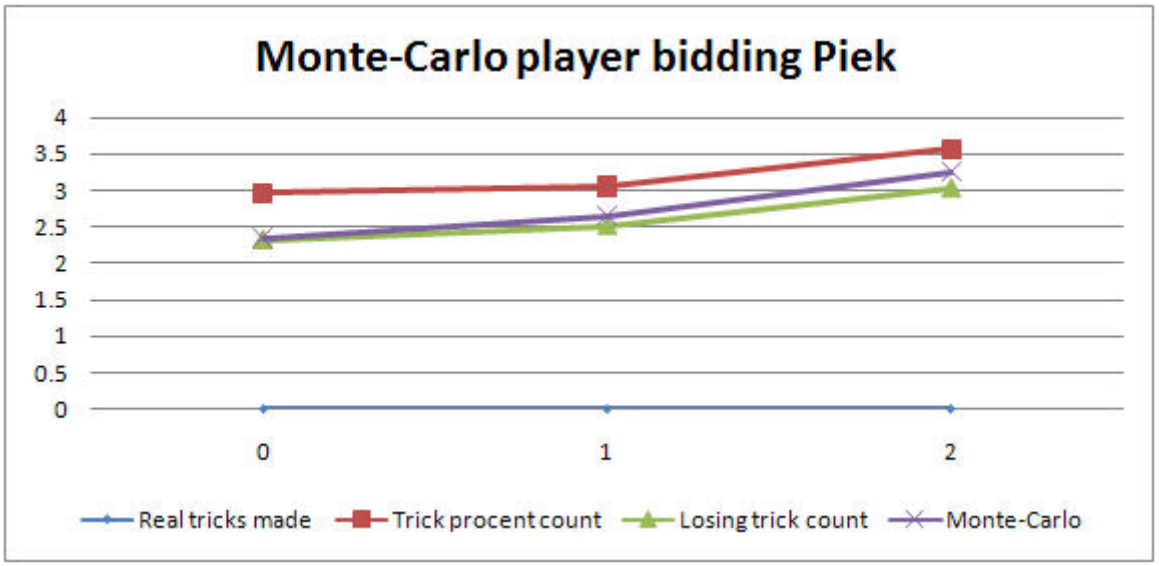


Figure 4.11: Results of the bidding methods with game type Piek for the Monte-Carlo players.

Figure 4.12: Results of the bidding methods with game type Misère for the Monte-Carlo players.

	TPC				LTC				MC			
Estimate	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere	Rik	Solo 8	Piek	Misere
Over	1.14	0.48	9.59	5.22	6.58	4.67	7.88	4.25	8.11	6.87	8.26	4.46
Under	22.83	22.29	-	-	7.09	6.76	-	-	9.31	8.63	-	-
Total	23.96	22.76	9.59	5.22	13.66	11.43	7.88	4.25	17.42	15.50	8.26	4.46
Overall		61.53				37.22				45.64		

Table 4.3: Results of the bidding methods for Monte-Carlo players.

4.2.2 Players' strength results

In this section we present results of the experiments regarding the strength of the players. We will first present the results of the basic AI player for the four game types. The best results, for all players and for all the game types are made against the random player. We will not mention this unless this is not the case.

Playing Rik is presented in Figure 4.13. The best partner for playing Rik is the random player and the basic AI player is the worst opponent against the advanced AI player. The results of Solo 8 can be found in Figure 4.14. The basic AI player scores worst against the advanced AI player, both in declaring as in opponent role. The same results are made with playing Piek, Figure 4.15. The results of playing Misère are shown in Figure 4.16. The advanced AI player is the best opponent when the basic AI player is the declarer. The basic AI player is the worst opponent against the Monte-Carlo player.

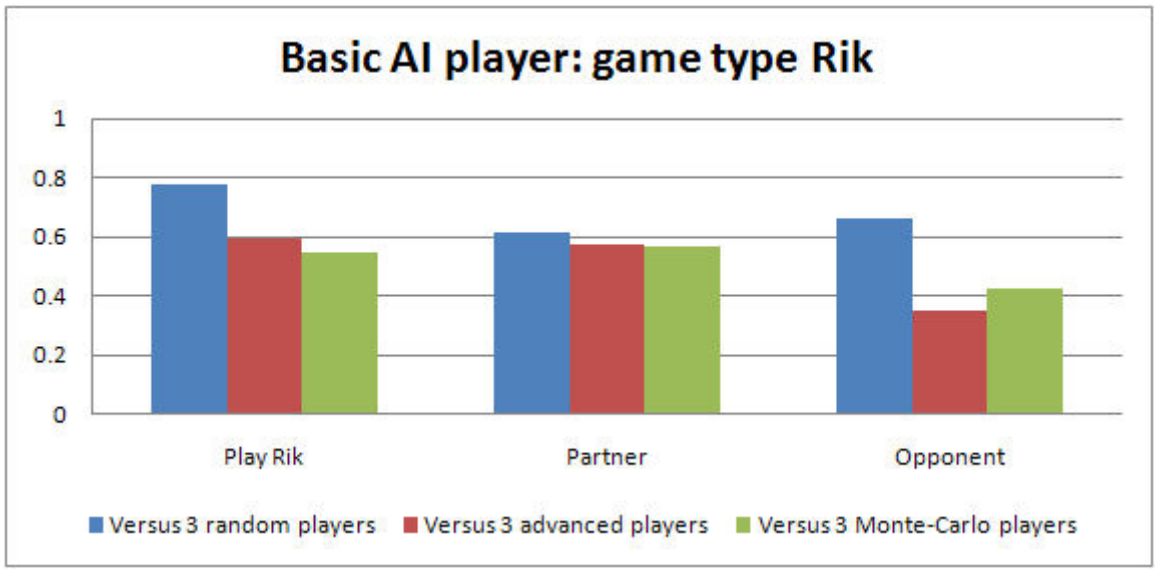


Figure 4.13: Basic AI players' strength when playing Rik.

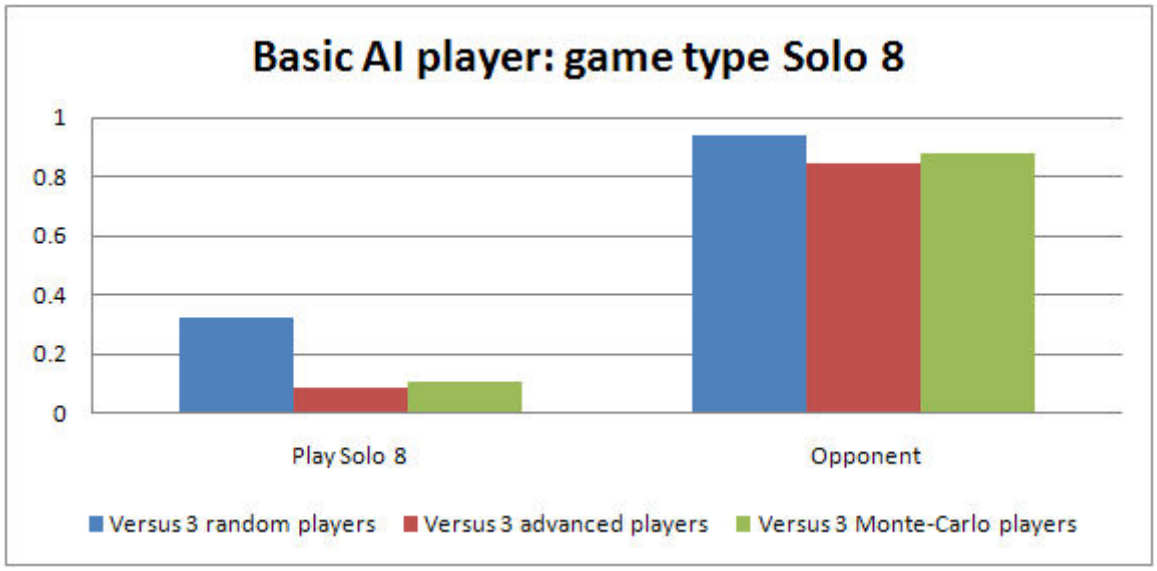


Figure 4.14: Basic AI players' strength when playing Solo 8.

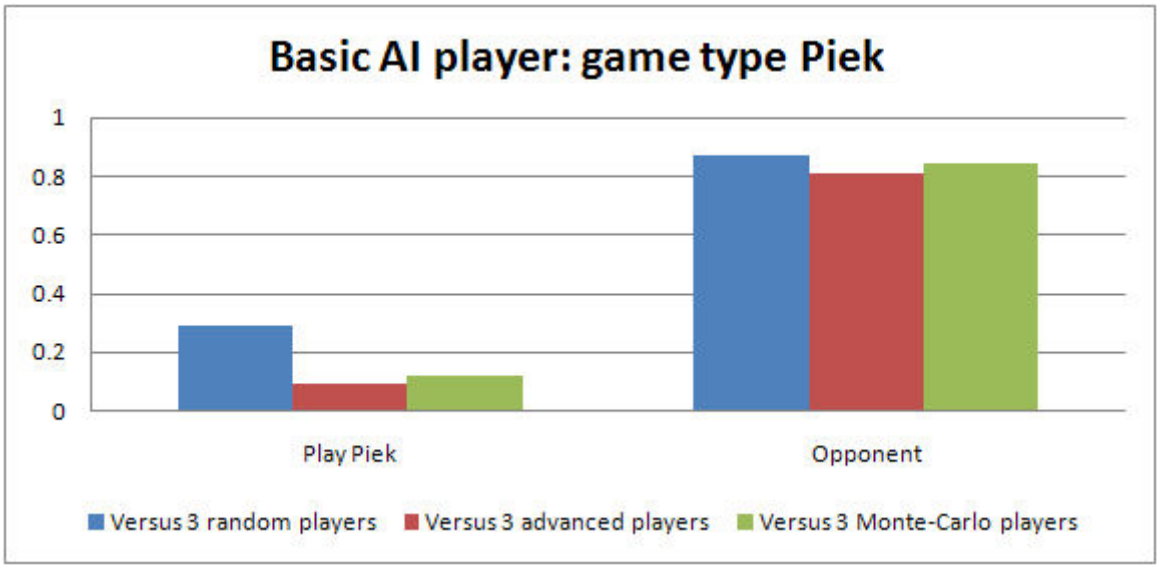


Figure 4.15: Basic AI players' strength when playing Piek.

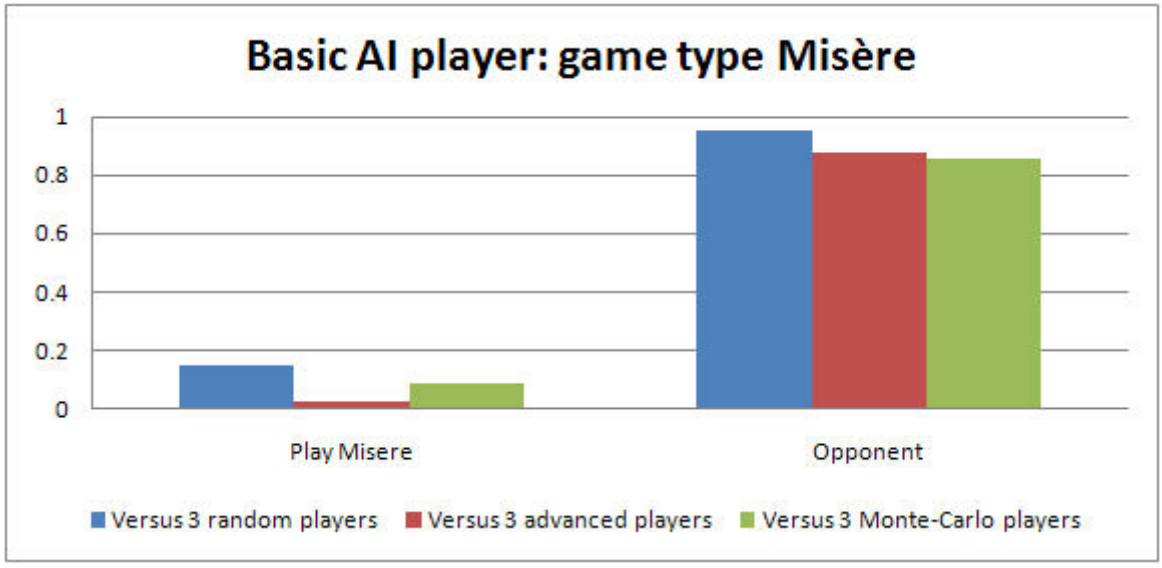


Figure 4.16: Basic AI players' strength when playing Misère.

The results of the advanced AI player for the four game types are presented next. Playing Rik is presented in Figure 4.17. The advanced AI player is evenly strong against the basic AI player and the Monte-Carlo player. The best partner for playing Rik is the Monte-Carlo player and the advanced AI player is the worst opponent against the basic AI player. The results of Solo 8 can be found in Figure 4.18. The advanced AI player scores worst against the Monte-Carlo player when he is in the declaring role. The advanced AI player is the worst opponent against the basic AI player. The result of the advanced AI playing Piek can be found in Figure 4.19. The advanced AI player scores worst against the Monte-Carlo player, both in declaring as in opponent role. The same goes for playing Misère, which can be seen in Figure 4.20.

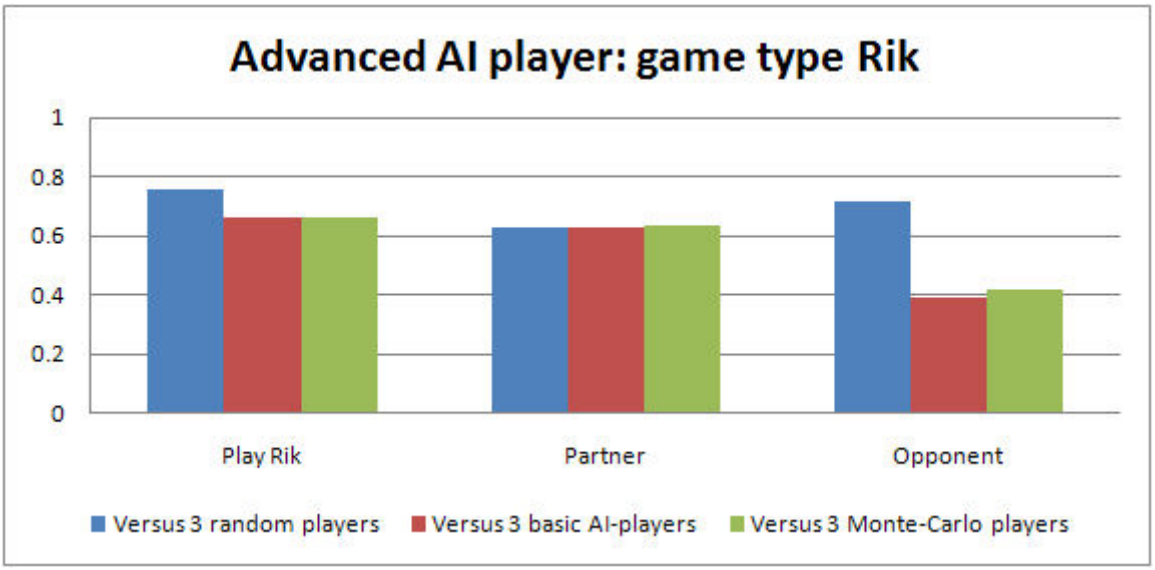


Figure 4.17: Advanced AI players' strength when playing Rik.

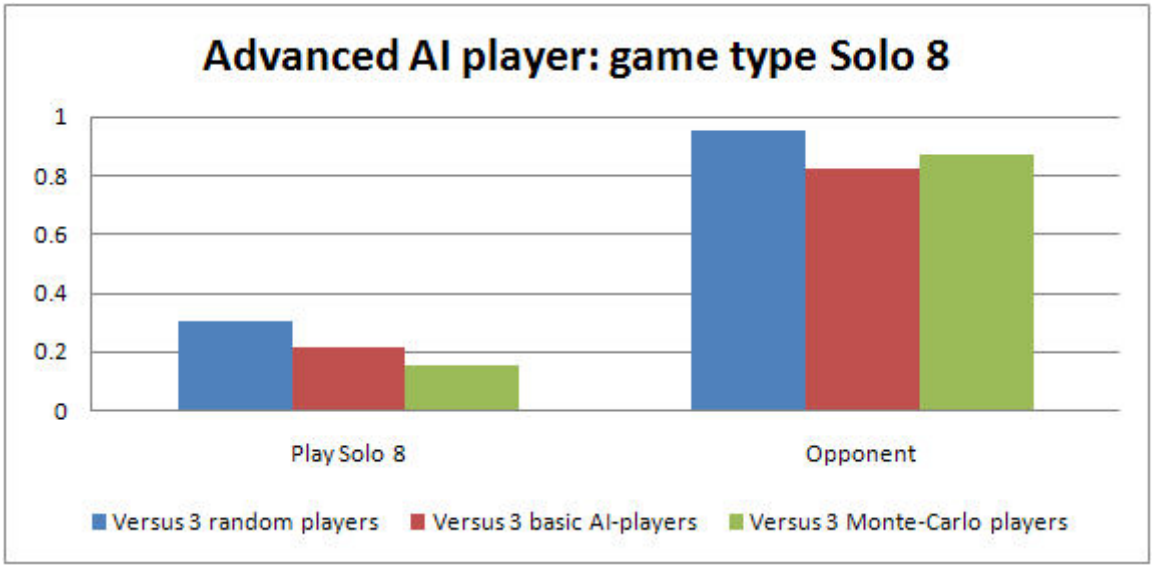


Figure 4.18: Advanced AI players' strength when playing Solo 8.

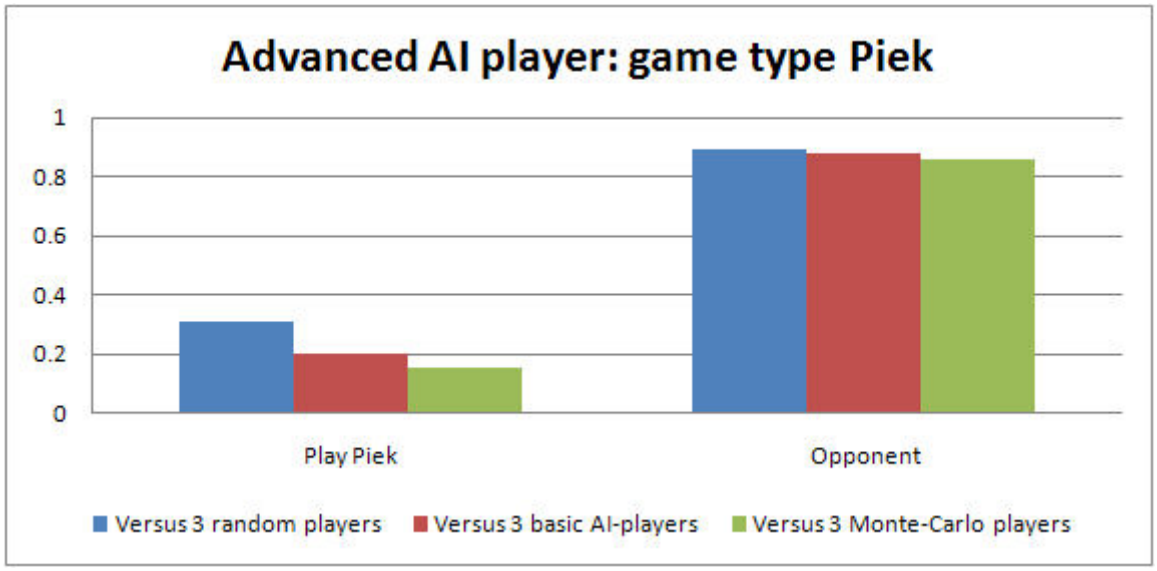


Figure 4.19: Advanced AI players' strength when playing Piek.

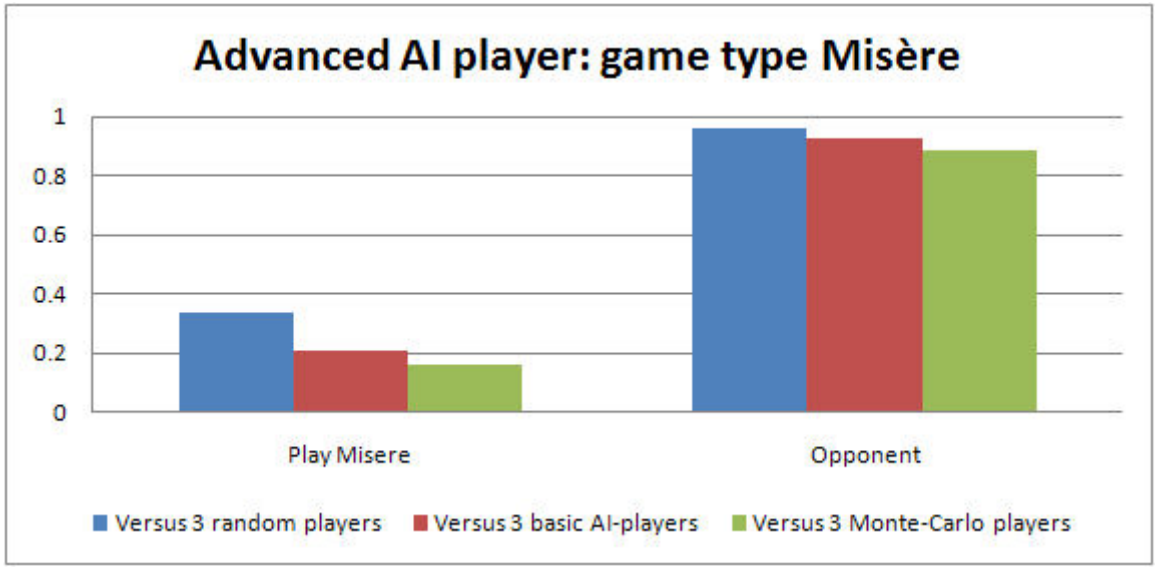


Figure 4.20: Advanced AI players' strength when playing Misère.

Finally we present the results of the Monte-Carlo player for the four game types. Figure 4.21 presents the results of the Monte-Carlo player playing Rik. The worst results are made when playing against the advanced AI player. The best results are made when the partner for the Monte-Carlo player is the basic AI player. The Monte-Carlo player is the worst opponent against the basic AI player. The results for Solo 8 can be found in Figure 4.22. The results show the same winning order of the players as the winning order of the Monte-Carlo player playing Rik. The Monte-Carlo player is as bad against the basic as the advanced AI player when playing the opponent. Piek results can be found in Figure 4.23. Again the same winning order when declaring. The Monte-Carlo player is the worst opponent against the basic AI player. The same results apply for the game type Misère, see Figure 4.24.

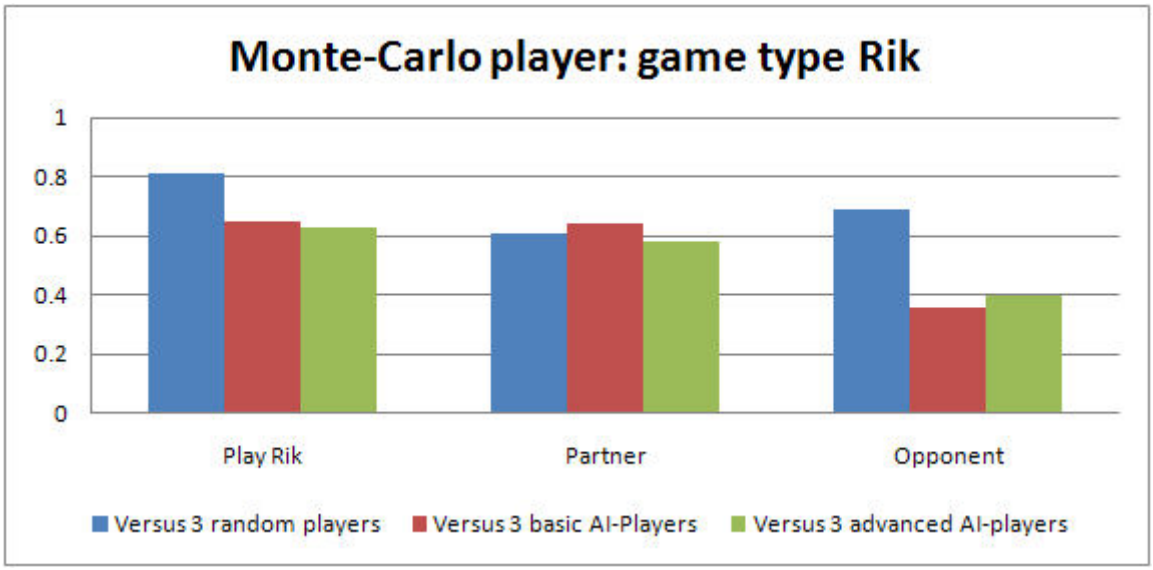


Figure 4.21: Monte-Carlo players' strength when playing Rik.

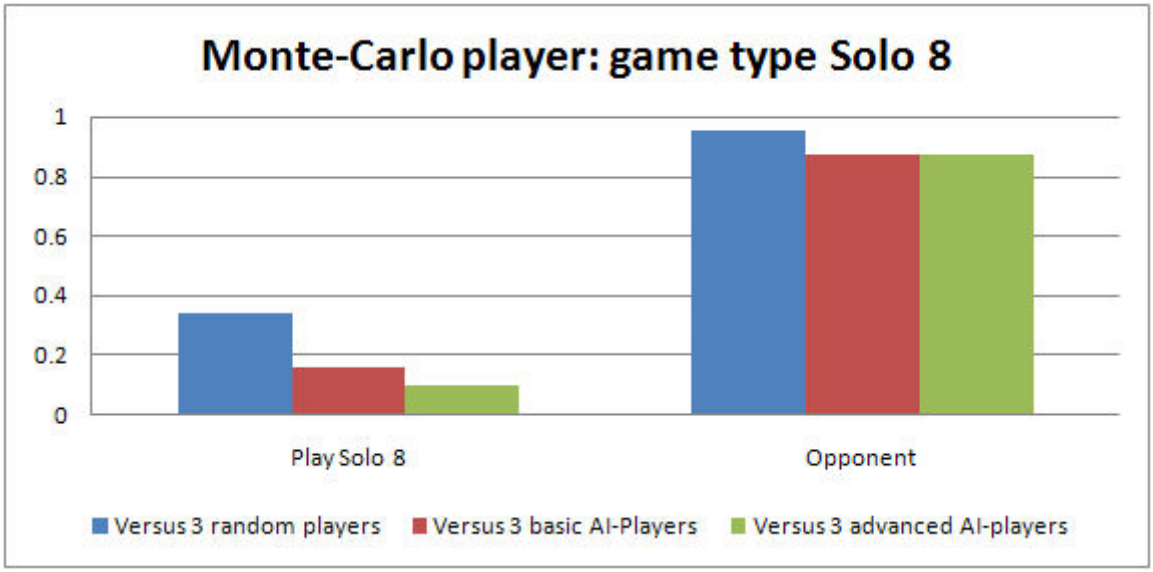


Figure 4.22: Monte-Carlo players' strength when playing Solo 8.

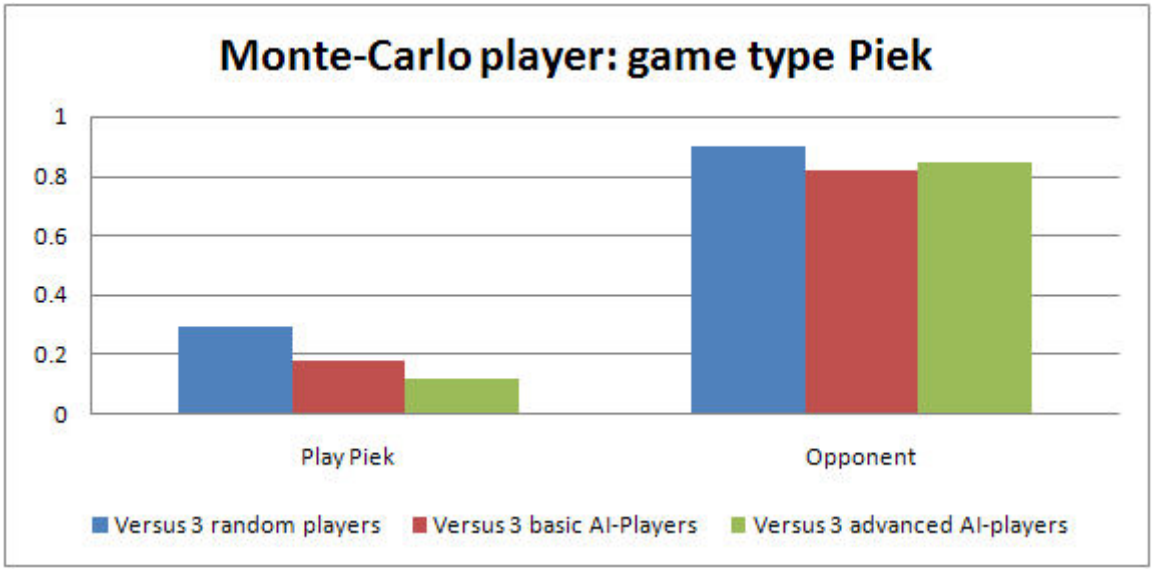


Figure 4.23: Monte-Carlo players' strength when playing Piek.

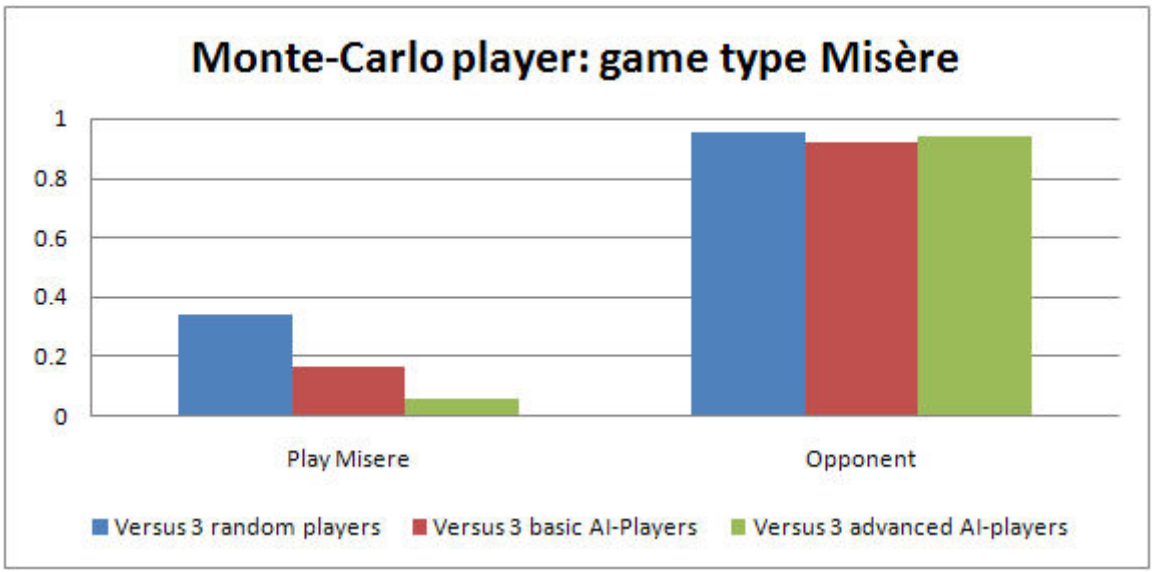


Figure 4.24: Monte-Carlo players' strength when playing Misère.

4.2.3 Tournament results

We first present the results of the phase of the tournament that was played with all types of players. In Figure 4.25 is shown how many times each player won the auction for a certain game type during the bidding round. The basic AI player bids in a very conservative way, therefore the basic AI player became declarer of the least amount of games compared with the other players. The Monte-Carlo player has the most amount of games in which he was declarer. The amount of games in which the advanced AI player was declarer of lies between the two other players.

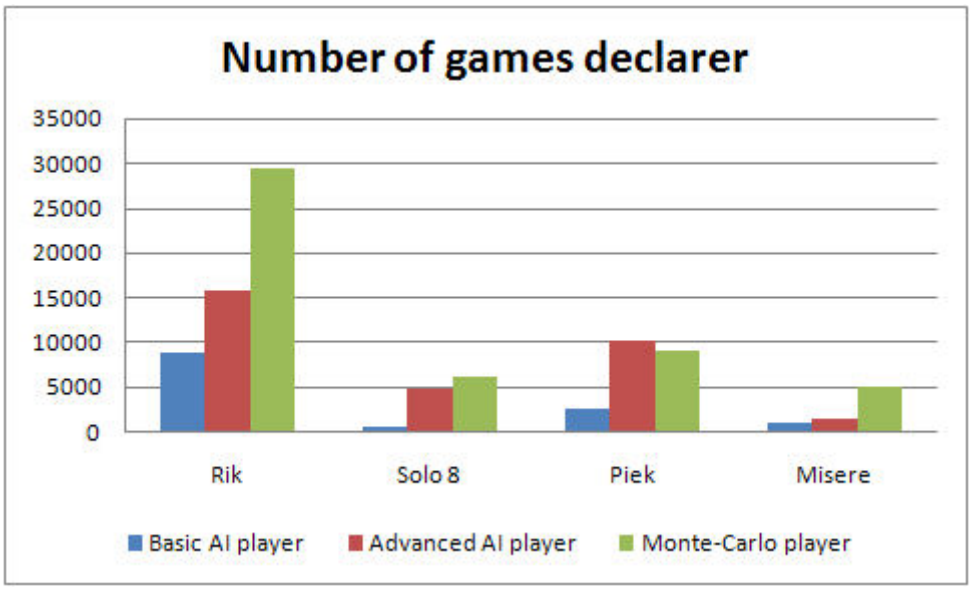


Figure 4.25: Tournament bidding results first phase.

The results of the players playing Rik is presented in Figure 4.26. It shows for each player the percentage of winning over the games in which that player was declarer. The number above each of the blocks represents the number of games in which that player was victorious. The strength of the player depends on the number of chips won, a high percentage does not automatically means strong player. In this case the basic AI player has the highest percentage of winnings but also the least amount of games won. The Monte-Carlo player has the second highest percentage and the most games won. The advanced AI player has the highest percentage and the second most games won.

The results of the players playing Solo 8 can be found in Figure 4.27. The basic AI player has the best winning percentage but won the least amount of games. The advanced AI player and the Monte-Carlo have almost the same winning percentage but the Monte-Carlo has won the most number of games.

The results of the players playing Piek are shown in Figure 4.28. The advanced AI player has the highest winning percentage and also the highest amount of games won. The basic AI player and the Monte-Carlo player have more or less the same winning percentage but the Monte-Carlo player won 3 times as many games.

The results of the players playing Misère can be found in Figure 4.29. The advanced AI player has the highest winning percentage, followed by the Monte-Carlo player. The basic AI player has the lowest bidding percentage and has also won the least amount of games. The Monte-Carlo won the most amount of games.

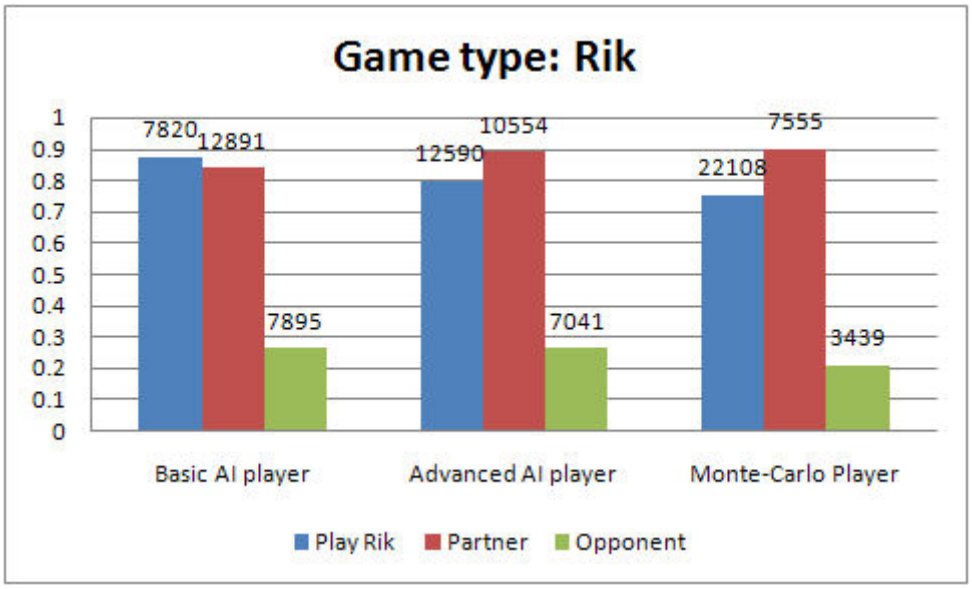


Figure 4.26: Tournament results for Rik.

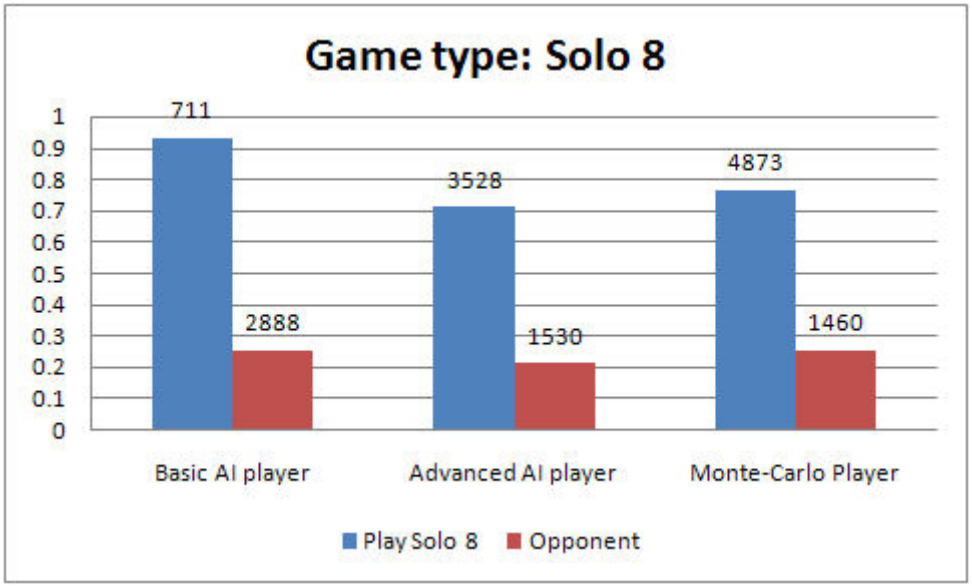


Figure 4.27: Tournament results for Solo 8.

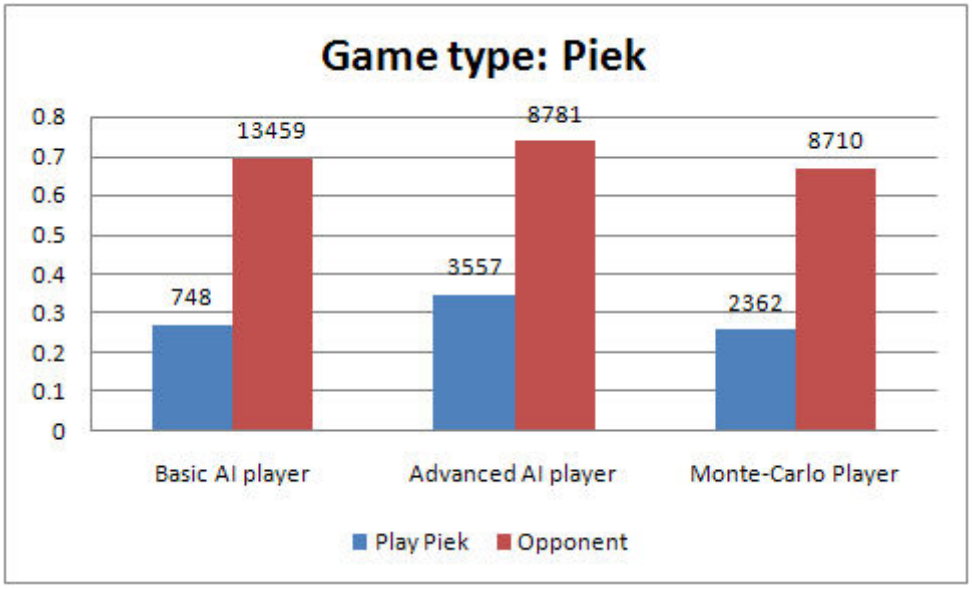


Figure 4.28: Tournament results for Piek.

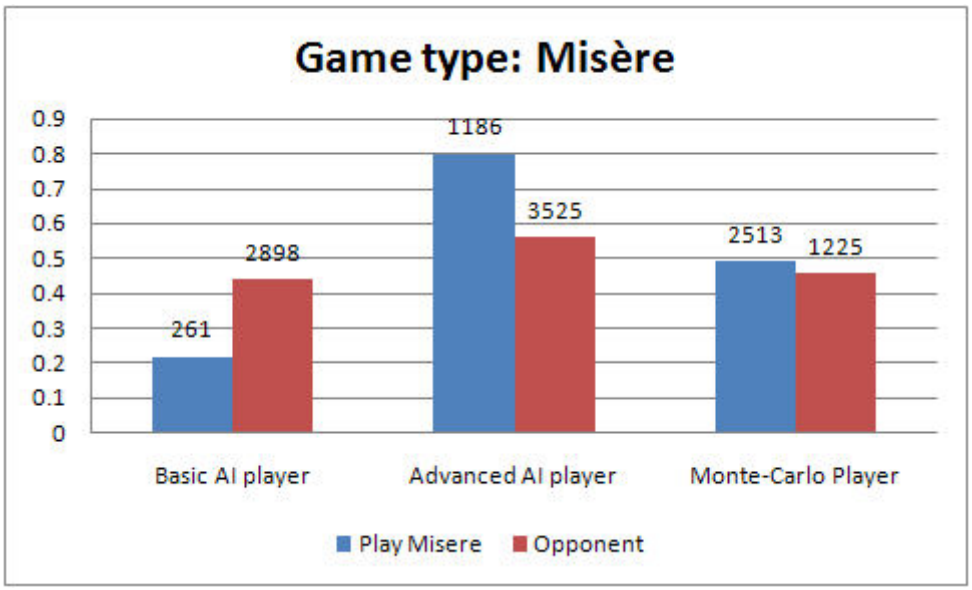


Figure 4.29: Tournament results for Misère.

The following results are from the final phase of the tournament. The best two types of players play head-to-head to decide which type of player has the best skills. Those two types are the advanced AI player and the Monte-Carlo player. In Figure 4.30 we present the amount of times each player won the auction for a certain game type during the bidding round. The Monte-Carlo player wins about twice as many games for the game types Rik and Misère. For Solo 8 and Piek these amounts are about the same.

The results of them playing Rik can be found in Figure 4.31. It can be seen that they all have about the same winning percentage but the Monte-Carlo player has played about twice as many games as the advanced AI player. The results of playing Solo 8 (Figure 4.32) are about the same as the results for Rik.

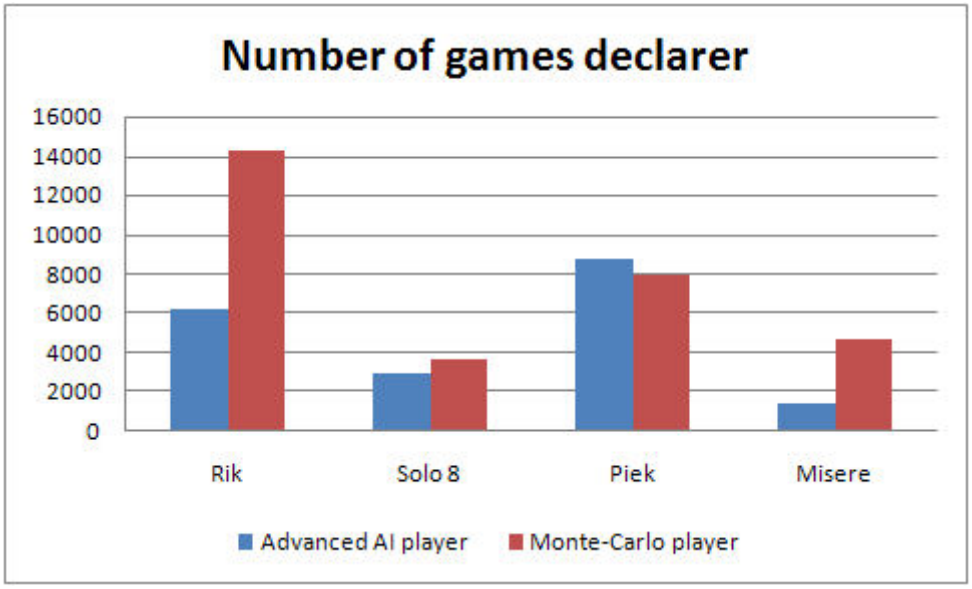


Figure 4.30: Tournament bidding results second phase.

Both types of players have about the same winning percentage but the Monte-Carlo player played about twice as many games.

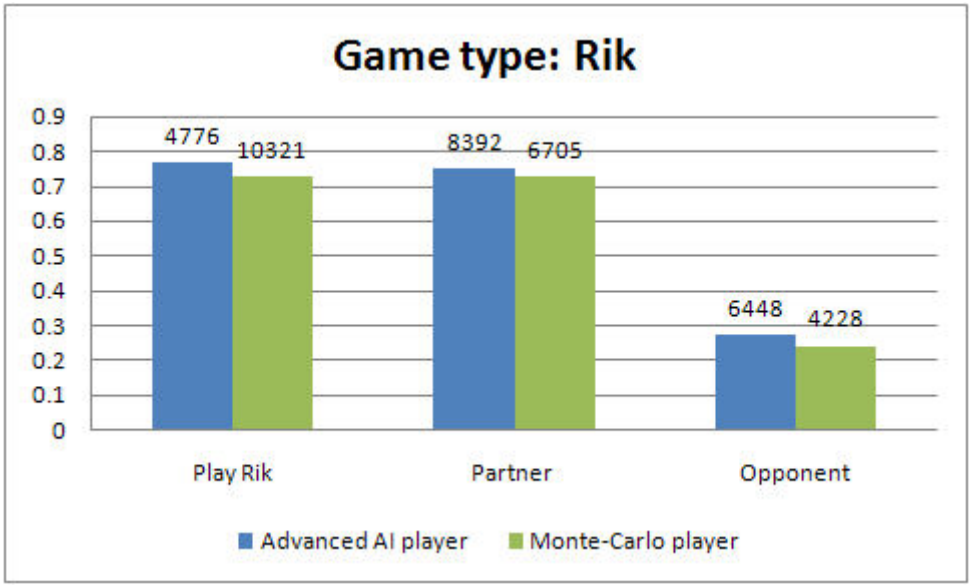


Figure 4.31: Tournament finals for Rik.

The results of Piek and Misere are found in Figures 4.33 and 4.34. The results for Piek show that the advanced AI player not only played more games but also has a higher winning percentage. The results for playing Misère show that the advanced AI player has a higher winning percentage. The Monte-Carlo player played a bit more games than the advanced AI player.

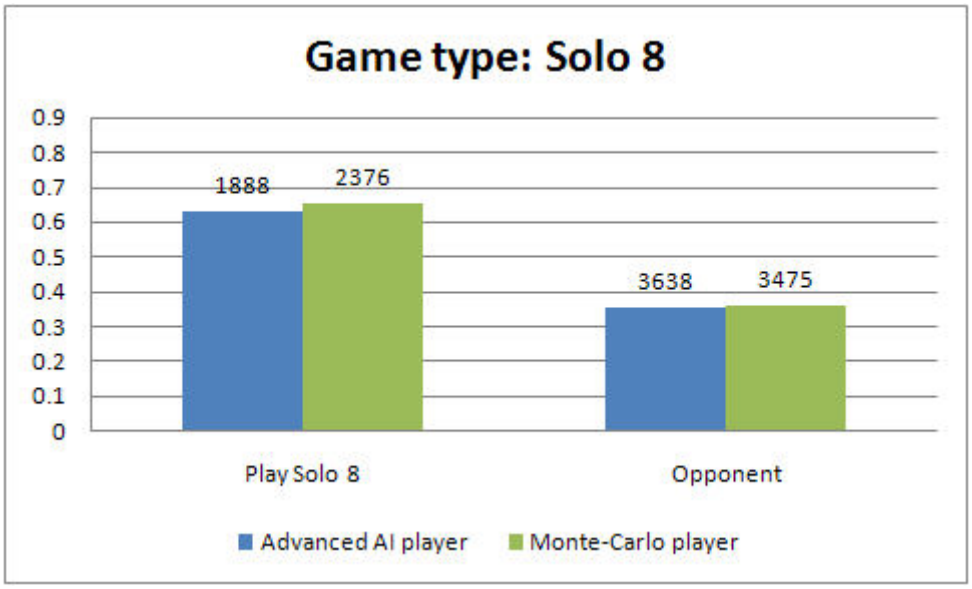


Figure 4.32: Tournament finals for Solo 8.

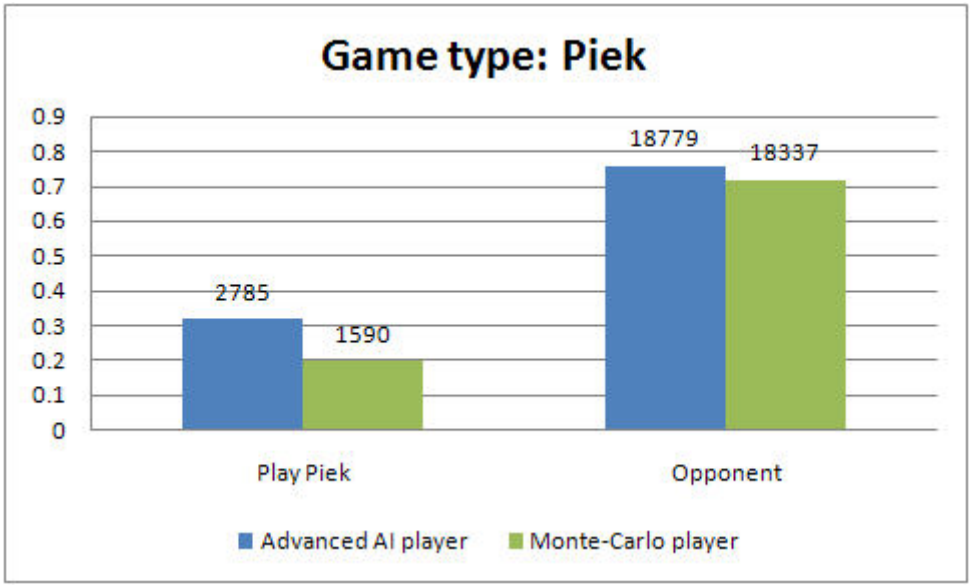


Figure 4.33: Tournament finals for Piek.

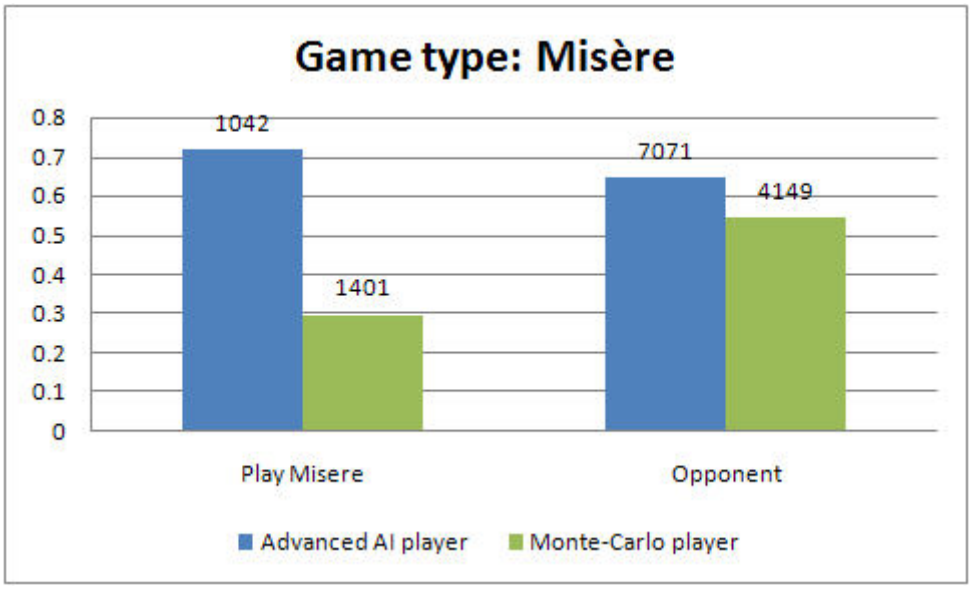


Figure 4.34: Tournament finals for Misère.

Chapter 5

Discussion

In this chapter we will analyse the results that were produced during the experiments (see Section 4.2) and give answer to *RQ 2*, *RQ 3*, and *RQ 4*. In Section 5.1 we will explain the results of the bidding experiments. Section 5.2 contains the analysis of the players' strength results. The results of the tournament are discussed in Section 5.3.

5.1 Bidding results analysed

In this section we will try to explain the results that were produced during the bidding experiments. The goal of the experiments was to find out which of the three bidding methods (TPC, LTC, and MC) is most accurate. For each bidding method, we repeat the properties of the method, the results, and the advantages and disadvantages of the method.

Trick Procent Count
This bidding method was developed for the basic AI player.

Functioning: For every card of a hand a calculation of chance is made of that card winning a trick. All the chances of a hand are then summed. This number gives a prediction of the number of tricks the hand will make.

Results:

- Often an underestimation of the real number of tricks made for Rik and Solo 8.
- Piek and Misère are always overestimated.
- The least accurate of all tested methods.
- Performed best with basic AI players.

Advantages:

- By underestimating the game types Rik and Solo 8 the chance of winning when playing is higher.

Disadvantages:

- By underestimating, some games that could be won are not played.
- Calculation of the chance does not take into account trump or the distribution of the suits.
- Not accurate.

Improvements:

- Different approach for Piek/Misère

- Add a constant factor in order to make the method more accurate.

Losing Trick Count
This bidding method was developed for the advanced AI player.

Functioning: Each suit contains 0 to 3 "loser" cards, based on the number of cards in that suit and the value of those cards. Counting the number of loser cards gives a prediction of the number of tricks that will be lost. With a maximum of 12 loser cards in a hand, the prediction of the number of tricks that will be made is 12 minus the number of loser cards.

Results:

- Overestimation of Piek and Misère.
- Most accurate bidding method tested.
- Performed best with MC players.

Advantages:

- Takes into account the value of the cards and the distribution of the suits in a hand when predicting the number of tricks a hand can make.
- Most accurate bidding method.

Disadvantages:

- Method is designed for maximizing the number of tricks. Bad for Piek and Misère where the goal is to make 1 and 0 tricks respectively.
- Method does not take into account the value of trump.

Improvements:

- Different approach prediction Piek and Misère.

Monte-Carlo bidding
This bidding method was developed for the MC player.

Functioning: By simulating the game, the average trick value can be determined that belongs to the hand of cards of the MC player.

Results:

- Overestimation of Piek and Misère.
- Second best accurate bidding method tested.
- Performed best with the basic AI players.

Advantages:

- Takes trump/no trump into account when simulating.
- Clear distinction between maximising (Rik, Solo 8) and minimising (Piek, Misère) trick game types.

Disadvantages:

- The simulations refrain from using knowledge of the game.

Improvements:

- Different approach prediction Piek and Misère.

Summary
The most accurate bidding method was the Losing Trick Count, which was developed for the Advanced AI player, a rule-based player. It seemed very difficult for all of the bidding methods to predict if a hand is suitable for Piek. There is no obvious Piek hand, but for the other game types there are obvious hands. Also the prediction when maximising the number of tricks is more straightforward than the prediction of minimising the number of tricks. The rule-based technique LTC gives more accurate results compared to the Monte-Carlo technique is the answer to the second research question *RQ 2*.

5.2 Players' strength results analysed

In this section we analyse the results of the experiments where the players' strength was tested.

Gametype Rik
In Table 5.1 an overview of the results of the strength of the players playing Rik is shown. The first part of the table shows the order of the strength for each player individually. For example, the basic AI player has the strongest play against the random player, then against the advanced AI player and plays worst against the Monte-Carlo player. What stands out in this part of the table is the fact that the basic AI player and the Monte-Carlo player have about the same strength as an opponent against the advanced AI player.

The second part of the table shows the order of strength for all players. For example, against the random player the Monte-Carlo player is the strongest player followed by the basic AI player and the least strong player is the advanced AI player. This example is remarkable because one would not expect the basic AI player being stronger than the advanced AI player.

	Individual best against:			Order of strength compared to all players:			
	1	2	3	Random	Basic	Advanced	Monte-Carlo
Basic	Random	Advanced	MC	2		2	2
Advanced	Random	Basic/MC		3	1		1
Monte-Carlo	Random	Basic	Advanced	1	2	1	

Table 5.1: Results analysed for game type Rik.

Partners: The advanced AI player is the best partner overall.
Opponents: The advanced AI player is the strongest opponent for the basic AI player and the MC player. The basic AI player is the strongest opponent for the advanced AI player.

Gametype Solo 8
Table 5.2 shows an overview of the results of the players' strength for the game type Solo 8. The table follows the same structure as the table above. The results of the strength regarding the individual players are as expected. In the order of the strength of all the players we notice that the basic AI player and the advanced AI player are evenly strong against the random player.

Opponents: The advanced AI player is the strongest opponent for the basic AI player and the MC player. The basic AI player is the strongest opponent for the advanced AI player.

Gametype Piek
An overview of the results of the players' strength for the game type Piek can be found in Table 5.3. The

	Individual best against:			Order of strength compared to all players:			
	1	2	3	Random	Basic	Advanced	Monte-Carlo
Basic	Random	MC	Advanced	2/3		2	2
Advanced	Random	Basic	MC	2/3	1		1
Monte-Carlo	Random	Basic	Advanced	1	2	1	

Table 5.2: Results analysed for game type Solo 8.

one thing that is remarkable about these results is the equal strength of the three players against the random player.

	Individual best against:			Order of strength compared to all players:			
	1	2	3	Random	Basic	Advanced	Monte-Carlo
Basic	Random	Advanced	MC			2	2
Advanced	Random	Basic	MC	1/2/3	1		1
Monte-Carlo	Random	Basic	Advanced		2	1	

Table 5.3: Results analysed for game type Piek.

Opponents: The advanced AI player is the strongest opponent for the basic AI player and the MC player. The MC player is the strongest opponent for the advanced AI player.

Gametype Misère
Table 5.4 gives an overview of the results of the strength of the player playing Misère. The strength of the individual players is what we would expect. The results of the order of strength against the random player is remarkable since the advanced AI player and the Monte-Carlo player have the same strength.

	Individual best against:			Order of strength compared to all players:			
	1	2	3	Random	Basic	Advanced	Monte-Carlo
Basic	Random	MC	Advanced	3		2	2
Advanced	Random	Basic	MC	1/2	1		1
Monte-Carlo	Random	Basic	Advanced	1/2	2	1	

Table 5.4: Results analysed for game type Misère.

Opponents: The MC player is the strongest opponent for the basic AI player and the advanced AI player. The advanced AI player is the strongest opponent for the MC player.

Summary
The strongest player is the advanced AI player who played best in 10 of the 12 games he participated in. Second is the MC player with 8 out of 12. The basic AI player finished last with 1 out of 12. The best opponent is the advanced AI player in three out of four game types. This answers the third research question *RQ 3*.

5.3 Tournament results analysed

In this section we analyse the results of the tournament. In the first phase of the tournament all the developed types of players competed. The random player is refrained from bidding and only acts as a opponent. The results of the random player are therefore not presented with the other players. The two types of players that performed best continued to play head-to-head.

The results of the first phase can be found in Figures 5.1, and 5.2. The basic AI player plays a bit better than break even. The best game types for the basic AI player are Rik and Piek and the worst one is Misère. The advanced AI player is the best player in this phase of the tournament. He makes profit

from three out of four game types. The only negative game type for the advanced player is Piek. The second best player is the Monte-Carlo player, who has the best results in Rik and Solo 8. Piek is by far the worst game type for the Monte-Carlo player. The results indicate that the advanced AI player is the best player and the Monte-Carlo player the second. Those two players continue to the next phase of the tournament.

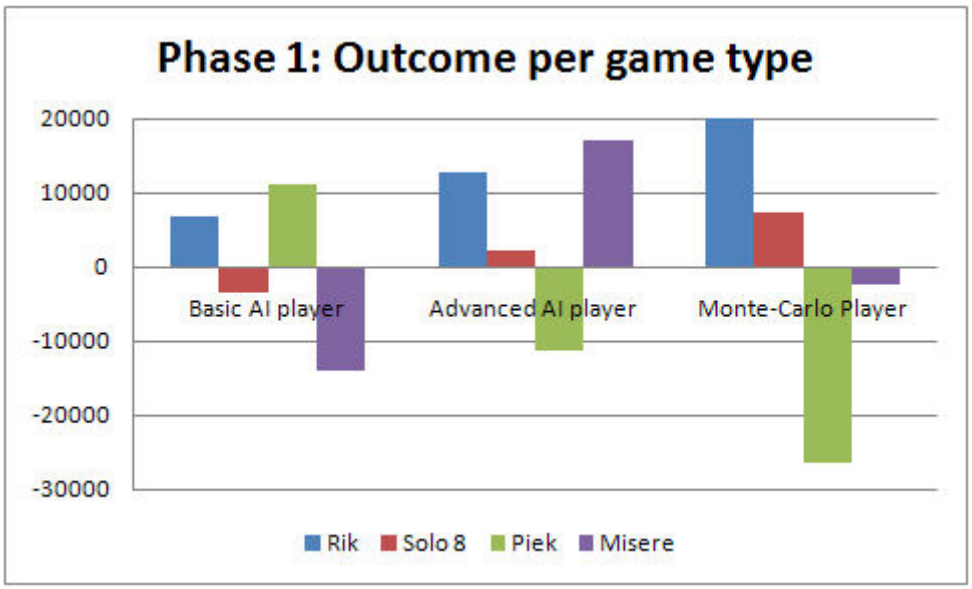


Figure 5.1: Tournament results in chips.

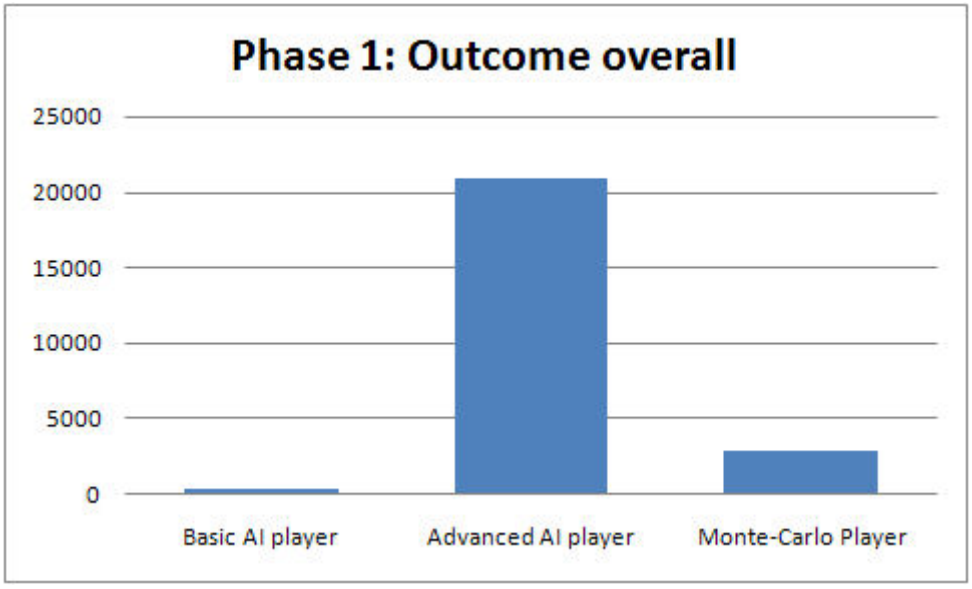


Figure 5.2: Tournament results overall in chips.

Figures 5.3, and 5.4 show the results of the final phase of the tournament. Two advanced AI players competed against two Monte-Carlo players. The averages of the two types of players are shown and we can clearly see that the advanced AI player has won the tournament. Although the Monte-Carlo players were better in playing Rik and Solo 8, they could not win from the advanced AI player when playing Piek and Misère. Since those game types have the highest payoff, the advanced AI player convincingly

won the tournament.

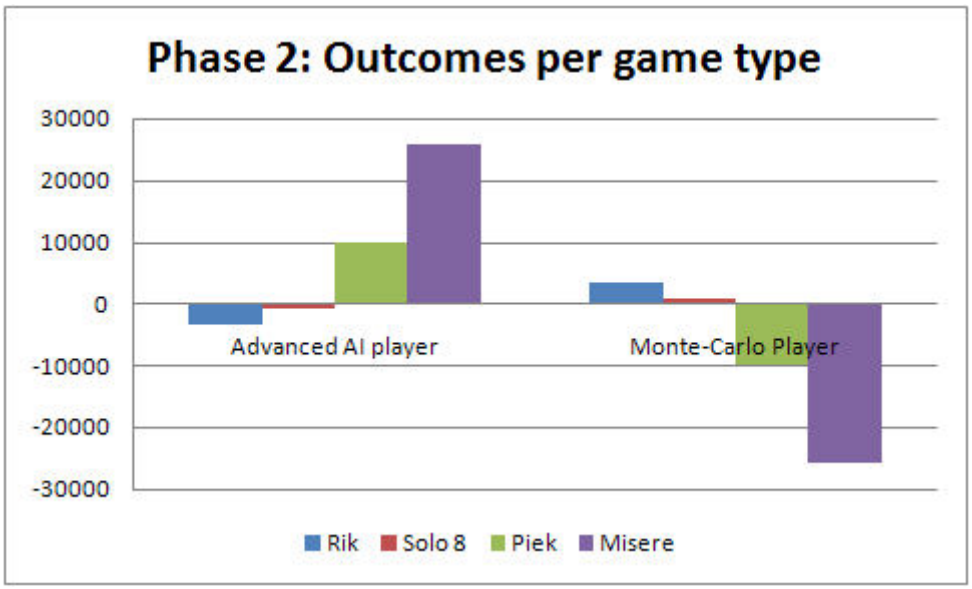


Figure 5.3: Tournament finals in chips.

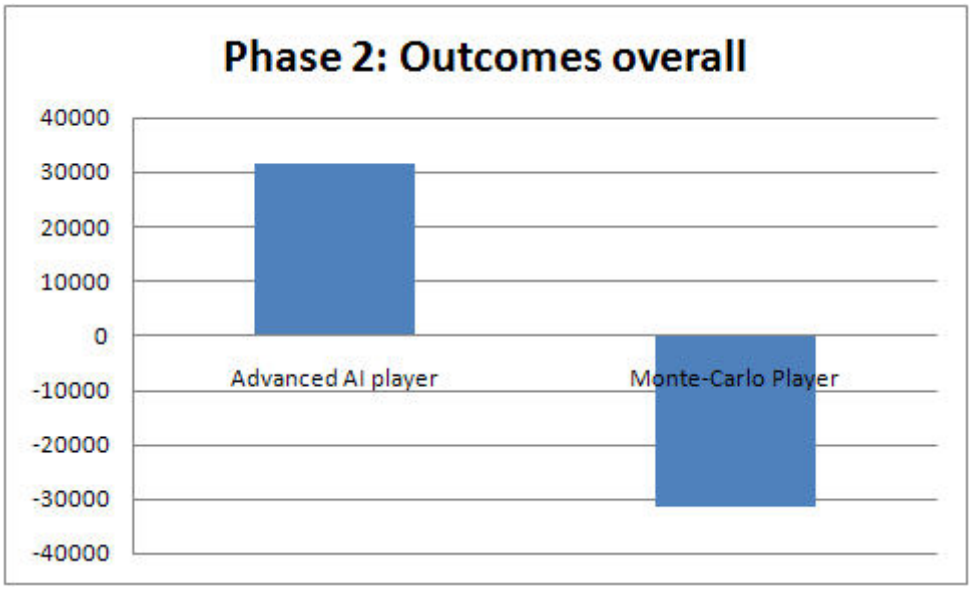


Figure 5.4: Tournament finals overall in chips.

Summary
In both phases of the tournament the advanced AI player won the game. The Monte-Carlo player was better in Rik and Solo 8 but those game types had a smaller reward than the game types Piek and Misère. This analysis is the answer to the last research question *RQ 4*.

Chapter 6

Conclusions

In this chapter we will give the conclusions of the research and future research possibilities. In Section 6.1 we give answers to the research questions and the problem statement. The conclusions that are drawn are based on the results of the experiments (Section 4.2), and the analysis of those results (Chapter 5). In Section 6.2 we give recommendations on future research possibilities.

6.1 Problem statement and research questions revisited

In Section 1.3 the following research questions were stated,

> *1. How can rule-based techniques be compared to Monte-Carlo techniques in the game of Rikken?*

In Chapter 3 we described four AI players that we developed for this research. One player bids and plays randomly and is only used for comparison. Two of the players use rule-based techniques to decide on game type in the bidding round and the cards to play during the playing round, and one player uses Monte-Carlo techniques to make those decisions. By letting the players compete against each other, the strength of the players can be determined. Comparing the strength of the players is a way to compare the two AI techniques.

> *2. How do the performances of the techniques in the bidding phase compare?*

In Subsection 4.1.1 we described the experiments that were designed to compare the accuracy of the bidding methods of the AI players. The accuracy was measured by comparing (1) the real number of tricks won during the game with (2) the number of tricks predicted by the bidding method of the AI player. The results of these experiments were presented in Subsection 4.2.1. In Section 5.1 these results were analysed. The results clearly showed that the 'Losing Trick Method' (see Subsection 3.3.1) was the most accurate bidding method. This method was developed for the advanced AI player.

> *3. How do the performances of the techniques in the playing phase compare?*

In Subsection 4.1.2 we described the experiments that were designed to compare the strength of the players during the playing phase. The strength is measured by distributing the same hands in 24 different ways among the four players. The bidding phase is skipped. The players all have to be declarer of one particular hand meaning every player becomes declarer 6 times. The results of these experiments can be found in Subsection 4.2.2. In Section 5.2 these results were analysed. It transpired that the advanced AI player with the rule-based AI techniques was the strongest player.

> *4. How do the performances of the techniques in both phases combined compare?*

In Subsection 4.1.3 we described the tournament in which the AI players competed against each other. In this tournament the AI players played both phases of the game. The results of the tournament were presented in Subsection 4.2.3. Section 5.3 analysed the results of the tournament. The player who won the most chips during the tournament was the advanced AI player. The Monte-Carlo player did win the game types Rik and Solo 8, but the advanced AI player won Piek and Misère which have a bigger payoff. The following problem statement was formulated for this research:

Can we determine whether rule-based techniques are more suitable than Monte-Carlo techniques to create an effectively and efficiently playing AI player in the game of Rikken?

For both techniques, players were developed who could play the game effectively and efficiently. From the experiments we learned that the advanced AI player had the best bidding method, the strongest play and that he won the tournament. The Monte-Carlo player had a better performance in Rik and Solo 8 during the tournament but the advanced AI player was stronger in the game types Piek and Misère which have a higher payoff. Therefore we may conclude that the rule-based techniques, used by the advanced AI player, are more suitable to create an effectively and efficiently playing AI player in the game of Rikken.

6.2 Future research

In this section we describe two ideas that can be used to continue this research. The first idea is to create an adaptable bidding strategy. The second idea is to design a player that is able to compute the maximum number of tricks that can be won when played with perfect knowledge.

The player that will use the adaptable bidding strategy will base his bidding decisions on similar hands that he has played before. The first step is to generalise card hands, in order to decide which card hands are simular. For every suit we determine the maximum number of tricks we believe we can win. One way to determine this number is using the trick counting technique of the basic AI player. The result gives us a certain combination, for example 2-1-0-3. Each number is the maximum number of tricks we believe we can win with a certain suit. This combination can be written as 3-2-1-0 because it does not matter in which sequence the suits are evaluated (all suits are equal). When all possible hands are evaluated and transformed to a high-to-low combination, we obtain 194 different combinations of generalised card hands.

The second step is to create databases that contain results from previously played games. Each database contains 194 entries, one for each generalised card hand. Each entry is divided into four, one for each game type. In each of these four subentries we keep track of the number of games played and the number of victories made. This information is used to calculate a winning percentage for that entry. For every combination of players, such a database can be created. The databases need to be initialised and this is done by playing many games. Each time a game is played, the matching entry is updated with a win or loss. After a certain amount of games, the player will only play when the database has a certain winning percentage for his hand of cards. This percentage keeps changing during the play. It gives the player a change to adapt to the strength of his opponents. When stronger opponents arrive the winning percentages will drop and the player will adapt by bidding less or when weaker players join, the player will try to play more games by bidding more.

The second idea is to create an external player, an Oracle, that has perfect information. With perfect information, other techniques like min-max search can be used to determine the optimal game strategy for each of the players. This could give us more information about the strength of the players. Comparing them to optimal play is a method different from comparing the players to the other players which is a relative comparison.

Next to these new ideas, there is space for a better implementation of the players and the game to increase speed and efficiency. Also better bidding methods can be developed for the game types Piek and Misère. One could also try to combine the knowledge of the advanced AI player with the simulations of the Monte-Carlo player.

References

Andrieu, Christophe (2003). Monte Carlo Methods for Absolute Beginners. *Advanced Lectures on Machine Learning*, pp. 113–145. [2]

Biethahn, Jrg and Nissen, Volker (1994). Combinations of simulation and Evolutionary Algorithms in management science and economics. *Annals of Operations Research*, Vol. 52, No. 4, pp. 181–208. ISSN 0254–5330. [1]

Bonnet, Christine, Taterode, Helene, Kouloumdjian, Jacques, and Hacid, Mohand-Said (1990). Using artificial intelligence techniques for intelligent simulation in memory re-education. *IEA/AIE '90: Proceedings of the 3rd international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pp. 874–881, ACM Press, New York, NY, USA. ISBN 0–89791–372–8. [1]

Cavazza, M. (2000). Al in computer games: Survey and perspectives. *Virtual Reality*, Vol. 5, No. 4, pp. 223–235. [2]

Cowan, Richard (1987). Hand Evaluation in the Game of Contract Bridge. Vol. 36, No. 1, pp. 58–71. ISSN 0035–9254. [16]

Gibbons, Robert (1992). *A Primer in Game Theory*. Pearson Higher Education. ISBN 0745011594. [1]

Ginsberg, M. L. (1999). GIB: Steps Toward an Expert-Level Bridge-Playing Program. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pp. 584–589. [2]

Knuth, Donald E. (1998). *The Art of Computer Programming Volumes 1-3 Boxed Set*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. ISBN 0201485419. [8]

Ligeza, Antoni (2006). *Logical Foundations for Rule-Based Systems*, Vol. 11/2006. Springer; 2 edition. ISBN 3–540–29117–2. [2]

Mooney, Christopher Z. (1997). *Monte Carlo Simulation*. Sage University Paper series on quantitative Applications in the Social Sciences, series no. 07-116, Thousand Oaks, CA. ISBN 0803959435. [23]

Rabin, Steve (2002). *AI Game Programming Wisdom*. Charles River Media, Inc., Rockland, MA, USA. ISBN 1584500778. [1]

Sint, Cees and SchipperHeyn, Ton (1994). *Bridge: Beter bieden, slimmer spelen. Bied- en speelvragen bij Van Start tot Finish 1 en 2*. Tirion. [2]

Sport, Tirion (2004). *Kaartspelletjes voor 3 of meer personen*. Tirion Uitgevers BV, second edition. [2, 28]

Urlings, Maurice (2006). Modelling task behaviour using hierarchical reinforcement learning. Master's project, Universiteit Maastricht. [1]

Wikipedia (2007). Game artificial intelligence — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Game_artificial_intelligence&oldid=99606050. [Online; accessed 12-January-2007]. [1]