Universiteit Leiden
The Netherlands

Opleiding Informatica

Determinization for Monte Carlo Tree Search

in the Card Game Tichu

Koen Castelein

Supervisors:
Walter Kosters
Jeannette de Graaf

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

20/07/2017

Abstract

Determinization and Monte Carlo Tree Search are both techniques that have been successfully used in card games such as Bridge. In this paper we use these techniques to create agents for the game Mini-Tichu, a simplified version of Tichu. Furthermore we try a novel approach for determining the opponents' cards by trying to predict them based on the current state of the game. We compare this method against regular random determinization.

Contents

Chapter 1

Introduction

Tichu [Wika] is an imperfect information four player card game that shares some aspects with Bridge and Poker. The game is played with two teams of two players. The goal of the game is to score more points than the opponents. The exact rules are explained in Chapter 2.

This paper will primarily compare basic Monte Carlo and Monte Carlo Tree Search agents that play a simplified version of Tichu we call Mini-Tichu. We will also try a new approach for the determinization of the opponent's cards. Determinization is the way by which we determine the unknowns before we execute Monte Carlo. For Tichu this means predicting the cards of the opponents. The simplest way of doing this is by giving away random cards. Players give various hints about what cards they may have during the game and have certain play styles. Using this information we can try to predict the cards in a way that is better than random.

Figure 1.1: Picture of the Tichu card game. Picture by C. Risher [Cla]

1.1 Thesis overview

This paper was written as a bachelor thesis for the Leiden Institute of Advanced Computer Science, the computer science department at the University of Leiden, with supervision from Walter Kosters and Jeannette

de Graaf. In this paper we discuss various Monte Carlo based agents for Mini-Tichu a simplified version of Tichu. First the rules for the game are explained in Chapter 2. Then the different agents are discussed in Chapter 4, along with the various choices that were made during implementation and some other related work. Chapter 5 expands on the experimental determinization method tested in this paper. Then we show the results we obtained in Chapter 6 and expand on them a bit. Finally, in Chapter 7, we conclude and discuss any future work that could be done.

Chapter 2

Rules

We will describe the games Mini-Tichu and Tichu. Since Tichu and Mini-Tichu share most of the rules, we will first explain the rules they have in common and later on expand on the differences between them. For simplicity we will describe a player as being a "he". Throughout the paper we will use Tichu to describe both Mini-Tichu and Tichu. We will make explicit distinctions where it is appropriate. Some of the rules in the official Tichu rulebook by Abacus Spiele [Aba] are open to interpretation, when the rules were not clear we followed the rules as described by Aaron D. Fuegi [Aar].

2.1 Common Rules

Tichu is a four player card game where the persons sitting opposite each other are together in a team. There are 56 cards divided into 4 suits of 13 cards, and 4 jokers called Mah Jong, Dog, Dragon, and Phoenix. The suits are called Star, Jade, Pagoda, and Sword. In this paper we will represent them using the symbols of the more commonly known suits respectively: ♢♡♣♠.

The goal of the game is to be the first to reach 1000 points. The game is also divided into rounds; at the start of each round the cards are shuffled and divided among the players. At the end of each round the points are counted and added to the score of each team. Also, similar to Bridge, rounds consist of tricks. Tricks in Tichu are started by playing a card combination; the player that starts puts down the cards he chooses to play. The player that has the Mah-Jong, one of the four jokers, begins the first trick. The following player then has to play a card combination with higher cards than the previous player, or pass. Passing is always allowed, even when you have a valid card combination to play. The trick ends when all remaining players except the one who played the last cards pass. This means you can even play after you have passed once and that a trick does not have a fixed length. The last player to play then wins all the cards in this trick and starts the next. The round ends when both players on a team finish their hand and this in turn implies that at some point in the game one or two players may not have any cards. When a player has no cards he automatically passes.

There are two kinds of winning in Tichu: A player can win and a team can win. A player is considered to be winning if he is the first player to lose all of his cards, while a team wins when they have more points than the other team. A losing player in turn is the player that finishes his hand last. Note that a team can win and not have the winning player, and the winning and losing player can be on the same team. If a team of players manages to finish before any player of the other team, they score 200 points; in this case the cards are not counted and the next round starts. Otherwise the losing player has to hand over all the cards he has won to the winning player. And all the cards that he has left in his hand go to the other team. Cards are then counted according to Table 2.1. The total number of points that can be gained in cards is 100. This results in fairly quick rounds, since there is a preference for scoring the 200 points instead.

card	score
5	5
10,king	10
dragon	25
phoenix	−25

Figure 2.1: Card values

The card combinations shown in Figure 2.2 and 2.3 are all the different types that are allowed in Tichu. Note that when playing straights or stairs the next player has to play one of exactly the same length. Four of a kind and straight flush are known as bombs and can even be played when other combinations are asked for. Bombs are always higher than any normal combination. A bomb that has more cards is always better than a smaller bomb and can be played over it as if it were a normal combination. One can also play a bomb with higher cards of the same length over another bomb just like normal play.

type	name	explanation	example
Normal combinations	single cards	Any single card.	5♣
	pairs	Two cards that have the same value.	8♢8♣
	stairs	Any number of consecutive pairs.	...
	three of a kind	Three cards that have the same value.	2♡2♢2♠
	full house	A combination of a pair and a three of a kind; only the three of a kind value is taken in consideration when determining which full house is higher.	4♡4♢4♣J♣J♠
	straights	Five or more consecutive cards.	4♠5♢6♢7♡8♣...
Bombs	four of a kind	All four cards of the same value, may not use the Phoenix.	3♣3♠3♡3♢
	straight flush	Five or more consecutive cards of the same suit, may not use the Phoenix.	4♢5♢6♢7♢8♢...

Figure 2.2: Valid combinations in Tichu

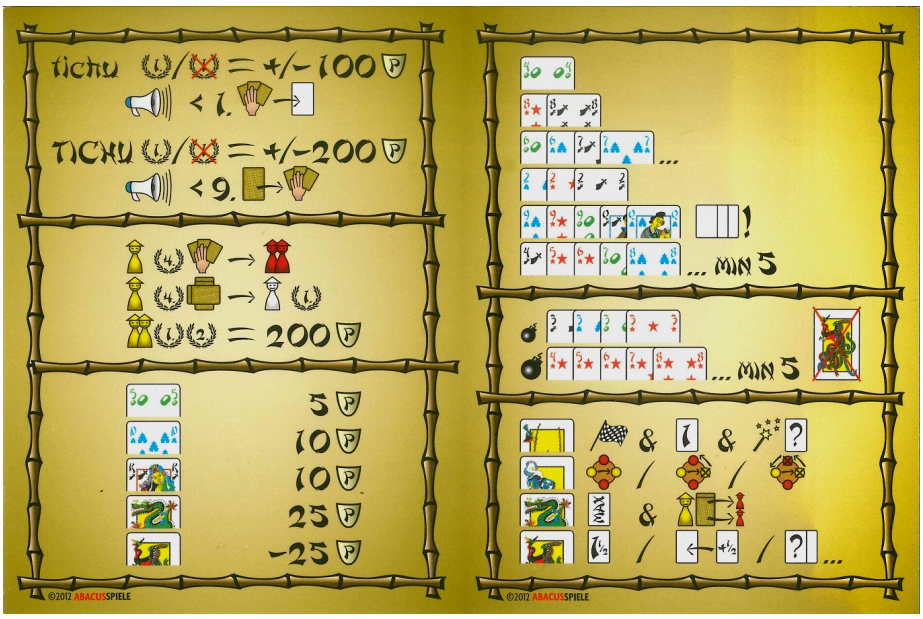

Figure 2.3: TICHU rebus rule card from the version of the game by Abacus Spiele [Aba]. Starting at the top left the following rules are explained: tichu/grand-tichu, what to do at the end of each round, card scores, normal card combinations, bomb combinations, and joker rules.

2.2 Jokers

The main difference between TICHU and MINI-TICHU is that all decisions not related to choosing which card to play are removed from MINI-TICHU. This causes the effect of some jokers to change slightly. The Dog and Phoenix remain unchanged, while the Dragon and especially the Mah-jong have extra rules.

Mah-Jong The player who has this card starts the game; it also acts as a 1 when playing.

> In TICHU the player that plays this card may make a wish. When making a wish a player can name any card value (2-ace) that is not a Joker. Until a card of this value is played, any player who can play a card of this value has to do so. The player is not allowed to pass in this case.

Dog When this card is played the turn goes to one's teammate; it can only be played at the start of a trick.

Dragon The Dragon is the highest single card and can only be played when single cards are asked for.

> In TICHU, if the trick is won with the Dragon, the player has to pick any player on the opposing team and give all the cards in the trick to him.

Phoenix The Phoenix can be used as a wild card in a combination; it can however not imitate other jokers. If it is played in tricks where a single card is asked, it is treated as being half a point higher than the last card played. If no card has been played yet, it counts as a value of .5, so even the Mah-Jong would be higher.

2.3 Rules unique to Tichu

In TICHU there are a number of rules that have nothing in common with MINI-TICHU. They are described as follows:

Tichu and Grand Tichu At any point during the game before a player has played any cards, he may decide to call Tichu, which is a bet of 100 points that he will be the first to finish his hand. There is also Grand-Tichu which has to be called while the cards are being divided, before a player gets his ninth card. This is a bet of 200 points that he will be the first to finish his hand. If a player called Tichu or Grand-Tichu and were right they win the points otherwise they lose them.

Trading cards At the start of each round after the cards are divided players have to give all of the other players one of their cards.

Bombs Players are allowed to play bombs even when it is not their turn.

2.4 Example of Mini-Tichu

This example starts during a game of MINI-TICHU. In this example the Mah-Jong has already been played so we know that no other players have it. At this point in the game we just won a trick. The situation is as follows:

North: 3 cards

Table: 0 cards

West:
6 cards

East:
8 cards

South (me):4♡4♣J♠K♣ Dr

Since we just won a trick we must now start a new one. We can choose any of the different card combinations we can play to start. At the moment we could play one of five different single cards or a pair of fours. We choose to play the pair so the following situation occurs:

North: 3 cards

Table: 4♡4♣

West:
6 cards

East:
8 cards

South (me):J♠K♣ Dr

Now the west player needs to play a pair of higher value than the pair of fours played by us, or play a bomb or pass. They choose to play: "7♢ Ph", in this case the Phoenix substitutes a 7.

North: 3 cards

Table: 7♢ Ph
2 cards worth 0 points

West:
4 cards

East:
8 cards

South (me):J♠K♣ Dr

Now the North Player, the East player, and I decide to pass. I did so because we could not play but both North or East could have had a higher pair. Since three players in a row passed the cards in the trick go to the West player and he starts a new trick. He now plays a stair with the cards: '5♡5♢6♠6♣'. This results in him having no cards and him winning as a player. All of the other players pass again. Since three players in a row passed again, the West player takes the cards from the trick. Since the West player doesn't have any cards left the next player (North) has to decide on the new trick.

The game continues on in this fashion until both the North player and we finish our hands, this means the East player is the losing player. The East player still has 3 cards remaining in his hand and has to give those cards to the opposing team, ours, the score of the cards is counted towards our total. The East player also has to give all the cards he got from winning tricks to the winning player, but since they are on the same team this does not affect the score of the teams.

After counting the total scores of each team it turns out that we have 60 points and the opponents have 40 points. This means we won the round as a team.

Chapter 3

Related Work

Combining Deliberation and Reactive Behavior for AI Players in the Mini-Tichu Card-game [VKV13] is a paper written by Martha Vlachou-Konchylaki and Stavros Vassos, that uses an alternative approach to create an agent. The paper proposes the use of a finite state machine to change the behaviour of the agents based on certain circumstances. The intention of this state machine is to provide guidance for agents until the search space becomes small enough for other agents like Negamax to take over.

Determinization and information set Monte Carlo Tree Search for the card game Dou Di Zhu [WPC11] is a paper by D. Whitehouse et al. about various agents for Dou di Zhu. Dou di zhu is a game similar to Tichu. The way in which they implemented MCTS with determinization inspired our implementation, especially for the selection phase of the algorithm.

Chapter 4

Agents

Since Tichu is an imperfect information game, it is not possible to use the default algorithm for MC and MCTS. To solve this we use determinization 5. Here we explain the agents and the different decisions that were made while implementing them.

4.1 Random

The implemented random agent plays completely random. Each possible play is equally likely. For example, suppose a player has 8♢8♡ as a hand. If the player is to start a trick he can either play a single eight or play both cards as a pair. In this case playing either eight is completely equivalent, but it still has a higher chance than playing the pair. We could consider to create a random agent that has an equal chance of playing all card combinations ignoring the suits of the cards. Using the example from above it would mean that the chance of playing either one of the eights as a single card is the same as playing them as a pair. We chose not to since it requires one to do a lot of checks and we did not expect any major improvements.

4.2 Basic Monte Carlo

Basic Monte Carlo (MC) [BPW+12] is an algorithm that uses random simulations of a game and compares the result of those simulations to find a good result. It does this by weighing the outcomes of all the randomly simulated games after a move has been made. It then selects a move by finding the move that results in the highest winrate. If there happen to be more moves with the same winrate we just use the first one that we found. This method is highly suited for finding good moves in games, since computers can easily simulate hundreds of games in a short time span.

In Tichu the winrate is not all that is important: the score can also be considered. We however chose to only consider the winrate since it empirically gave the best results.

4.3 Monte Carlo Tree Search

Monte Carlo Tree Search (MCTS) [BPW+12] is a modified version of Monte Carlo, that tries to select interesting moves to evaluate. It does this by creating a game tree while doing the playouts and by selecting interesting nodes to continue a playout from. It uses the steps detailed in Figure 4.1.

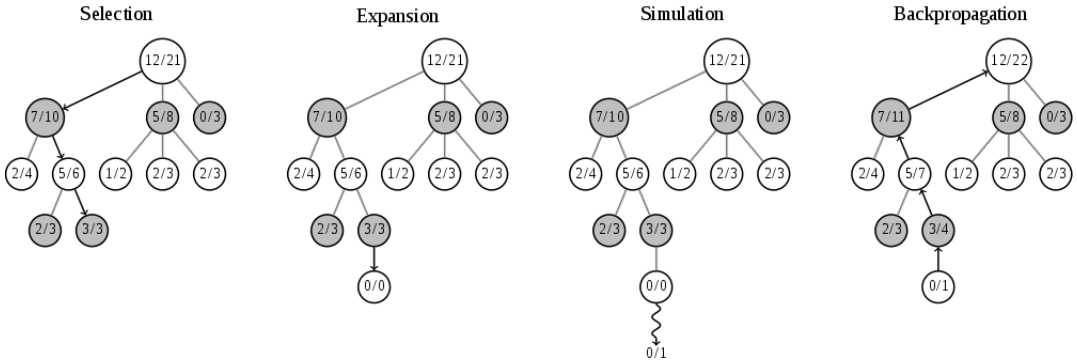

Figure 4.1: Steps of Monte Carlo tree search [Wikb]

Selection During the selection phase the algorithm considers each possible child node according to a predefined protocol and selects one of them. Normally this protocol gets applied recursively until a leaf node is found. However since we do not have perfect knowledge of the other players' cards, we also need to stop when a node has not been fully expanded. We do this because we often generate new cards for other players and they may have new plays available to them. In the same way plays may become unavailable, to solve this we only consider nodes that are currently possible with the hands of the players.

Expansion We fully expand the selected node with all possible plays at this point.

Simulation If the node is not a terminal node a playout is done starting at the selected node.

Backpropagation Use the result of the playout in the simulation phase to update all the parent nodes until the current state of the board is reached.

These steps are then repeated until we decide to stop the algorithm. In our case this is after a set number of playthroughs. With the selection protocol it is key to balance the exploration of new nodes and to verify how good the currently explored nodes are (exploitation). This implementation uses the formula recommened by Chang et al. [CFHM05]: $\frac{w}{n} + c * \sqrt{\frac{\ln t}{n}}$, where w_i is the total amount of times the ith child node resulted in a win. n_i is the total number of simulations done for the ith child node. c is the exploration parameter and equal to $\sqrt{2}$ for all our testing and t is the total number of simulations done for the parent node. In this formula the first term focusses on the exploitation, it makes it more likely to select nodes that were often won. The second term focusses on exploration: if a child node is played less often relative to other child nodes it is more likely to be selected.

Chapter 5

Determinization

Determinization is the way in which the unknowns in the current game state are determined. For Tichu this means determining the other players' cards. During this paper we refer to agents having determinization, when saying so we mean it has the determinization method described in this chapter. Otherwise when we say an agent does not have determinization the agent just has random determinization, where it is equally likely to get all cards. We want to improve on this random determization. To do this we try to predict some of the players' cards by comparing previous games played by the same agents to the current game. This is done by comparing the move we want to do a playout for, to previous games where a similar move was played. This information is then used to semi-randomly divide the unknown cards between the other players.

5.1 Information used

We store the following information on the moves agents make to use in future games:

Card combination We store the combination asked for in the played trick. We combine the two different bombs into one combination, since they are so rarely played. When players start a new trick we ignore it, since the combination is not necessarily the same as the previous move.

Turn We store the current turn the move was played in. A game of Tichu is on average about 70 turns long, but can theoretically last for around 200 turns. Since the game drives players to lose cards quickly, only the first 90 moves are stored. After this number of moves the information gained from this method is most likely near useless, since so few cards would be left.

Player We learn for every player at the same time, since it cannot be determined which player plays on the basis of which turn it is. We need to include this.

Has or has not The goal is to determine what cards the other players have. To do this we keep track of whether or nor other players can play a valid combination with their current hand. This information

could theoretically be obtained at the end of each game, so the agent does not use any information that it could not have known. This approach was chosen in favour of keeping track of whether the players passed or not, because it gives us more useful information. When comparing whether or not the other players passed, in this way, a lower chance of the opponents having a valid combination than they have in reality would be indicated. This happens because all players are always allowed to pass. Since we try to predict if the other players have a higher combination and not if they would pass, we use the first method.

This gives us a total of 7 combinations$\times$90 turns$\times$4 players$\times$2 possible states for a total of 5040 data points. We use these points to get a percentage chance for each player to have a valid combination.

5.2 Predictions

When doing MC or MCTS a player only knows his own cards. From these cards a move to play is determined according to the usual methods. The move is then used to determine the chances for the remaining players to have a valid combination higher than the last one played. We do this in the following manner:

First any cards the player knows other players to have, are assigned to the corresponding players. In Mini-Tichu this is only the Mah-Jong, as we know which player has this card since the player that starts the game has it. When playing Tichu a few more cards can be known because of the trading of cards at the start of the game, and a few cards could be excluded due to the "wish" from the Mah-Jong. If the player is the final of all players to pass, the next player must choose a new card combination to play. Trying to determine whether or not this player has a higher combination than the one we currently play makes no sense. So in this case we predict nothing and just fill the hands randomly. If the player is not the final of all players to pass, we determine if there remains any valid combination that is higher than the last one played in the game. If there is such a combination we get the chance from the dataset to determine whether we give one of these combinations to the next player.To finish determining all of the opponent's cards the remaining cards are filled out randomly. An overview of the process can be found in figure 5.1.

Since the cards are randomly dealt at the end it is possible for the next player to get one of the combinations even when we did not explicitly gave them one. This should lead to the players being able to play a little bit more often during MCTS than they could in reality.

Figure 5.1: Determinization process

Chapter 6

Experiments

Monte Carlo and MCTS can run indefinitely and therefore we have decided to limit the total amount of playouts they can consider each time one of the two agents decides on a move. The number of playouts is by default limited to 20 times the amount of moves the agent can make at one point. For some of the experiments we vary this, which will be noted at the individual experiments. Since Tichu is played in teams of two, each player can have a different agent. For these experiments players on the same team always have the same agent. Since a game of Tichu basically has numerous rounds, we have decided to only look at rounds and whether they are won by teams (by having more points than the opponents). All of the results were obtained by taking the average of 1000 rounds unless noted otherwise. For most of the experiments done we compare the winrate in percentages for the two teams. We find the winrate using the following formula: $100 \times \frac{\text{wins}}{\text{total games} - \text{ties}}$. We discard ties since they are only a very small portion of the results, namely around 3% of the games based on the agents.

6.1 Mini-Tichu experiments

In Table 6.1 we have the different winrates for each to win against one another which we try to improve with determinization. The figures around the diagonal should converge to exactly 50% over time, since players play against themselves and the starting player varies so no team has any advantage is this regard. In Figure 6.1 we compare how differing amounts of playouts effect the winrate of MC and MCTS against random. Here we can see that even when doing few playouts the results are significantly better than just random play. We also see that increasing the number of playouts gives increasingly smaller improvements.

	Random	MC	MCTS
Random	50.2	89.2	90.6
MC	10.8	50.4	50.6
MCTS	9.4	49.4	49.6

Table 6.1: Winrate without determinization (read as MC has a 89.2% chance to win against Random)

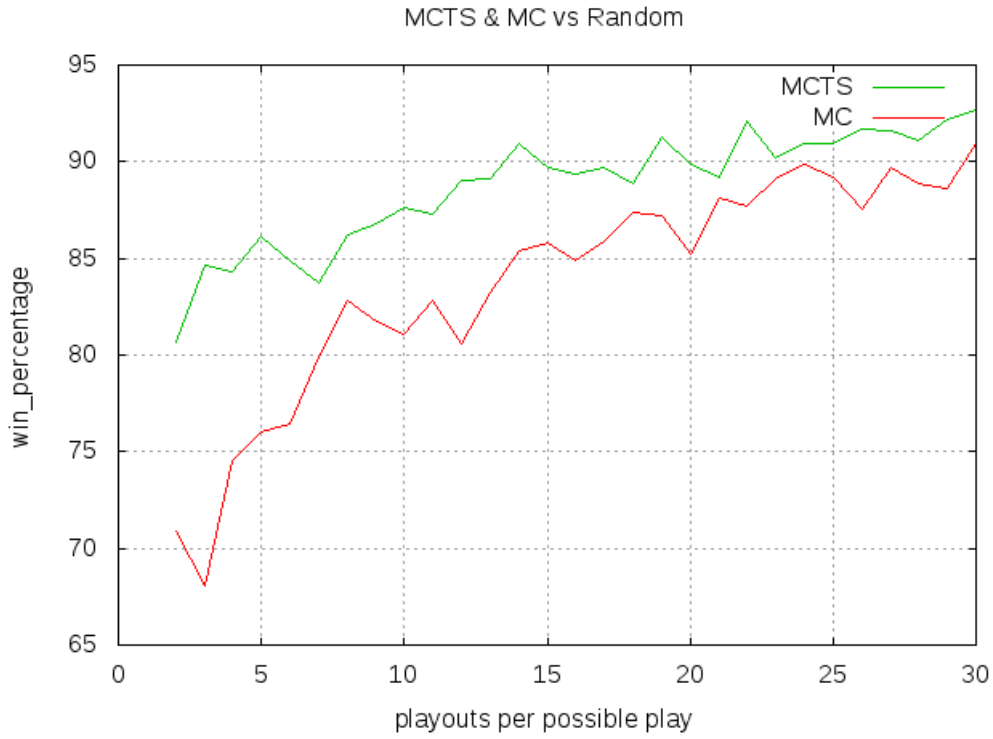

Figure 6.1: MCTS and MC vs random with varying number of playouts without determinization. The number of playouts done were varied to 20× the amount of possible plays.

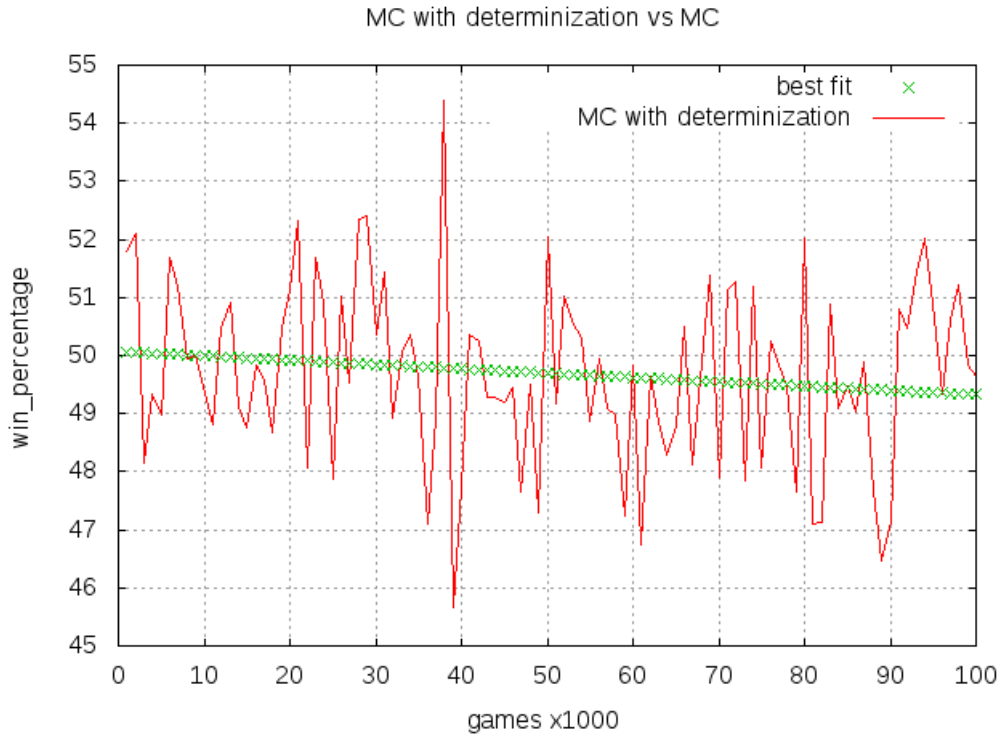

Figure 6.2: Winrate MC with determinization vs MC without determinization during training

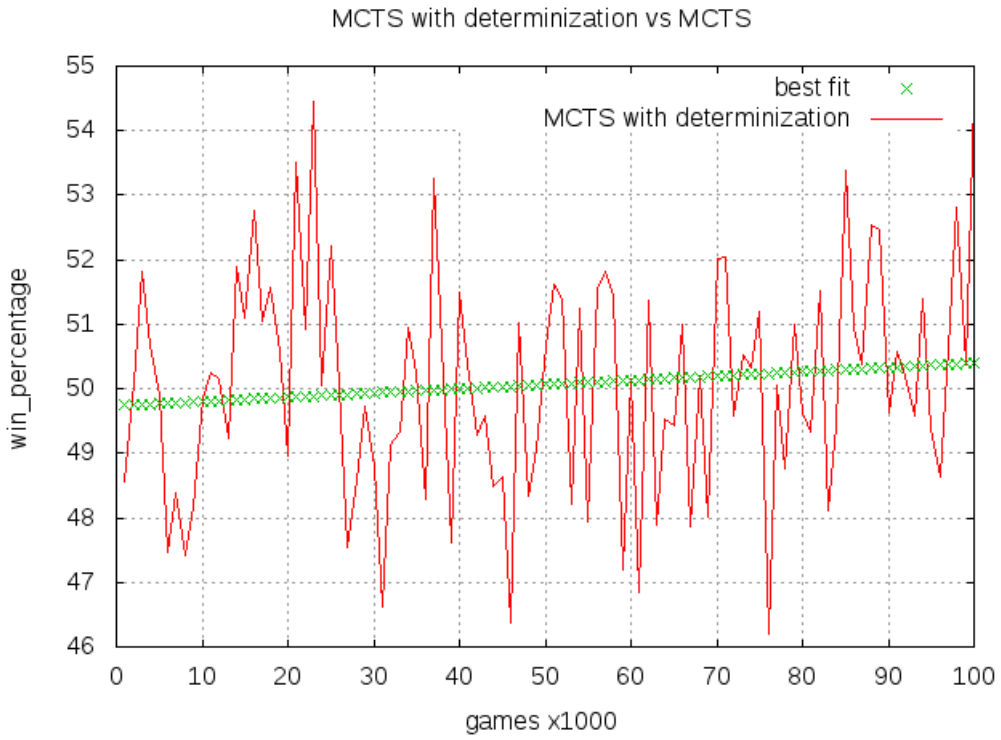

Figure 6.3: Winrate MCTS with determinization vs MCTS without determinization during training

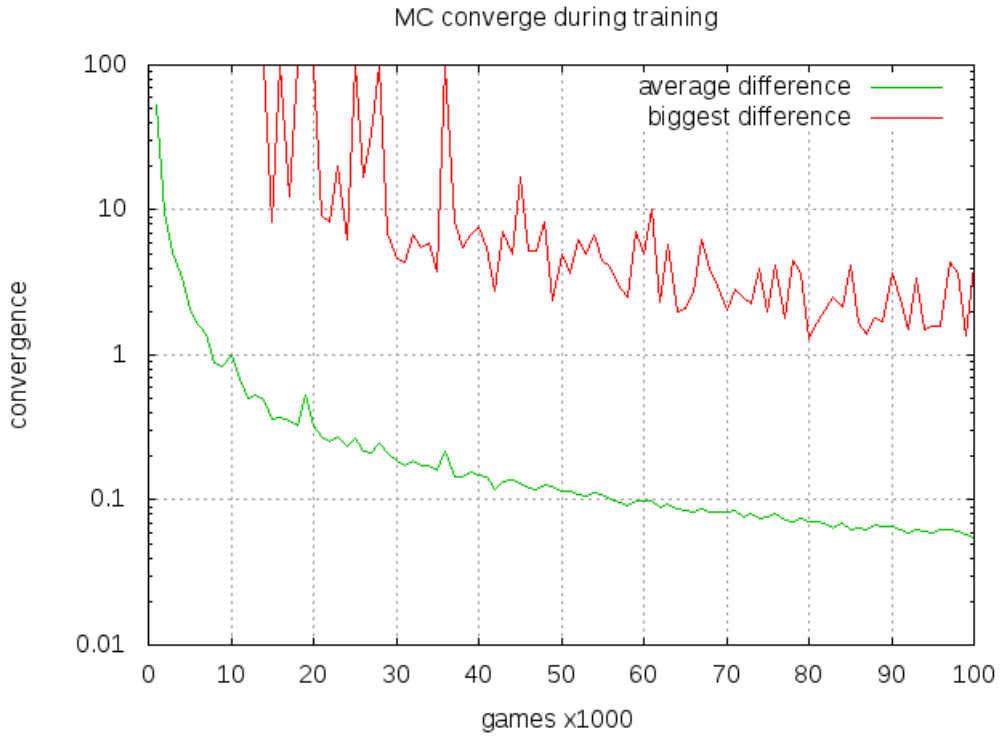

Figure 6.4: Difference in percentagepoints for all the chances we can predict for MC and MCTS.

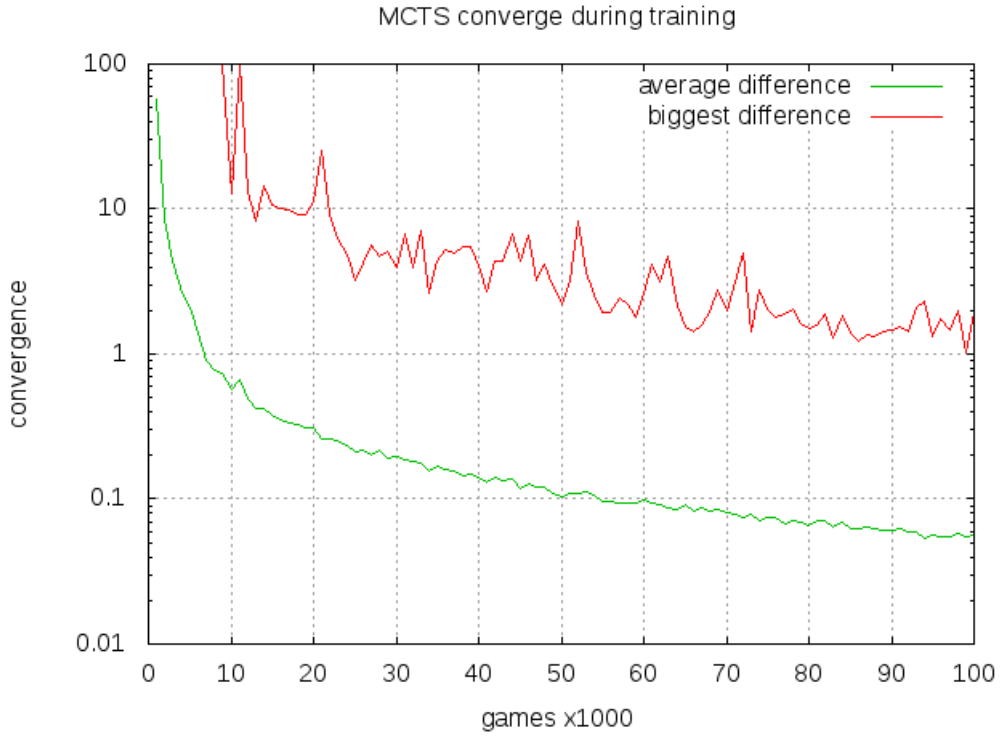

Figure 6.5: Difference in percentagepoints for all the chances we can predict for MCTS.

In Figures 6.2 and 6.3 we plot the improvement of the determinization algorithm during training. For every 1000 games that we play we plot the winrate for the agent with determinization. If the algorithm works we expect an increasing winrate over time. A best fit line was added to better illustrate what results would be expected at every point over time. With this we can see that even if there are any improvements they are minimal.

We know that there will not be any further improvements to the results because of what Figures 6.4 and Figure 6.5 show us. They show how much our predicted chances change during training. After only 10000 games the average predictions change by less than 1% and after about 60000 games this is just 0.1%. Based on this we expect that our predictions will not change. This leads us to believe that there will not be any major benefits when training any longer.

-	MC	MCTS
without determinization	87.3%	84.2%
with determinizaion	86.9%	88.2%

Table 6.2: Correctly predicted card combinations with and without determinization. Results were taken from over 1 billion predictions at various stages in the game.

The next question we should ask ourselves is whether this determization method even yields any improvements. In Table 6.2 we can see that random distribution of the cards already works for around 85% of the cases, and that determinization for MC is actually worse than just a random distribution. We attribute this to the fact that we predict agents to have card combinations slightly too often, which was discussed in Chapter 5.2. We see an

improvement for MCTS, however it seems that it is not significant enough to obtain any improvements in the winrate.

We tried to confirm that there was not any improvements by having MCTS with determization play against regular MCTS for 50000 games at the level of training it had after training for 100000 games. It won 24336 and lost 24212 times. The probability of winning this often or more, when we assume the winrate is 50%, is given by: $\sum_{i=k}^{n}\binom{n}{k})/2^n$ where n is the number of games played and k is the amount of times won. n in this case does not include the ties. This shows that we should expect to win this often about 28.8% of the time if the winrate where actually 50%. Since we win a bit more often than we lose the winrate might be a little be higher, suggesting that some improvement was made with the determinization. But since it is so minimal we would need to play many more games to confirm this.

All this leads us to conclude that this determinization in its current form does not seem to improve any of the tried agents for Mini-Tichu.

6.2 Tichu Experiments

Since MCTS has not been implemented for full Tichu the experiments below exclude it. We didn't implement it due to time constraints and we wanted to focus more on Mini-Tichu.

	Random	MC
Random	50.2	86.0
MC	14.0	52.1

Table 6.3: Winrate for Monte Carlo without determinization

In Table 6.3 we show the winrates for MC against Random for Tichu. We also tried applying the determinization method to Tichu, however we stopped the training earlier than for Mini-Tichu, since training after around 10000 games was found to be useless. The results are shown in Figure 6.6. They again do not seem to show any improvement nor any significant loss.

We see similar results when we compare Tichu with Mini-Tichu. If there is actually any effect due to the determinization we would expect it to be lessened for Tichu when compared to Mini-Tichu. This is due to the fact that there are many important decisions made in Tichu at the start of the game, where we cannot predict any cards. Namely swapping cards and calling Tichu or Grand-Tichu. This is because no tricks have been started at that point in the game and we can only predict cards when we know the card combination of the current trick.

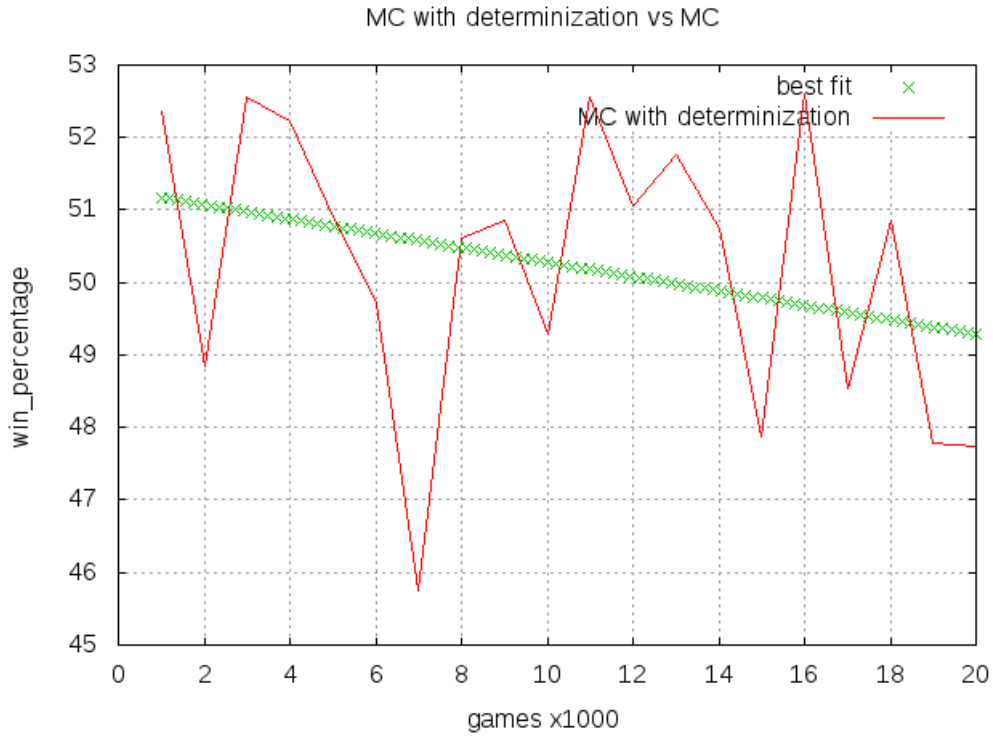

Figure 6.6: Winrate MC with determinization vs regular MC during training

Chapter 7

Conclusions and Future Work

We have shown that Monte Carlo based approaches are a viable method when creating an agent for Mini-Tichu and Tichu. The agents show significant improvement when compared to random play. However our way of determinization does not impact the results in any meaningful way as of yet. It shows some improvement for MCTS which leads us to believe that variations on this algorithm can be applied to obtain better results. Much more research remains to be done and some slight alterations to the algorithm might improve it a lot. For example, we might be able to look into predicting more cards, we could do this by predicting the other players' cards as well or just by predicting more cards for the next player. We could also try to incorporate more data in the prediction, the rank of the card combinations combinations for example. Any of these ideas might lead to better results it might however also impact the results negatively and future research will have to point this out. Different methods of determinization can be explored as well, we might be able to predict cards based on a neural network or even other methods.

More different implementations of Monte Carlo can be tried out as well. For example, different parallelisation techniques, such as leaf and root parallelisation could be tried. Or one of the MC variants discussed in the papaer by Browne et al. [BPW+12] could be implemented. Other types of agents need to be explored as well: rule based agents and Negamax agents could potentially be very good. Agents based around neural networks may be explored as well.

Bibliography

[Aar] Aaron D. Fuegi. Tichu. http://scv.bu.edu/~aarondf/Games/Tichu/. accessed on 15-7-2017.

[Aba] Abacus Spiele. The tichu card game. http://www.abacusspiele.de/spiele/tichu/. accesed on 18-7-2017.

[BPW+12] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[CFHM05] Hyeong Soo Chang, Michael C. Fu, Jiaqiao Hu, and Steven I. Marcus. An adaptive sampling algorithm for solving Markov decision processes. *Operations Research*, 53(1):126–139, 2005.

[Cla] Clarance Risher. Image of cards from a tichu game. https://www.flickr.com/photos/sparr0/3741349219/in/photostream/. accessed on 10-7-2017.

[VKV13] Martha Vlachou-Konchylaki and Stavros Vassos. Combining deliberation and reactive behavior for AI players in the Mini-Tichu card-game. *FDG*, pages 453–454, 2013.

[Wika] Tichu, *Wikipedia, the free encyclopedia*. https://en.wikipedia.org/wiki/Tichu. accessed on 20-7-2017.

[Wikb] Monte Carlo Tree Search, *Wikipedia, the free encyclopedia*. https://commons.wikimedia.org/wiki/File:MCTS_(English).svg. accessed on 10-7-2017.

[WPC11] Daniel Whitehouse, Edward Jack Powley, and Peter I Cowling. Determinization and information set Monte Carlo tree search for the card game Dou Di Zhu. In *2011 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 87–94. IEEE, 2011.