Measuring the Game State of Sichuan Mahjong

Songyan Jiang

Delft University of Technology

TU Delft

Measuring the Game State of Sichuan Mahjong

by

Songyan Jiang

Student Name	Student Number
Songyan Jiang	5726409

Instructor:	Robbert Fokkink
Project Duration:	December, 2023 - July, 2024
Faculty:	Faculty of Electrical Engineering, Mathematics & Computer Science, Delft

TU Delft

Contents

Abstract

Mahjong enjoys its status as the national game of China. The way it is played is philosophically described as: to create order out of chaos based on random drawings of tiles.

This study focuses on one specific type of mahjong: Sichuan mahjong. Sichuan mahjong is one of the most famous mahjongs in the world, and its unique rules lead to the two-player mahjong situation, which remains a gap in scientific research.

This report presents a Markov chain model for Sichuan mahjong, focusing on the quantitative measure of the game state. We took the combinatorial theory and algorithmic approaches to understand the game states. Based on the game state measure, we calculated the winning probability and expected number of game rounds in the two-player situation, and compared them with the experiment results. The results demonstrated the playability of Sichuan mahjong.

We also investigated the difference between aggressive and conservative players, and simulated the aggressive player's strategy in the calculation of winning probability and expected number of rounds.

The quantitative methods for game state measure contribute to applications in future mahjong AI, which provides the players with a broader understanding of Sichuan mahjong.

Keywords: Sichuan mahjong, Game state measure, Word number algorithm, Aggressive player, Two-player game

Acknowledgements

It has been 19 years since I was a student. I couldn't have overcome all the difficulties without the support of my teachers, and my friends, and my family.

I would like to thank the supervisor of my master's thesis, Dr. R.J. Fokkink, for providing me with great professional guidance, emotional support, and enjoyable weekly meeting during the process. I am also grateful to the professors who have taught me. I have greatly benefited from their lectures.

I would also like to extend the appreciation to my friends all over the world. I received lots of support from the Applied Mathematics group, which is a family of people from Netherlands, Italy, Greece, Germany, Chile, Portugal, Belgium, Sweden, and Austria. I am also deeply thankful to my Chinese friends here. We shared many wonderful times together.

Finally, I would like to express my sincerest gratitude to my family, especially my mom Xiuhong Liu, my father Dong Jiang, and my grandpa Daoyou Jiang. They have always supported my decisions unconditionally. I am profoundly grateful for their love.

Songyan Jiang
Delft, June 2024

1

The Game of Mahjong, History and Rules of The Game

1.1. The History of Mahjong

Mahjong, a traditional Chinese game, boasts a rich history spanning several centuries. Mahjong evolved from earlier Chinese card games and has undeniably become an integral part of Chinese culture, symbolizing both social interaction and strategic thinking. Over time, the game has transcended cultural boundaries, gaining popularity worldwide and influencing various regional adaptations.

1.1.1. Chinese Card Games before Mahjong

The earliest predecessors of card games in China can be traced back to the Tang dynasty (618 AD – 907 AD). Although no physical cards from this period have survived, historical records suggest the existence of ”leaf games” or ”ye zi”, which are likely the forerunners of later card games. These games are played with slips made from leaves or paper, which may have had numbers or instructions written on them.

By the time of the Song dynasty (960 AD – 1279 AD), there is evidence of more structured card games. The ”Yuanfeng Ji”, a historical record from this period, describes a game called ”yao qian”, which involves a set of 32 cards.

During the Ming dynasty (1368 AD – 1644 AD), a wide variety of card games became popular among all social classes. The most notable among them is ”ma diao”, a trick-taking game that bears some resemblance to modern card games. Some scholars suggest that mahjong may have developed from ma diao [5].

Ma diao is played with a deck of 40 cards, which typically comprises four suits with ten ranks each. The four traditional suits in ma diao are coins, strings of coins, myriads of strings, and tens of myriads. Each suit contains cards ranked from 1 to 9, plus a banner card (which is equivalent to the 10 in modern playing cards). The primary goal in ma diao is to win tricks, similar to many Western trick-taking games. The exact winning conditions can vary, but generally, the player who wins the most tricks is the winner of the game. Ma diao is typically played by four players. The deck is shuffled, and each player is dealt a hand of cards. The number of cards dealt depends on the specific variant being played. The player who leads the first trick is determined either randomly or by a predetermined rule. (S)he plays a card face up. Other players must follow the suit led if possible. If they cannot follow suit, they may play any card. The highest card of the led suit wins the trick unless a trump suit is in play. Play continues with the winner of each trick leading the next one.

The arrival of European traders and missionaries during the Ming and Qing dynasties introduced new card games to China. The adaptation of foreign card games led to a fusion of Eastern and Western gaming traditions, exemplified by games such as ”xiang pai”, a variation of bridge. The game of mahjong is also believed to be invented during this period of time.

1.1.2. The Development of Mahjong

Mahjong's precise origins have been lost to time. Writer Xu Ke traces mahjong to mid-nineteenth-century southern China, during the Taiping Rebellion [4]. Writer Du Yaquan suggests that mahjong grew out of the mixture of ma diao with various other tile games throughout the Qing dynasty (1644 AD - 1912 AD) [5]. He also locates the origins of modern mahjong in the first decade of the republican period. Du places mahjong's rise in southern China following the first opium war (1839 AD – 1842 AD) and the opening of treaty ports. In a fluid environment where "buyers from every province" gathered alongside drifters without permanent residences, the game of mahjong spread like wildfire. Mahjong soon gained considerable popularity all over the world in the early 20th century, notably in the United States during the 1920s, which marks a significant phase in its global spread.

In contemporary time, the advent of online gaming platforms has introduced mahjong to a broader, international audience, enabling cross-cultural interaction and digital preservation of the game. In 2020, mahjong's cultural significance was acknowledged with its inscription on UNESCO's representative list of the Intangible Cultural Heritage of Humanity.

1.2. Basic Rules of Sichuan Mahjong

Sichuan mahjong, also known as bloody mahjong, is a variant of the traditional Chinese mahjong that is particularly popular in the Sichuan province of China. This variant is known for its distinctive rules and characteristics. Here are the basic rules of Sichuan mahjong.

Sichuan mahjong is played with a set of 108 tiles. As shown in figure 1.1, there are 3 different types of mahjong tiles: dot, bamboo, and character. For each type, the tiles are numbered from 1 to 9, with each number having 4 identical copies. So in total, there are $9 \times 4 \times 3 = 108$ tiles.

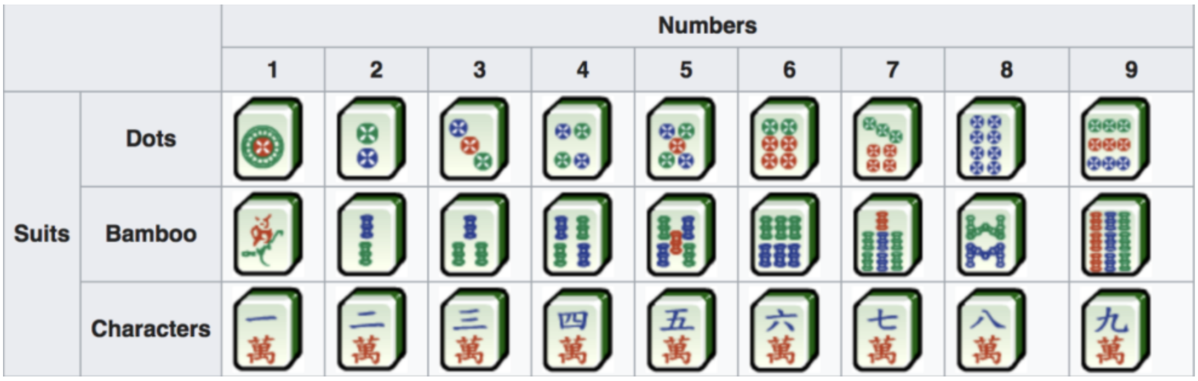


Figure 1.1: Sichuan Mahjong Tiles

For simplicity, we denote bamboo, character, and dot as B, C, and D respectively, and denote each mahjong tile as follows:

- Bamboos: $B1, B2, ..., B9$
- Characters: $C1, C2, ..., C9$
- Characters: $D1, D2, ..., D9$

Sichuan mahjong is a zero-sum game with imperfect information. In most cases, the game requires 4 players. The players play against each other. The aim for a player is to form a winning hand. Following the definition in [6], we give the definition of winning hand and other terminologies in Sichuan mahjong.

Definition 1. *A pair is two identical mahjong tiles.*

Definition 2. *A pong is three identical mahjong tiles.*

Definition 3. *A chow is a sequence of three consecutive mahjong tiles of the same type.*

Definition 4. *A meld is either a chow or a pong.*

Definition 5. *A kong is four identical mahjong tiles.*

When the player is able to form a kong and puts the 4 tiles down (that is called making a kong), (s)he receives extra points (1 point from each player) for the kong (s)he makes. In this case, a (put down) kong is considered as a pong when forming a winning hand.

Definition 6. *A winning hand is a 14-tile mahjong hand with 1 pair and 4 pongs or chows. The number of mahjong types in the winning hand cannot be more than 2 in Sichuan mahjong.*

For example, "$B1B1B2B3B4C1C1C1C2C2C2C7C8C9$" is a winning hand, where $B1B1$ is a pair, $B2B3B4$ is a chow and $C1C1C1$ is a pong. $B2B3$ or $C2C2$ is a pmeld.

We also have the pseudo version of the chow and the meld:

Definition 7. *A pchow is a sequence of two consecutive tiles of the same type.*

Definition 8. *A pmeld is either a pchow or a pair.*

Figure 1.2 provides a more direct explanation.

Figure 1.2: Sichuan Mahjong Terms

The tiles can be divided into the unknown tiles and the known tiles. The unknown tiles are tiles in the wall (see definition 9). The known tiles are either tiles in the player's hand or tiles on the table (see definition 10).

Definition 9. *The wall is the deck of remaining tiles on the table. The tiles in the wall are not known to the players. The players should draw from the right side of the wall every time.*

Definition 10. *Tiles that are discarded on the table by players are called table tiles. Table tiles are known to every player.*

The game is played with each player sitting on one side of a square table. At the beginning of the game, all players draw 13 tiles. Then they take turns to draw and discard a tile until they form a winning hand. The order is counter-clockwise. During the playing, suppose that player Alice has just finished her round of drawing and discarding, the player sitting on the right side next to Alice (because counter-clockwise) should start to play. Figure 1.3 gives an example of how the players are supposed to sit. The playing order in this figure is Alice - Bob - Carol - Dick - Alice and always the same.

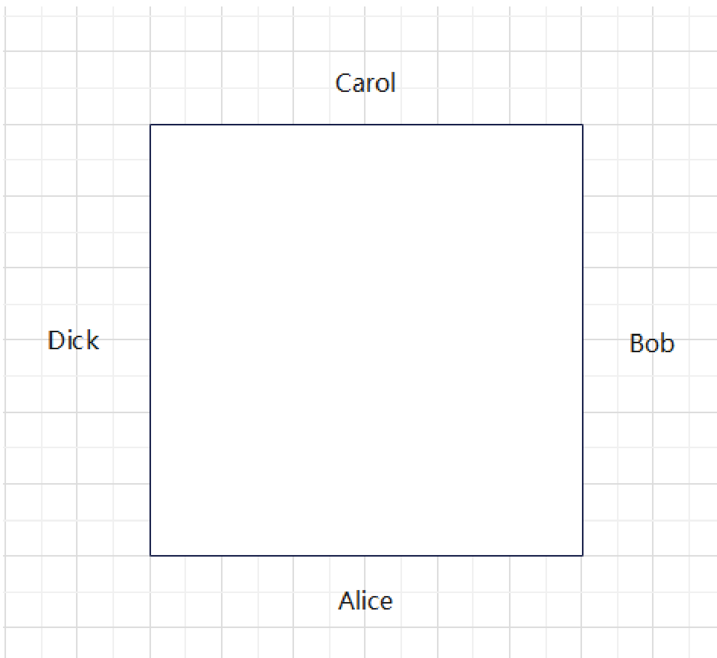


Figure 1.3: Players' Sitting Example

Suppose in the last game, Alice is the first player to form a winning hand, then in this game Alice is the first player to start. In Alice's round, she draws a tile from the right side of the wall and forms a hand of 14 tiles. If it is a winning hand, Alice wins 6 points and each of the other 3 players loses 2 points. Then Alice drops out of the game and the other players continue to play. For the rest of the game, Alice is not involved anymore. She will not win or lose any more points from the other players until the game ends. If Alice can not form a winning hand, she has to make decision to discard a tile from her hand.

After Alice has discarded a tile, the other 3 players also face a decision-making problem. Suppose that player Carol is able to form a winning hand by adding Alice's discard into her 13-tile phand (pseud hand), she can directly grab Alice's discard and claim to win. In this case Carol wins 1 point and Alice loses 1 point. However, Carol can choose to not grab Alice's discard. She can always wait until her next round and try to draw a tile and form a winning hand by herself, which will give her 6 points.

Besides forming a winning hand, the other 3 players can also make a pong or a kong by grabbing Alice's discard. Suppose that player Dick already has a pair of $D1$ in his phand and Alice happens to discard a $D1$ tile. Dick can make a pong by grabbing Alice's discarded $D1$ tile, putting down all the 3 $D1$ tiles and discarding another tile from his hand. Or suppose that player Dick already has 3 $D1$ tiles in his phand and Alice happens to discard a $D1$ tile. Dick can make a kong by grabbing Alice's discarded $D1$ tiles, putting down all the 4 $D1$ tiles, drawing another tile, and discarding a tile.

If neither of the other 3 players is able to make a pong, a kong, or form a winning hand by grabbing Alice's discard, the next player Bob, sitting right next to Alice, starts his round. The procedures in Bob's round are the same as Alice's. The game continues until either the mahjong tiles have been exhausted or 3 of the 4 players have formed a winning hand. Figure 1.4 shows the player action flow chart.

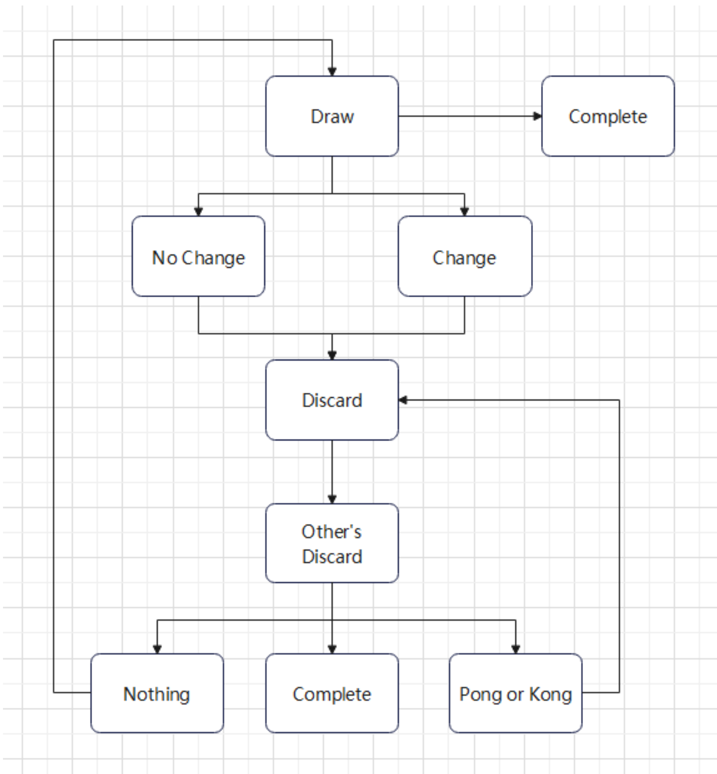


Figure 1.4: Player Action Flow Chart

Suppose this flow chart is designed for player Alice. In this flow chart, Alice first draws a tile from the right side of the wall, then she decides whether she can complete her hand. If she cannot, she should discard a tile. After this, Alice's round is over and the next player Bob starts to draw a tile from the right side of the wall and discard a tile.

After Bob's discard, Alice can decide if she can make use of Bob's discard. She can either complete a winning hand or form a pong or kong by robbing Bob's discard. After forming a pong by robbing Bob's discard, Alice has to discard another tile. After forming a kong by robbing Bob's discard, Alice has to draw one tile and discard one tile.

Besides the playing procedures, Sichuan mahjong has a special rule that is different from all the other versions of mahjong. The rule is called ”lack of a type”.

Definition 11. *Lack of a type means that at the beginning of the game, the player has to choose and declare a type that (s)he does not want to keep in his hand. During the game, he has to discard the type he declares first. When completing a winning hand, he cannot keep any tile of the type in the winning hand.*

2

Literature Review

The complexity of mahjong makes it an interesting subject for modern science. Our literature research on mahjong involves three perspectives: mathematics, computer science, and psychology. We give an overview of these literatures in table 2.1.

Table 2.1: Literature Overview

Name of The Paper	Perspective
Mathematical Aspect of The Combinatorial Game ”Mahjong” [1]	Mathematics
Let’s Play Mahjong [6]	Mathematics
An analysis of play style of ad- vanced mahjong players toward the implementation of strong AI player [11]	Computer Science & Psychology
Building a Computer Mahjong Player Based on Monte Carlo Simulation and Opponent Models [10]	Computer Science
Mass Mahjong Decision System Based on Transfer Learning [17]	Computer Science
A novel deep residual network-based incomplete information competition strategy for four-player mahjong games [14]	Computer Science
Official International Mahjong: A New Playground for AI Research [8]	Computer Science
Player Characteristics and Video Game Preferences [13]	Psychology
The Effects of Player Type on Performance: A Gamification Case Study [7]	Psychology

2.1. Mathematical Foundation in Mahjong

We introduce the two main papers which provide the theoretical foundation of mathematical application in mahjong. The paper ”Mathematical Aspect of The Combinatorial Game ’Mahjong’” [1] is the first paper to use mathematics in mahjong research. Inspired by this, the paper ”Let’s Play Mahjong” [6] researches on making the optimal decision based on mathematics.

2.1.1. The Combinatorial Theory of Mahjong

The paper ”Mathematical Aspect of The Combinatorial Game ’Mahjong’” [1] illustrates the combinatorial theory in playing mahjong. It uses generating functions to calculate the probability of forming certain types of winning hand, such as the ”nine gates”. I will discuss one of the main propositions of this paper, to give an idea of what type of combinatorial arguments are used.

The term "nine gates" refers to the phand "$X_1X_1X_1X_2X_3X_4X_5X_6X_7X_8X_9X_9X_9$" where X can be either bamboo, character, or dot. This phand is so interesting in that adding any mahjong tile of the same type would result in a winning combination. For example, suppose that we have a phand: "$B_1B_1B_1B_2B_3B_4B_5B_6B_7B_8B_9B_9B_9$", which is a nine gates. For any B_i tile ($1 \le i \le 9$), the combination "$B_iB_1B_1B_1B_2B_3B_4B_5B_6B_7B_8B_9B_9B_9$" is a winning hand. In some local mahjong rules (such as Hongkong mahjong rule), the payoff is extremely high when the player reaches this nine gates combination.

The following proposition is the main result from this paper. It consists of 6 parts. Part (a) states that consider only one type of mahjong tile (let's say bamboo) with in total $9 \times 4 = 36$ tiles. The number of ways to choose 13 random tiles from 36 tiles is:

$$\binom{36}{13} = 2310789600$$

Part (b) states that according to the multiplication principle, the number of ways of getting 3 B_1 tiles and 3 B_9 tiles and 1 B_i tile for $2 \le i \le 8$ each is:

$$\binom{4}{3}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{3} = 262144$$

Part (c) states that to have a nine gates is almost a miracle. The probability of getting the nine gates phand is:

$$\frac{\binom{4}{3}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{1}\binom{4}{3}}{\binom{36}{13}} = 0.00011344347$$

Part (d) states that the number of selections of m tile out of the 36 tiles is the same as the number of the number of degree m terms in the expansion:

$$\begin{aligned}&(1 + X_1 + X_1^2 + X_1^3 + X_1^4)(1 + X_2 + X_2^2 + X_2^3 + X_2^4)\cdots(1 + X_9 + X_9^2 + X_9^3 + X_9^4)\\ &= \sum_{0 \le n_1 + \cdots + n_9 \le 36} X_1^{n_1} X_2^{n_2} \ldots X_9^{n_9}\end{aligned}$$

where $X_1X_1X_1$ is expressed as X_1^3 and so on.

Based on part (d), part (e) states that the number of different 13-tile phands of the same type equals to the total number of summands $X_1^{n_1}X_2^{n_2}\cdots X_9^{n_9}$ with $n_1+n_2+...+n_9 = 13$ in the equation:

$$(1 + X_1 + X_1^2 + X_1^3 + X_1^4)(1 + X_2 + X_2^2 + X_2^3 + X_2^4)\cdots(1 + X_9 + X_9^2 + X_9^3 + X_9^4)$$

The total number of summands is also the same as the coefficient of X^{13} in the expansion:

$$(1 + X + X^2 + X^3 + X^4)^9 = \sum_{i=0}^{36} \alpha_i X^i$$

The result is $\alpha_{13} = 93600$.

Part (f) talks about the duality. X_i and X_{10-i} are dual tiles. The phand "$X_{j_1}X_{j_2}\ldots X_{j_r}$" has a dual of "$X_{10-j_1}X_{10-j_2}\ldots X_{10-j_r}$". The reason is that "$X_1X_1X_2X_3X_4X_5X_6X_7X_8X_9$" are symmetric around X_5. The two tiles with their subscripts adding up to 10 are said to be symmetric. Thus every single phand of 13 tiles has a dual phand if X_k is replaced by X_{10-k}. The tile that the dual phand can win is symmetric (around X_5) to the tile that the original phan can win.

Besides "Nine Gates", the probability of getting "Eight Gates", "Seven Gates"... can also be calculated. The "Nine Gates" is the phand requiring anyone of the 9 tiles $X_1, ..., X_9$ to form a winning hand (we call this winning 9 tiles), whereas the "Eight Gates" is the phand winning 8 tiles, the "Seven Gates" is the phand winning 7 tiles...

The study is the first attempt to apply mathematical techniques to mahjong. Its results contribute to further research in mahjong AI.

2.1.2. Let's Play Mahjong

Inspired by the previous paper, the paper "Let's Play Mahjong" [6] introduces the notion called "deficiency", which is used to define the distance from winning. Based on the deficiency, the paper provides a theoretical approach to make the optimal decision.

Definition 12. *The deficiency of a mahjong hand is the minimum number of tiles needed to change in order to form a winning hand in mahjong.*

For example, the hand "$B1B1B1B2B2B2B4B4B4B9B9C1C2C8$" has 1 deficiency, because changing $C8$ into $C3$ will complete the hand.

In order to compute the deficiency in an efficient way, the paper introduces a quadtree algorithm, which computes deficiency for a hand. The quadtree algorithm is based on the notion of pre-decomposition.

Definition 13. *A pre-decomposition of a mahjong hand is a sequence π of five sequences, $\pi[0], \ldots, \pi[4]$, s.t.*

- *$\pi[4]$ is a pair, a single tile, or empty;*
- *for $0 \le i \le 3$, each $\pi[i]$ is a meld, a pmeld, a single tile, or empty.*

The pre-decomposition π is said to be complete if $\bigcup_{i=0}^{4} \pi[i]$ contains all of the 14 tiles. For a mahjong hand H, the set of tiles in H that are not in $\bigcup_{i=0}^{4} \pi[i]$ is said to be the remainder of H under π.

Definition 14. *Suppose π and π' are two pre-decompositions. We say π' is finer than π if, for each 0 ≤ i ≤ 4, $\pi[i]$ is identical to or a subsequence of $\pi'[0]$. A pre-decomposition π is completable if there exists a decomposition π' that is finer than π. If this is the case, we call π' a completion of π.*

The cost of a pre-decomposition π, written as $cost(\pi)$, is the number of missing tiles required to complete $\pi[4]$ into a pair and complete $\pi[i]$ into a meld for $0 \leq i \leq 3$. The cost of pre-decomposition is closely connected to the deficiency of a hand.

Theorem 1. *The deficiency of a 14-tile mahjong hand is the minimum cost of its pre-decompositions.*

The proof of this theorem can be found in [6]. Although it is hard to reproduce the proof, the result forms the basis of our investigation into mahjong. In our report, the playing process is simplified into 2-player mahjong, which can be modelled by a Markov chain.

The quadtree method calculates the deficiency of a 14-tile mahjong hand by constructing all of its possible pre-decompositions. The initial value of the minimal pre-decomposition cost starts from 6, which is equal to the largest deficiency number. For each pre-decomposition, if its cost is less than the current minimal pre-decomposition cost, the minimal pre-decomposition cost is updated as this pre-decomposition cost. After exhausting all possible pre-decompositions, the minimal pre-decomposition cost is taken as the deficiency of the mahjong hand.

The paper also introduces the decision-making methodology, which is based on the notion of deficiency, when the mahjong hand is not complete.

The knowledge base w is defined as the player's knowledge of available tiles. $w(c, n)$ denotes the number of tiles available for tile (c, n), where c = 0,1,2 means bamboo, character, and dot respectively. n = 1, ..., 9 means the tile number.

The 14 tiles in a player's hand are denoted as $T[i]$ with $0 \leq i \leq 13$. For each tile $T[i]$, its value $\delta_{T,\omega}(i)$ is defined as

$$\delta_{T,\omega}(i) = \sum_{0 \leq c \leq 2, 1 \leq n \leq 9} \{\omega(c, n) \mid \text{dfncy}(T[i/(c, n)]) < \text{dfncy}(T)\}$$

where $i/(c, n)$ means replacing tile $T[i]$ with tile (c, n) and $dfncy(T)$ means the deficiency number of hand T.

For each tile $T[i]$, $\delta_{T,\omega}(i)$ denotes the total number of available tiles (c, n) such that hand $T[i/(c, n)]$ has a smaller deficiency than T, i.e. the total number of tiles available that are better (in deficiency) than $T[i]$.

One way of choosing the tile to discard is to pick the $T[i]$ that has the maximum value of δ, i.e.,

$$\text{discard}(T, \omega) = \underset{T[i]:0 \le i \le 13}{\arg\max}\, \delta_{T,\omega}(i).$$

where arg max means choosing the argument $T[i]$ such that $\delta_{T,\omega}$ reaches the maximum value.

Another way to discard $T[i]$ is to consider the best chance for completing (winning) T within k tile changes.

The paper defines the step k value of the i-th tile of T w.r.t. ω, written $\text{val}_k(T, \omega, i)$, as the chance of completing T within k tile changes if $T[i]$ is discarded first.

The step 0 value of T is defined as

$$\text{val}_0(T) = \begin{cases} 1, & \text{if } T \text{ is complete (winning hand)} \\ 0, & \text{otherwise} \end{cases}$$

The step k value of T w.r.t. ω is defined as

$$\text{val}_k(T, \omega) \equiv \max_{0 \le i \le 13} \text{val}_k(T, \omega, i).$$

where $\text{val}_k(T, \omega, i)$, which is the step k value of tile $T[i]$ w.r.t. ω, can be recursively calculated in the following way:

$$\text{val}_1(T, \omega, i) = \sum_{(c,n):\omega(c,n)>0} \frac{\omega(c,n)}{\|\omega\|} \times \text{val}_0\big(T[i/(c,n)]\big).$$

$$\text{val}_k(T, \omega, i) = \sum_{(c,n):\omega(c,n)>0} \frac{\omega(c,n)}{\|\omega\|} \times \text{val}_{k-1}\big(T[i/(c,n)], \omega - (c,n)\big)$$

$\|\omega\| \equiv \sum\{\omega(c,n) \mid 0 \le c \le 2, 1 \le n \le 9\}$ is the total number of available tiles. $\frac{\omega(c,n)}{\|\omega\|}$ is the chance of selecting (c, n) to replace $T[i]$. $\omega-(c, n)$ denotes the knowledge base obtained by decreasing $\omega(c, n)$ by 1, as the tile (c, n) has been used for replacing $T[i]$ with it.

As is shown in the formula, if $T[i]$ is discarded first, the chance of completing T within 1 tile change is the sum of chances of getting a new tile (c, n) that can complete T, which is the weighted sum of chances of completing $T[i/(c, n)]$ within 0 tile changes. The chance (i.e. the probability of drawing a certain tile from the wall) of completing T within k tile changes is the weighted sum of chances of completing $T[i/(c, n)]$ within k−1 tile changes.

For choosing the best tile to discard first, define

$$\text{discard}_k(T,\omega) = \underset{T[i]:0\leq i\leq 13}{\arg\max}\, \text{val}_k(T,\omega,i)$$

Discarding $T[i]$ = discard$_k(T,\omega)$ first has the best chance for completing T within k tile changes. So $T[i]$ = discard$_k(T,\omega)$ should be discarded first.

2.2. AI in Mahjong

AI has been very successful in the analysis of games like chess and go, where the computer easily beats the strongest human player. AI has been less successful in card games such as bridge, poker, and mahjong.

The research of mahjong AI can be divided into two perspectives: the opponent model and the self-perspective research [16]. The papers ”An analysis of play style of advanced mahjong players toward the implementation of strong AI player” [11] and ”Building a Computer Mahjong Player Based on Monte Carlo Simulation and Opponent Models” [10] belong to the opponent model research. They try to collect information from the opponents to make better decisions. The papers ”Mass Mahjong Decision System Based on Transfer Learning” [17], ”A novel deep residual network-based incomplete information competition strategy for four-player mahjong games” [14], and ”Official International Mahjong: A New Playground for AI Research” [8] focus on the self-perspective research.

In the opponent modeling research, ”An analysis of play style of advanced mahjong players toward the implementation of strong AI player” [11] tries to solve the multi-party game problem by proposing a strategy-adopting model that matches the opponent players’ strategies. The innovative ideas of this paper promote the future mahjong AI research. When studying opponent models, we can assess their strategies based on their behaviors and make corresponding decisions accordingly.

The paper ”Building a Computer Mahjong Player Based on Monte Carlo Simulation and Opponent Models” [10] also presents a method for developing a mahjong-playing AI that uses opponent models and Monte Carlo simulations. The approach involves predicting three aspects of an opponent’s play: waiting to win, winning tiles, and winning scores. These predictions are made using logistic and linear regression models trained on game records of expert players. The AI uses these models to guide its moves during Monte Carlo simulations. The resulting AI was evaluated on an online mahjong platform, achieving a rating significantly higher than the average human player.

In the self-perspective research, the paper ”Mass Mahjong Decision System Based on Transfer Learning” [17] proposes a mahjong decision system based on transfer learning, which is a machine learning technique where a pre-trained model on one task is adapted to improve performance on a related but different task, to address the issues of data scarcity and difficulty in constructing models. The design of the decision-making system includes the following main parts:

- Pre-training: Selecting a well-performing model on a similar task for pre-training.
- Weight transfer: Using the trained model weights as initial weights for the new task, which is about removing specific features for retraining.
- Model optimization: Fine-tuning the mahjong discard model through self-playing, generating data, and updating the model to better adapt to mahjong rules.

The paper ”A novel deep residual network-based incomplete information competition strategy for four-player mahjong games” [14] introduces a novel competition strategy based on deep residual networks (ResNet) specifically designed for the four-player mahjong game. The new competition strategy was compared with several shallow learning and deep learning methods, demonstrating superior qualitative and quantitative performance.

The paper ”Official International Mahjong: A New Playground for AI Research” [8] explores the potential of mahjong as a testbed for AI research and summarizes related competition activities. The competitions, which are the first Mahjong AI competition [2] and the second Mahjong AI competition [3], attracted teams from universities and companies, using various algorithms including heuristic methods, supervised learning, and reinforcement learning to develop their AI agents.

The results show that modern game AI algorithms based on deep learning, such as supervised learning and reinforcement learning, significantly outperformed heuristic methods based on human knowledge. Reinforcement learning methods performed the best in the competitions, using stable policy gradient algorithms like Proximal Policy Optimization (PPO). Despite the impressive performance of AI in the competitions, it still could not surpass top human players in a human-versus-AI match.

2.3. Game Player Analysis

The psychology of mahjong players has been studied in several papers. The papers ”Player Characteristics and Video Game Preferences” [13], ”The Effects of Player Type on Performance: A Gamification Case Study” [7], and ”An Analysis of Play Style of Advanced Mahjong Players Toward the Implementation of Strong AI Player” [11] focus on different aspects when analyzing the game players.

The paper ”Player Characteristics and Video Game Preferences” [13] explores how player traits and other factors influence video game choices. This study investigates how different player traits, game elements, and playing styles explain video game preferences. It finds that certain player traits are strongly associated with preferred games. For instance, players with high social orientation favored multiplayer games like ”Call of Duty” and ”Counter-Strike,” while those with lower social orientation preferred single-player games like ”Civilization” and ”The Sims”. Mahjong players are mostly with high social orientation. Besides, preferred game elements and playing styles were also related to specific game choices. The competitive and interactive environment is crucial for social orientation. Games like ”Defense of the Ancients” and ”FIFA” are popular among players with high social orientation.

The paper ”The Effects of Player Type on Performance: A Gamification Case Study” [7] explores how an individual's Hexad player type influences their performance in gamified applications. The study examines the effects of an individual's Hexad player type on their performance in gamified applications. The Hexad is a framework that introduces six player types: (i) Philanthropists, (ii) Disruptors, (iii) Socialisers, (iv) Free Spirits, (v) Achievers, and (vi) Players. They are summarized in table 2.2.

Table 2.2: Hexad Player Type

Type	Description
Philanthropists	Motivated by purpose and meaning. These individuals often display altruistic behavior and are willing to give without expecting a reward.
Disruptors	Motivated by change. They enjoy challenging systems, testing their limits, and causing disruption.
Socialisers	Motivated by social connections and relationships. They thrive on interaction with others and building social networks.
Free Spirits	Motivated by autonomy and self-expression. They value freedom, exploration, and self-discovery.
Achievers	Motivated by competence and mastery. They aim to complete tasks, achieve goals, and tackle challenges to demonstrate their skills.
Players	Motivated by external rewards. They are driven by points, badges, and other tangible rewards, often focusing on achieving these rewards rather than the activities themselves.

In this paper, a ”gamified application” refers to an application that incorporates game-like elements and mechanics to engage users and motivate them to complete certain tasks or achieve specific goals. By comparing a gamified application with a non-gamified application, the study aims to determine how different player types respond to gamified environments and whether these game-like features improve performance. It finds that player type correlates with individuals' perception of game elements and performance in the gamified application. After controlling for player type, participants who interacted with the gamified application performed better than those with the non-gamified application.

However, the Hexad framework is mostly used in online computer games. For strategy games like poker and mahjong, this frame is not applicable. Our report chooses the player type metioned in the next paper.

The paper ”An Analysis of Play Style of Advanced Mahjong Players Toward the Implementation of Strong AI Player” [11] explores advanced mahjong players' behaviors and strategies to implement a strong AI player. It classifies advanced Japanese mahjong players' behaviors into four types.

1. Neutral Players:
    - These players maintain a balanced approach, neither too aggressive nor too defensive.
    - They tend to adjust their strategy according to their hand's development but avoid taking excessive risks.
    - For example, if their hand has some advantages, they might take a moderately offensive strategy. However, when the risk is high, they tend to play defensively, discarding safer tiles.
2. Aggressive Players:
    - These players are inclined to take risks to achieve higher scores, often opting for bold offensive strategies.
    - They prioritize completing their hand even if it means taking a risk to give their opponents useful tiles.
    - For example, if they are one tile away from a winning hand, they discard high-risk tiles to complete their hand as quickly as possible, despite potential threats from opponents.
3. Defensive Players:
    - These players are cautious and avoid providing opponents with any opportunity to complete their hand.
    - They often analyze other players' discarded tiles to deduce potential risks and act accordingly.
    - For example, if they sense an opponent is close to a winning hand, they will avoid discarding tiles that could potentially complete that hand.
4. Adaptive Players:
    - These players exhibit flexibility, adjusting their strategy depending on the game's progress.
    - They switch between offensive and defensive approaches based on their hand's potential and the current game situation.
    - For example, if they have a strong hand, they may adopt an offensive approach. However, if the game reaches a critical stage where other players appear to be close to winning, they switch to a more defensive style, focusing on avoiding risky discards.

3 Sichuan Mahjong Expert Analysis

The research of this report starts with the analysis of Sichuan mahjong experts. These experts are players who have at least 5 years of playing experience. They master the rules and complex strategies. Different types of players have different playing strategies. To delve deeper into the player types and the experts' playing strategies, we sent a questionnaire survey to these experts, and analyzed the results in this chapter. Based on the outcome of the questionnaire, we divided the players into types, that we took from the psychological study [11]. We singled out only 2 types based on the statistical analysis, and compared their playing strategies.

3.1. An Analysis of Player Types

We designed questions to investigate the experts' playing types. The 8 questions use the five-point Likert-type scale:

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

A higher level of scale score represents a higher level of conservation in playing Sichuan mahjong.

The questions are:

(1) I observe other players' emotions and movements to gather information.
(2) I discard tiles I need to prevent others from winning.
(3) I remember key tiles played by other players.
(4) I often calculate tiles in my opponents' hands.
(5) When my hand is not powerful, I try to prevent other players from scoring.
(6) When the game is not going well for me, I will adjust my strategy.

(7) I choose to play seemingly insignificant tiles to confuse my opponents.
(8) I won't take risks to quickly win even if my hand is in an advantageous state.

A sample of 30 participants took part in the study. The participants are all experienced Sichuan mahjong experts with more than 5 years' experience. These experts are selected from students at TU Delft. Questionnaires were sent via email and WeChat. Participants were asked to answer all questions completely.

Cronbach's alpha [12] is a measure of internal consistency, which assesses the reliability of a scale or test. It is calculated as:

$$\alpha = \frac{N}{N-1}\left(1 - \frac{\sum_{i=1}^{N} \sigma_i^2}{\sigma_t^2}\right)$$

where N is the number of items, σ_i^2 is the variance of the i-th item, and σ_t^2 is the total variance of the sum of all items (or the total test score). In the current survey, Cronbach's alpha is 0.82, indicating a high level of internal consistency within the survey.

In order to have a detailed categorization of players, we conducted a clustering analysis on the questionnaire. In this report we choose the k-means clustering method. We use the elbow method [9] to determine the optimal clustering number.

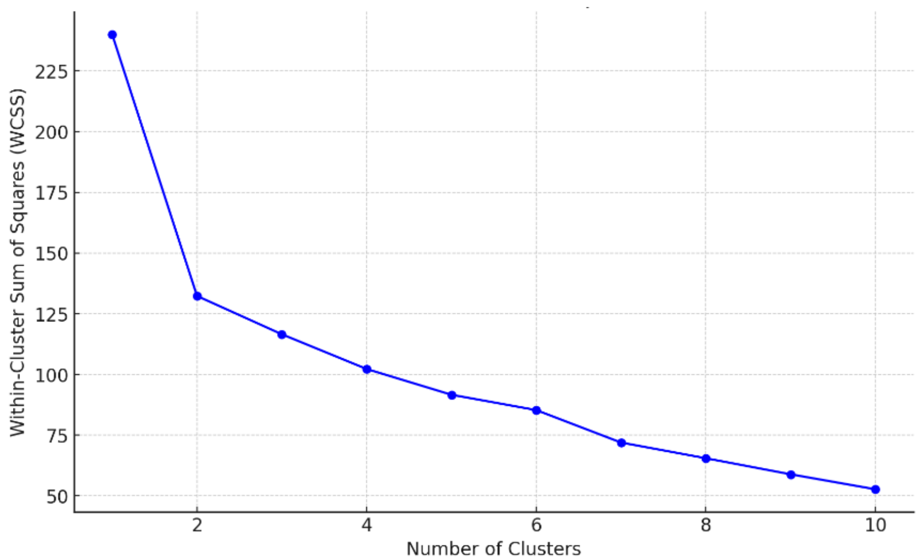


Figure 3.1: Elbow Method for Optimal k

As shown in figure 3.1, when the number of clusters is 2, the inertia decline rate [9] begins to slow down, which indicates that k=2 is an appropriate number of clusters.

The samples are thus divided into 2 clusters: cluster 0 and cluster 1. Cluster 0 consists of 13 samples and cluster 1 consists of 17 samples. Figure 3.2 and table 3.1 show the clustering result.

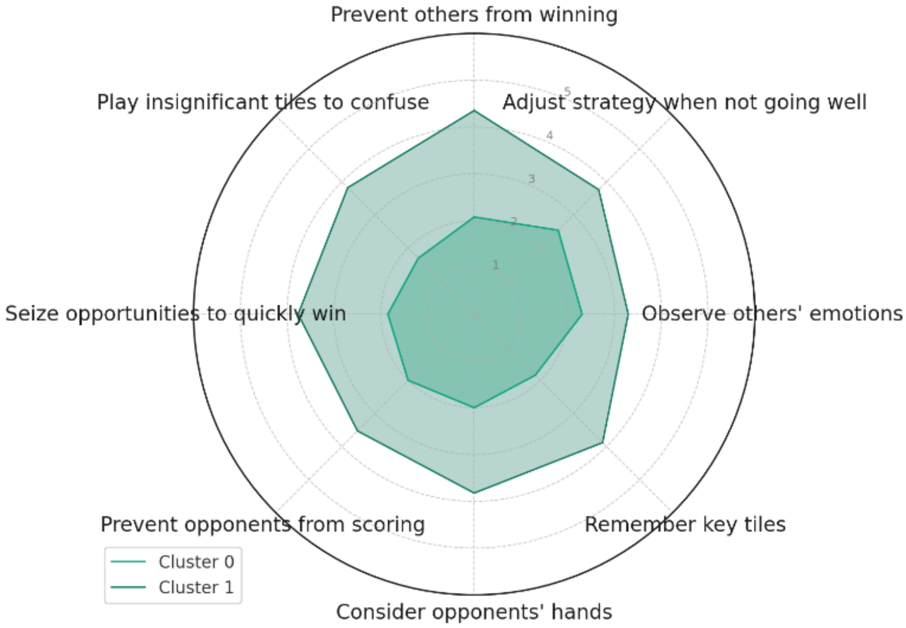


Figure 3.2: Survey Results for The 2 Clusters. The averages of each question are computed and compared in the figure.

Table 3.1: Survey Results for Different Clusters

	Cluster 0	Cluster 1
Question 1	2.08	4.35
Question 2	2.54	3.76
Question 3	2.31	3.29
Question 4	1.85	3.88
Question 5	2.00	3.82
Question 6	2.00	3.53
Question 7	1.85	3.76
Question 8	1.69	3.82

Based on the average values of the two sample clusters, the following characteristics can be observed:
Cluster 0:

- These samples scored relatively lower across all strategies.
- They prioritize completing their hand even if it means taking a risk to give their opponents useful tiles.
- They are inclined to take risks to achieve higher scores, often opting for bold offensive strategies.

Cluster 1:

- These samples scored higher across all strategies.

- They often analyze other players' discarded tiles to deduce potential risks and act accordingly.
- They are cautious and avoid providing opponents with any opportunity to complete their hand.

In summary, Cluster 0's strategy is more conservative, they are similar to the conservative players described in [11]. Cluster 1's strategy is more aggressive, they are similar to the aggressive players described in [11].

From the first part of the survey we know that there are two types of Sichuan mahjong expert. We also want to know exactly how these experts play differently in Sichuan mahjong. In the second part of the survey we designed a setting that is important in Sichuan mahjong: to form a uniform winning hand. The chance of forming a uniform winning hand is highest in Sichuan mahjong, compared to other types of mahjong in the world. This is because the rules of Sichuan mahjong require the players to keep at most two types of tiles in hand. This gives the players who want to form a uniform winning hand more chances to collect tiles (such as making pongs and kongs) from other players. Also, there is no wind tiles, or flower tiles in Sichuan mahjong, which increases the probability of drawing a uniform tile.

3.2. An Analysis of Player Tactics

Several questions were asked to the same group of Sichuan mahjong experts in the previous survey to investigate how they make the decision during play. To simplify the analysis, we focused on uniform hands. The questions are:

(1) You go for a uniform winning hand when you have ( ) uniform tiles.

(2) Suppose that your opponent is trying to make a dot uniform hand. You stop discarding dot tiles when your opponent has ( ) dot tiles.

(3) When counting tiles left in the wall, what is the cut-off point for you to believe your opponent is ready to win a uniform winning hand?

(4) When trying to form a uniform hand, how many rounds do you think it takes?

(5) If you have 3 words for a uniform hand and your opponent has 4 words for a uniform hand, what do you think the probability of winning before your opponent?

(6) How much (in percentage) do you think it is worth to go for a uniform hand?

The results for conservative players and aggressive players are shown in the tables below.

Table 3.2: Mean of Questions

	Conservative Players	Aggressive Players
Question 1	10.93	8.20
Question 2	9.67	12.20
Question 3	13.40	6.73
Question 4	0.34	0.68
Question 5	13.73	9.47
Question 6	0.25	0.52

Table 3.3: Confidence Interval of Questions

	Conservative Players	Aggressive Players
Question 1	[10.14, 11.73]	[7.53, 8.87]
Question 2	[8.81, 10.52]	[11.72, 12.68]
Question 3	[11.11, 15.69]	[5.51, 7.91]
Question 4	[0.24, 0.44]	[0.64, 0.72]
Question 5	[12.40, 15.06]	[8.63, 10.30]
Question 6	[0.2, 0.3]	[0.44, 0.59]

The results are also shown in figure 3.3.

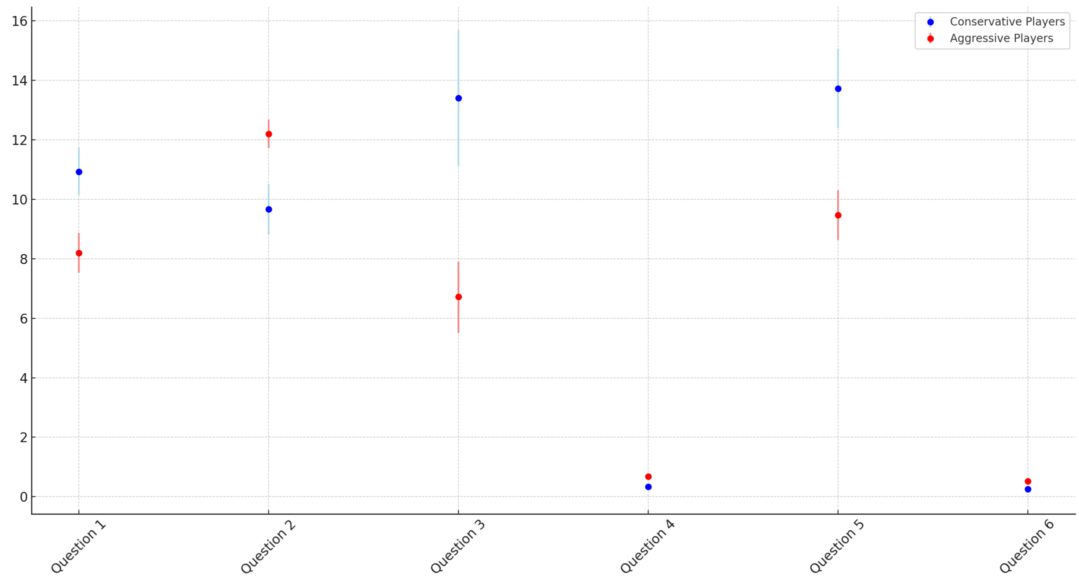


Figure 3.3: Question Results

We conducted a t-test to test if the sample means are different. The null hypothesis is that the means of the two types of player are equal. The alternative hypothesis is that the means are not equal.

The p-values for the 6 questions are:

Table 3.4: P-values

	P-value
Question 1	5.37×10^{-6}
Question 2	1.41×10^{-5}
Question 3	1.67×10^{-5}
Question 4	1.40×10^{-6}
Question 5	5.67×10^{-6}
Question 6	1.30×10^{-6}

The t-test results show that there are significant differences in the means between conservative players and aggressive players in all questions.

3.3. An Analysis of General Strategies

In the last part we designed two general strategies for the players. The first one is to focus on optimizing their own hand, and the second one is to play differently based on the state of their opponent's hand. We asked the players to choose their general strategy. 12 out 13 aggressive players focus on optimizing their own hand. 16 out of 17 conservative players play differently based on the state of their opponent's hand. The result shows that aggressive players and conservative players are also different in general strategies.

3.4. Conclusion

In conclusion, the survey clearly shows that there are two different types of Sichuan mahjong experts. They showed obvious differences in their choice of completing a uniform winning hand, and are divided into conservative and aggressive types. They also demonstrate differences in general strategies. The conservative player adopts different strategies based on the opponent's hand. The aggressive player chooses to optimize his(her) own hand as possible as (s)he can.

Since conservative players are researched in many literatures such as opponent modeling in mahjong AI, and there is still a gap in the research on the aggressive mahjong player, in the following part we focus on the aggressive Sichuan mahjong player. In the section 4.4 we simulate the aggressive player's strategy.

4

Measuring The Game State

The academic research on Sichuan mahjong, especially using quantitative methods to measure its game state, remains a gap. This chapter introduces different quantitative measures on the game state. They can be divided into two parts: measuring the state of the opponent's hand and measuring the state of your own hand. Section 4.1 introduces a way to measure the opponent's hand by estimating the number of a certain type of tiles. In section 4.2 and section 4.3, we focus on measuring the state of your own hand. We introduce the word number algorithm to measure the state of a hand, and based on this we calculate the number of tiles away from a winning hand. In section 4.4 we conduct a statistical simulation to calculate the winning probability and the expected number of rounds to win for different game states.

4.1. Estimating The Number of Uniform Tiles

In the survey of Chapter 3 we asked the players to guess the opponent's hand. In this section, we give a quantitative measure of the opponent's hand in a very restricted 2-player situation. The following example explains this measure.

Suppose that Alice and Bob are the only players allowed to complete their hand with the dot tiles. This means according to definition 11, Carol and Dick are not allowed to keep dot tiles.

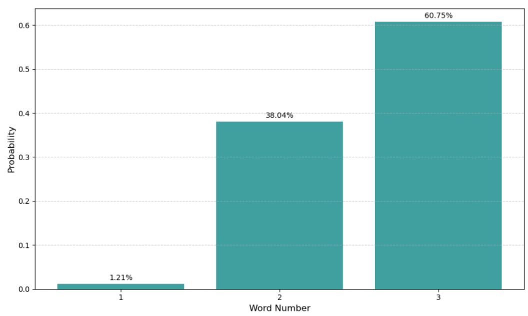
For player Alice, it is possible to guess the number of dot tiles in Bob's hand. Let Q denote the number of dot tiles in Bob's hand, N denote the number of dot tiles in Alice's hand, M denotes the number of dot tiles known on the table (i.e. played out by the players or a put-down pong, kong), X denotes the number of dot tiles in the wall. 36 is the total number of dot tiles. The random variable Q denotes the number of dot tiles in Bob's hand, and is calculated as:

$$Q = 36 - N - M - X \tag{4.1}$$

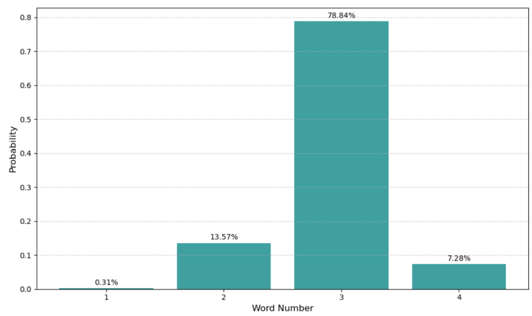
Alice can measure her opponent's state based on the Q. If $Q \geq 10$, Bob has a great chance to form a special winning hand which is called the "uniform winning hand".

Definition 15. *A uniform winning hand refers to a winning hand which only consists of one type of mahjong tile. The reward for the uniform winning hand is 4 times the normal winning hand.*

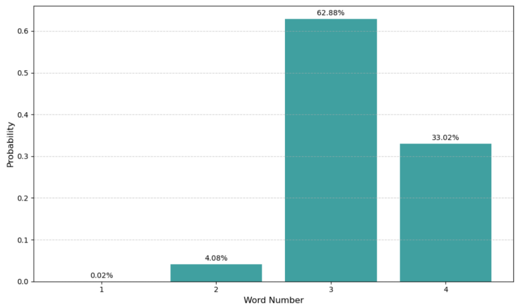
We launched a statistical simulation to explain the ”great chance”. We want to simulate the word numbers when Bob do have at least 10 dot tiles. By the word number measure, which is introduced in section 4.2, we know how Bob is close to a uniform winning hand. We randomly generate 1000 hands with 10, 11, 12, and 13 dot tiles respectively. For each 1000 hands, we count the tiles’ word number, and output the probability distribution. The word number distribution for $Q \geq 10$ are shown in the figures below.



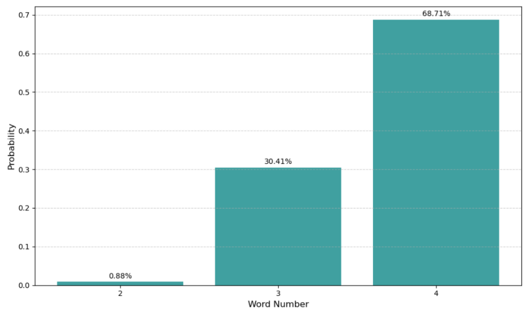
(a) Probability of Word Numbers in 10 Dot Tiles



(b) Probability of Word Numbers in 11 Dot Tiles



(c) Probability of Word Numbers in 12 Dot Tiles



(d) Probability of Word Numbers in 13 Dot Tiles

Figure 4.1: Combined Probability of Word Numbers in Dot Tiles

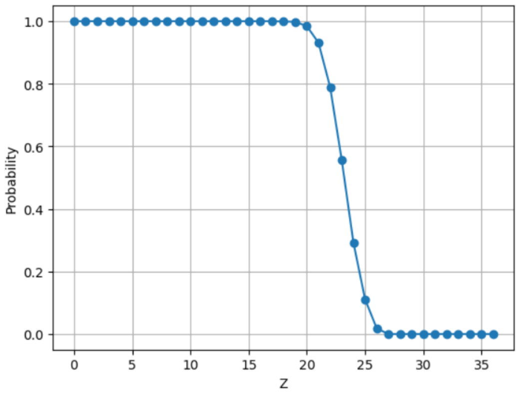
As shown in the figures, for a hand with at least 10 dot tiles, the chance that it has at least 3 words is very high, and the chance increases as the number of dot tiles increases. Bob has a great chance to win by forming 5 dot words.

For Alice, it is important to know if $Q \geq 10$. The values of N and M are clear to Alice. However, the number of dot tiles in the wall X is a random variable. Denote n as the number of tiles in the wall. Since the tiles are uniformly distributed in the wall, which means the probability of a random tile in the wall being a dot tile is $\frac{1}{3}$, X follows a binomial distribution $B(n, p)$, with $p = \frac{1}{3}$.

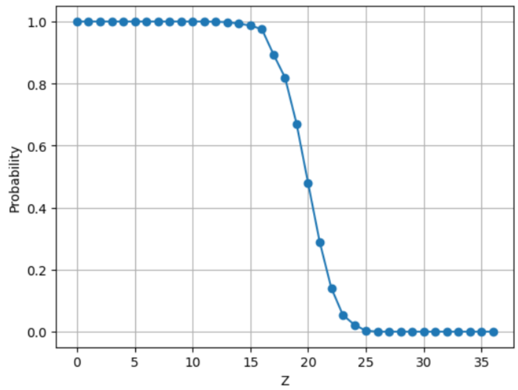
Based on the assumptions above, we simulated 10000 times the number of dot tiles for $n =$ 10, 20, 30, 40, and 50 respectively. Let $Z = M + N$, equation 4.1 can be rewritten as

$$Q = 36 - Z - X \tag{4.2}$$

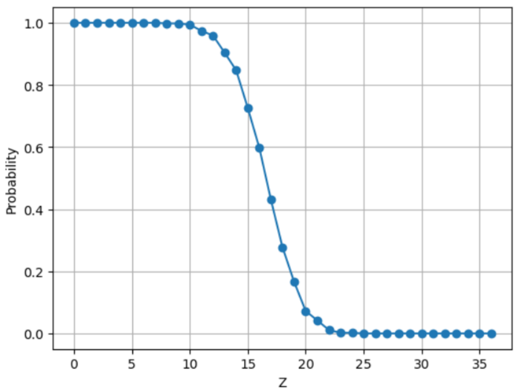
With the number of Z ranging from 0 to 36, we calculated the probability of Bob holding at least 10 dot tiles in his hand (the probability of $Q \geq 10$). The results are shown in the following plots.



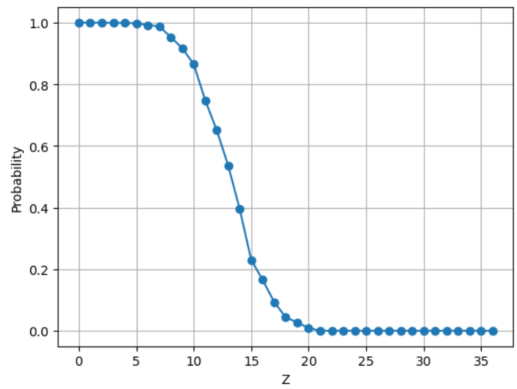
(a) Probability of Bob Holding at Least 10 Dot Tiles When $n = 10$



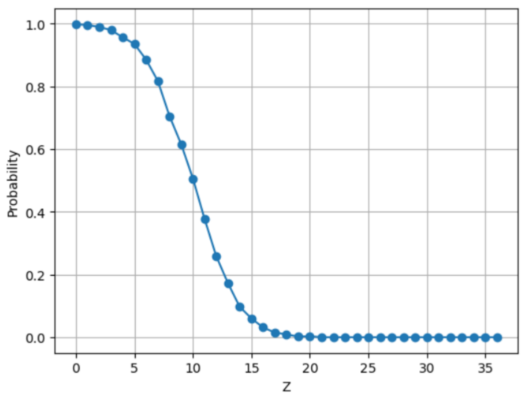
(b) Probability of Bob Holding at Least 10 Dot Tiles When $n = 20$



(c) Probability of Bob Holding at Least 10 Dot Tiles When $n = 30$



(d) Probability of Bob Holding at Least 10 Dot Tiles When $n = 40$



(e) Probability of Bob Holding at Least 10 Dot Tiles When $n = 50$

Figure 4.2: Combined Probability of Bob Holding at Least 10 Dot Tiles for Different n Values

4.2. Word Number Calculation

Mahjong experts always count how many words they have while playing. The term ”word” refers to the tiles in a hand which is either a pong, a chow, or a pair.

Definition 16. *A word is either a meld or a pair. A complete hand of mahjong contains and can only contain 5 words.*

According to the definition, a complete mahjong hand contains 4 meld words and a pair word.

In this report, we take the maximum number of words in our hand as a measure of the game state. The maximum number of words in the hand is also referred to as the word number.

Definition 17. *The word number is the maximum number of words in a hand of 14 mahjong tiles.*

4.2.1. Word Number Algorithm

Inspired by the quadtree algorithm in [6], we designed the word number algorithm for computing the word number. This algorithm counts the number of words in a given hand.

A word tree is constructed in the word number algorithm, which simulates all paths to represent different word division results. The algorithm continues in a depth-first search way. The original hand is recorded in the root node, and the new node is generated by subtracting a pong, chow, or pair (if it has not been subtracted before). All pongs, chows, and pairs that can be subtracted from the previous node will be recorded, and the remaining part goes to the next node. Then each adjacent node of the previous node is recursively visited. An example of one path in the word tree is shown in figure 4.3.

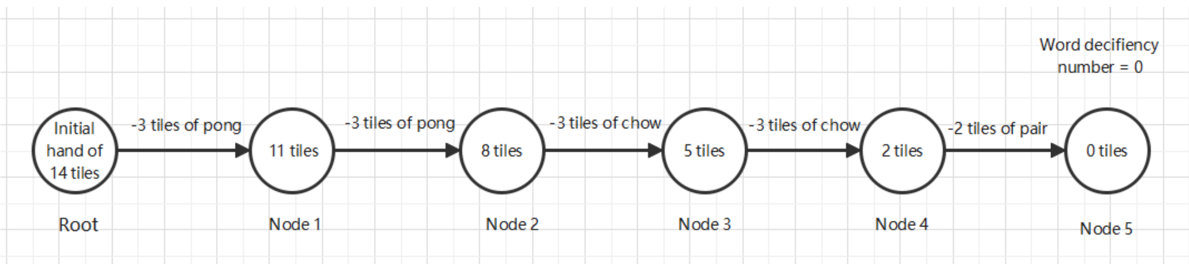


Figure 4.3: Word Tree Flow Chart

The complete procedures of the word number algorithm are:

1. The word tree is built. The root note represents the initial hand and the other nodes represent a remaining part of the hand. The edge (path) of the tree represents a possible move, which subtracts a word from the node.

2. For each node, if it has a depth less than 5 (i.e. it has less than 5 words), the algorithm checks all possible moves and executes them, creating child nodes.
3. The process is repeated until a node reaches a depth of 5 or there are no more moves to be executed.
4. The word number is then calculated as the maximum depth among all leaf nodes.

The code for the word number algorithm can be found in appendix A.1.

4.2.2. Efficiency Analysis

To demonstrate the efficiency of the word number algorithm, 1000 random hands of mahjong are generated. The maximum computing time and the average computing time for all hands are shown in table 4.1.

Table 4.1: Execution Time for Word Number Calculation

Max Time	Average Time
0.014 Seconds	0.0026 Seconds

Compared to the original quadtree deficiency number algorithm in [6], this algorithm saves a lot of computing time. This is because the word number algorithm only considers a simplified situation. As mentioned in [15], in the worst-case scenario, the quadtree deficiency number algorithm runs for 9.3 seconds. For the word number algorithm, the maximum time is 0.014 seconds, 664 times faster. Mahjong AI needs to respond within 8 seconds, so the word number algorithm would be a good application in mahjong AI.

4.2.3. Statistical Analysis

The word number measures the game state we are at. However, to compare with the deficiency number mentioned in [6], it is easier to use the definition ”word deficiency number”, which is the minumum number of words that a hand needs to complete.

Definition 18. *The word deficiency number is the difference between 5 and the word number.*

In the following part of this section we choose the word deficiency number as our research object. In order to conduct a statistical analysis on the word deficiency number, 1000 mahjong hands are randomly generated. The histogram of word deficiency numbers for 1000 randomly generated mahjong hands is shown in figure 4.4.

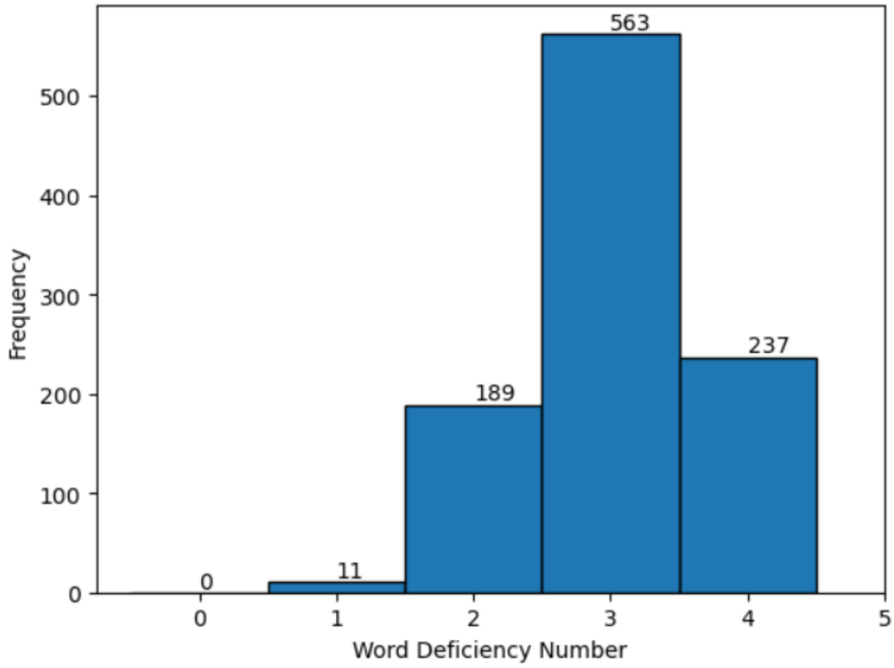


Figure 4.4: Histogram of Word Deficiency Numbers for 1000 Random Mahjong Hands

Figure 4.4 is an example of simulating 1000 mahjong hands. In order to get more statistical information, we conducted 100 simulations, with 1000 random mahjong hands generated in each simulation. The results are shown in figure 4.5.

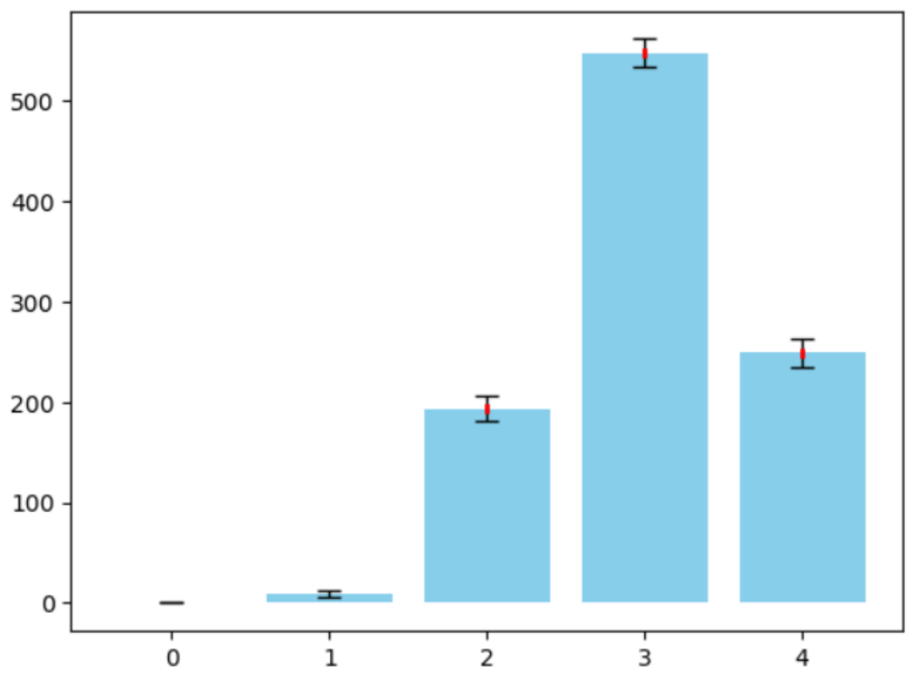


Figure 4.5: Word Deficiency Number Statistical Information of 100 Simulations

More detailed information can be found in table 4.2.

Table 4.2: Word Deficiency Number Statistical Information of 100 Simulations

	0	1	2	3	4
Mean	0.03	9.21	193.50	547.90	249.36
Variance	0.03	8.51	150.31	192.29	212.53
Confidence Interval	[0.00, 0.06]	[8.63, 9.79]	[191.06, 195.94]	[545.13, 550.67]	[246.45, 252.27]

As is shown in table 4.2, the word deficiency number ranges from 0 to 4, with 3 having the highest frequency. In theory, the word deficiency number should range from 0 to 5, but the chance of generating a hand with 5 word deficiency numbers is very low.

The interesting fact is more than half of the 1000 randomly generated hands have a word deficiency number of 3. The probability suggests that in a 4-player mahjong game, 2 players would start with a hand with 3 word deficiencies. A player who starts with less than 3 word deficiencies should be considered to have an advantage in the game.

The word deficiency number is also related to the deficiency number introduced in [6]. To compare the word deficiency number with the the deficiency number of a hand, we generated 1000 random hands and made a scatter plot. The horizontal axis represents the word deficiency number, and the vertical axis represents the deficiency number. The scatter plot is shown in figure 4.6.

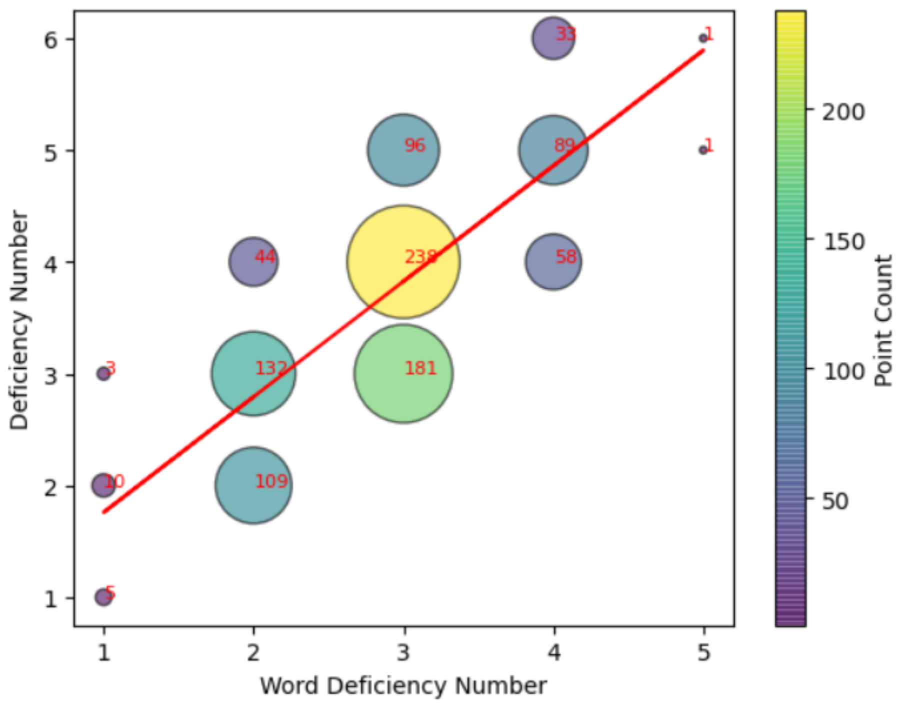


Figure 4.6: Scatter Plot of Word Deficiency Number and Deficiency Number

The plot shows that there exists a linear relationship between word deficiency number and deficiency number. Pearson's correlation coefficient is often used to measure the linear correlation between two sets of data. Pearson's correlation coefficient here is 0.73, with the p-value 0.0038. The low p-value (typically ≤ 0.05) indicates that the linear correlation is statistically significant. We also conducted a quadratic regression. However, the p-value of the quadratic term is 0.69 (> 0.05), meaning that the quadratic relationship is not significant.

The results suggests that there exists a moderate positive linear correlation (if the correlation coefficient r satisfies $0.4 \leq r \leq 0,8$ then it is moderate positive) between word deficiency number and deficiency number. This result is consistent with our common sense. If hand A has a larger word deficiency number than hand B, there is a great chance that hand A also has a larger deficiency number than hand B.

One interesting fact is that the word deficiency number is always less than or equal to the deficiency number, and the maximum difference between them is 2. This could be stated as the following theorem and conjecture.

Theorem 2. *Let X denote the difference between the deficiency number and the word deficiency number of a hand H:*

$$\textit{Deficiency Number} = \textit{Word Deficiency Number} + M.$$

Then $M \geq 0$.

We present a heuristic proof for theorem 2:

Proof. A word deficiency is caused by a certain situation, which is between the range of two extreme situations: TXT and $XXTXX$. Here T means a tile and X means not a tile in the neighborhood. TXT requires 1 change of tile to become a word. $XXTXX$ requires 1 or 2 changes of tile to become a word: if you want to form a pong or chow it takes 2 tile changes, if you want to form a pair it takes 1 tile change. Thus 1 word deficiency corresponds to at least 1 tile change (which is by definition the deficiency). So we have: deficiency number $\geq$ word deficiency number, i.e. $M \geq 0$.

As for the upper bound of M, we have the following conjecture:

Conjecture 1. *Let X denote the difference between the deficiency number and the word deficiency number of a hand H:*

$$\textit{Deficiency Number} = \textit{Word Deficiency Number} + M.$$

Then $M \leq 2$.

The word number algorithm also outputs the tiles that can not form a word. In this report we define those tiles as the ”non-word tiles”.

> **Definition 19.** *The non-word tiles are the tiles that do not belong to any word after running the word deficiency algorithm.*

The non-word tiles are important in the following research. In section 4.3 we calculate the tile away number based on the information of the non-word tiles.

4.3. Number of Tiles away from A Winning Hand

The deficiency number mentioned in [6] by definition is the minimum number of tile changes to form a winning hand. It can also be understood as the minimum number of tiles that is needed to form a winning hand.

However, for Sichuan mahjong, the deficiency number does not work anymore. The reason is that Sichuan mahjong only allows for at most two types of mahjong tiles in the final winning hand. The deficiency number does not take into account this specific rule. We define the tiles that the player wants to keep as ”legal tiles”.

> **Definition 20.** *The legal tiles in Sichuan mahjong are the tiles in the player's hand that belong to the (at most two) types that the player wants to keep in the final winning hand.*

We apply the word number algorithm to the legal tiles, and it outputs the non-word part of the legal tiles. We calculate how many legal tiles are needed to add to the non-word part, and thus form a winning hand. We define this number as the ”number of tiles away from the winning hand”, or ”tile away number” (see definition 21).

> **Definition 21.** *The number of tiles away from the winning hand (the tile away number) is the minimum number of legal tiles that is needed to complete the non-word part of legal tiles.*

4.3.1. Tile away Number Calculation

To calculate the tile away number, we designed a method and implemented it in the computer using Python. The general idea of this method is to make new words by completing the non-word tiles, and calculating the total number of tiles added in each new word.

For a pair in the non-word tiles, adding one more tile would make it a word of pong. This means for the pair, one more tile is needed to make it a word, and the tile away number for the pair is 1. Similarly, for a pchow in the non-word tiles, one more tile is needed to make it a chow, i.e. the tile away number for the pchow is 1.

If the sum of original words and new words is still less than 5, the single tiles in the remaining non-word tiles are taken into account. If there is already a pair in the words, the other words can only be a pong or a chow. Making a word requires adding two tiles. If there is no pair in the words, then a new pair should be firstly made, which only requires adding one tile. Then for other new pongs or chows to be made, two tiles are still needed for each.

The procedures for calculating the tile away number are summarized as follows:

1. Apply the word number algorithm to the legal tiles. Output the non-word tiles and the word deficiency number. Set the initial tile away number as 0.
2. For the non-word tiles, if there are pairs such as B1B1, for each pair, the word deficiency number -1 and the tile away number $+1$. The pairs are removed from the non-word tiles. This step continues until all pairs are exhausted.
3. For the rest non-word tiles, if there are pchows such as B2B3 or B2B4, for each pchow, the word deficiency number -1 and the tile away number $+1$. The pchows are removed from the non-word tiles. This step continues until all pchows are exhausted.
4. For the rest single tiles, when no pair word needs to complete, for each single tile, the tile away number $+2$ and the word deficiency number -1. When a pair word needs to complete, we complete the pair first by making the tile away number $+1$ and the word deficiency number -1, and for each of the remaining tiles, the tile away number $+2$ and the word deficiency number -1.
5. When the word deficiency number becomes 0, the calculation terminates and we output the tile away number.

4.3.2. Statistical Analysis

In order to conduct statistical analysis on the tile away number, 1000 mahjong hands are randomly generated. The histogram of tile away numbers for 1000 random generated mahjong hands is shown in figure 4.4.

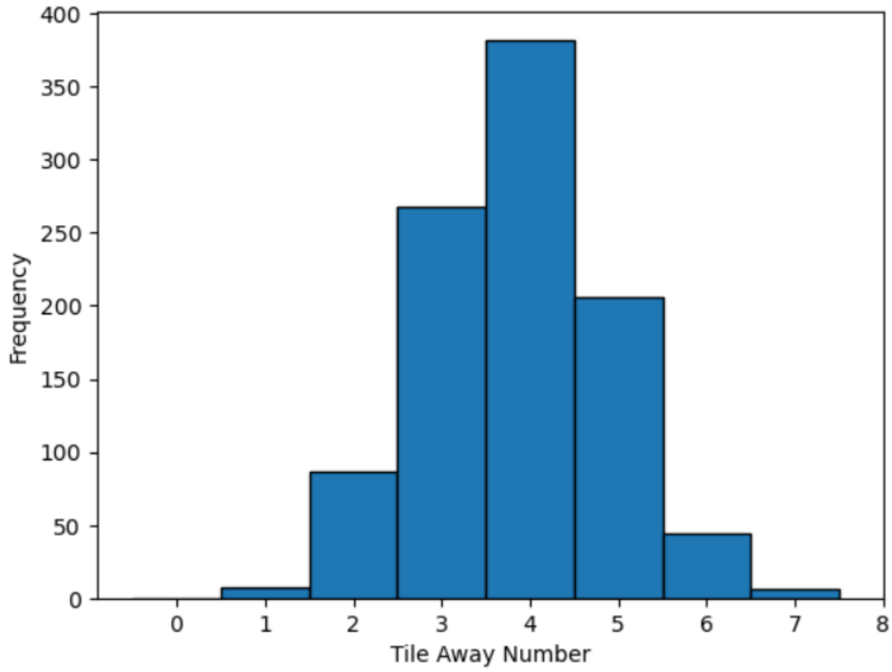


Figure 4.7: Histogram of Word Deficiency Numbers for 1000 Random Mahjong Hands

Figure 4.7 is an example of simulating 1000 mahjong hands. In order to get more statistical information, we conducted 100 simulations, with 1000 random mahjong hands generated in each simulation. The results are shown in figure 4.8.

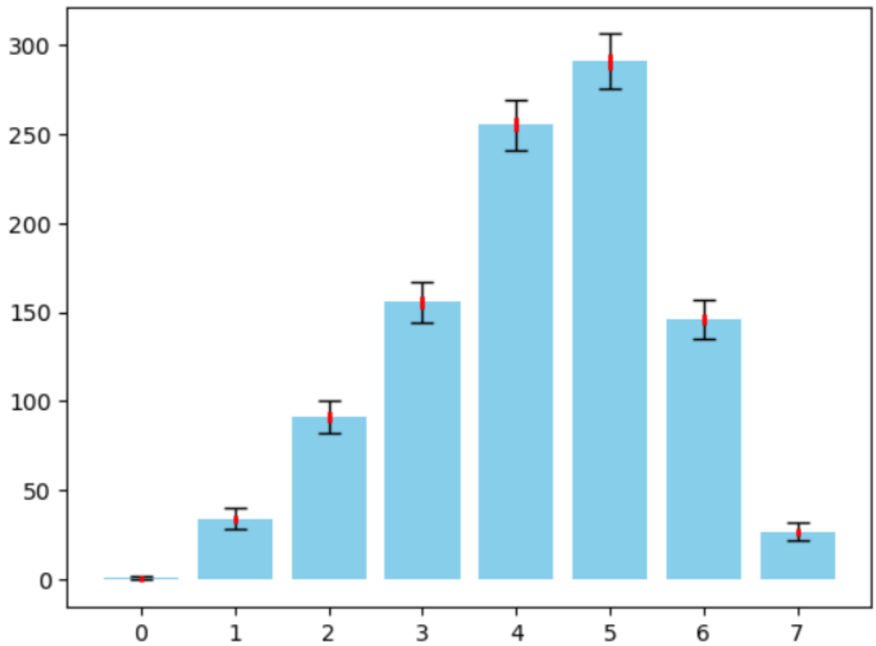


Figure 4.8: Tile Away Number Statistical Information of 100 Simulations

More detailed information can be found in table 4.3.

Table 4.3: Tile Away Number Statistical Information of 100 Simulations

	0	1	2	3	4	5	6	7
Mean	0.62	33.84	91.24	155.58	255.17	290.82	146.03	26.69
Variance	0.62	35.41	87.70	132.96	194.18	235.39	113.93	24.83
Confidence Interval	[0.46, 0.78]	[32.65, 35.03]	[89.37, 93.11]	[153.28, 157.88]	[252.39, 257.95]	[287.76, 293.88]	[143.90, 148.16]	[25.70, 27.68]

The tile away number can be compared to the word deficiency number. To do so, we generated 1000 random hands and made a scatter plot. The horizontal axis represents the word deficiency number, and the vertical axis represents the tile away number. The scatter plot is shown in figure 4.9.

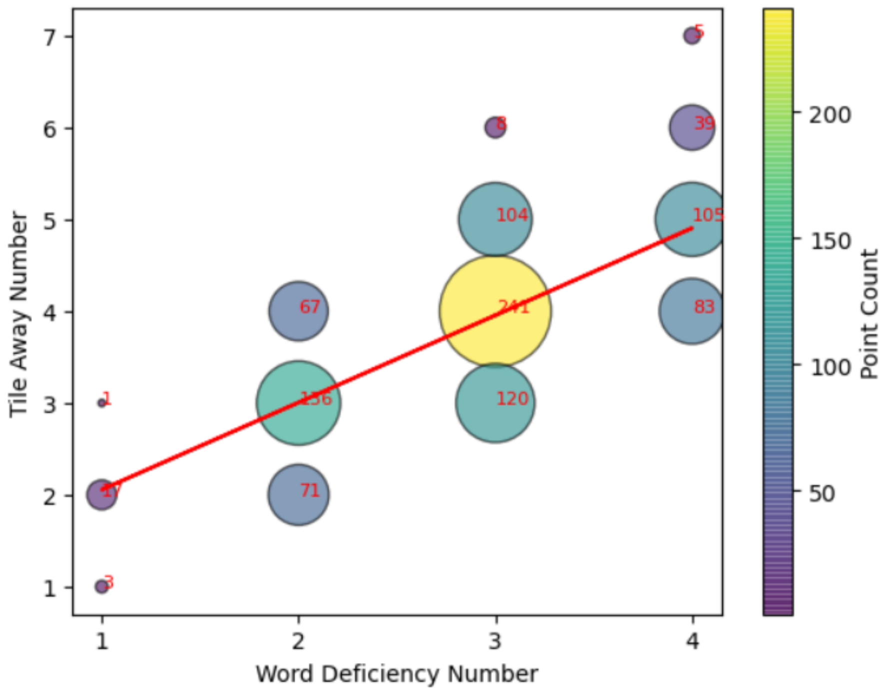


Figure 4.9: Scatter Plot of Tile Away Number and Deficiency Number

The plot suggests that there exists a linear relationship between the word deficiency number and the tile away number. Pearson's correlation coefficient is often used to measure the linear correlation between two sets of data. The Pearson's correlation coefficient here is 0.703, with the p-value 0.00413 (≤ 0.05), which indicates that there exists a moderate positive linear correlation between the word deficiency number and the tile away number.

4.4. Simulation of Sichuan Mahjong Hands

The unique rule of Sichuan mahjong states that the game should continue until the first three players win the game. This rule results in a situation in which there are only two players left in the game. The research for the two-player mahjong remains a gap. In this section we launched the simulation based on the two-player mahjong. We consider two types of two-player mahjong: uniform hands and general hands. We make the assumption that the mahjong players in our simulation are aggressive. We use the word number as the game state measure.

4.4.1. Uniform Hands Simulation

We give the definition of a uniform hand:

Definition 22. *A hand that consists of only one type of tile is called a uniform hand.*

In the uniform two-player mahjong, the players both consider forming a uniform winning hand. We simulated the distribution of word numbers for 10000 random dot uniform hands of 14 tiles. The result is:

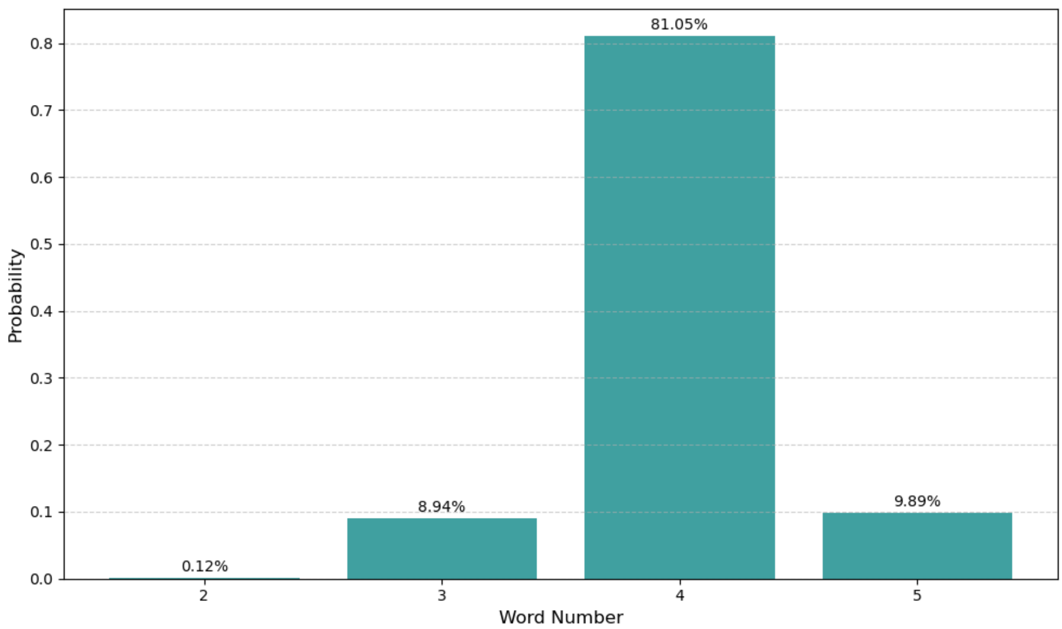


Figure 4.10: Probability of Word Numbers in 14 Dot Tiles

As is shown in figure 4.10, a 14-tile uniform hand contains either 2, 3, 4 or 5 words.

We designed a simulation in which we separately generated 1000 random uniform dot hands containing 2 words, 3 words, and 4 words each. For each hand, we randomly removed one dot tile from the non-word part, and added one. The probability of adding a dot tile is proportional to its remaining number, which equals to 4 minus its number contained in the hand. We calculated the probability that the word number increases as a result of the tile change. We call this probability the transition probability.

We launched 10 simulations, and then calculated the mean and variance. The result of the success rate is show in the table 4.4.

Table 4.4: Success Rate of Word Number Transition

	Transition from 2 to 3	Transition from 3 to 4	Transition from 4 to 5
Average Success Rate	0.77	0.62	0.13
95% Confidence Interval	[0.73, 0.81]	[0.59, 0.65]	[0.12, 0.14]

The game could also be represented in the form of figure 4.11. In the figure, the probabilities of staying in the same state and transitioning to the next state are shown. The number in each state means the word number.

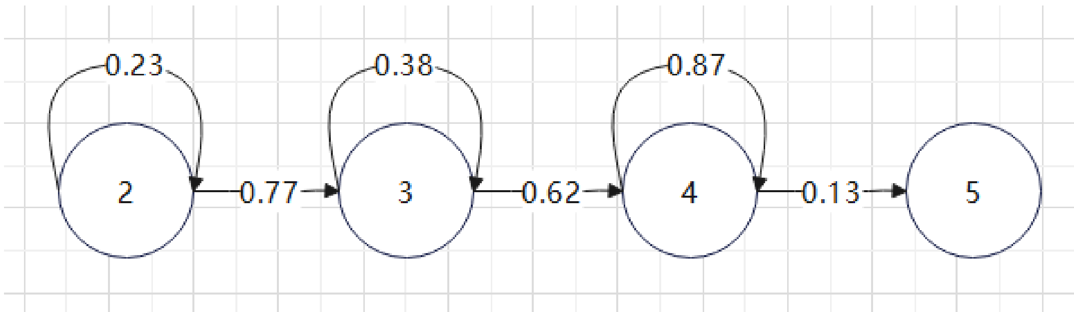


Figure 4.11: Success Rate

Based on table 4.4 and figure 4.11, we can also calculate the probability for a player to win in a two-player game.

Supposed that player Alice and Bob are the only two players that need the dot tiles to win. Consider situation 1: player Alice is in state 3 and player Bob is in state 4. The probability that Alice wins (Alice reaches state 5) can be calculated as follows:

Let P_1 denote the probability that Alice wins. With a probability of $0.38 \times 0.87 = 0.3306$ Alice and Bob stay at their original state. With a probability of $0.62 \times 0.87 = 0.5394$ Alice moves from state 3 to state 4 while Bob remains at state 4. Thus we have the following equation for P_1:

$$P_1 = 0.38 \times 0.87 \times P_1 + 0.62 \times 0.87 \times \frac{1}{2}$$

Solving the above equation we have $P_1 = 0.403$.

Consider situation 2: Alice is in state 2 and Bob is in state 4. Let P_2 denote the probability that Alice wins in this situation. With a probability of $0.23 \times 0.87 = 0.2001$ Alice and Bob stay at their original state. With a probability of $0.77 \times 0.87 = 0.6699$ Alice moves from state 2 to state 3 while Bob remains at state 4. Thus we have the following equation for P_2:

$$P_2 = 0.23 \times 0.87 \times P_2 + 0.77 \times 0.87 \times P_1$$

Solving the above equation we have $P_1 = 0.337$.

Consider situation 3: Alice is in state 2 and Bob is in state 3. Let P_3 denote the probability that Alice wins in this situation. With a probability of $0.23 \times 0.38 = 0.0874$ Alice and Bob stay at their original state. With a probability of $0.77 \times 0.38 = 0.2926$ Alice moves from state 2 to state 3 while Bob remains at state 3. With a probability of $0.77 \times 0.62 = 0.4774$ Alice moves from state 2 to state 3 while Bob moves from state 3 to state 4. With a probability of $0.23 \times 0.62 = 0.1436$ Alice stays at state 2 while Bob moves from state 3 to state 4. Thus we have the following equation for P_3:

$$P_3 = 0.23 \times 0.38 \times P_3 + 0.77 \times 0.38 \times \frac{1}{2} + 0.77 \times 0.62 \times P_1 + 0.23 \times 0.62 \times P_2$$

Solving the above equation we have $P_3 = 0.42$.

The winning probability can be summarized in the table 4.5.

Table 4.5: Winning Probability of Alice

State of Bob / State of Alice	2	3	4
2	0.5	0.420	0.337
3	0.580	0.5	0.403
4	0.663	0.597	0.5

During the simulations, we also calculated the average number of rounds it takes between states. The results are shown in table 4.6 and table 4.7.

Table 4.6: Mean of Average Number of Rounds Takes between States

To / From	3	4	5
2	1.20	2.89	10.54
3	0	1.63	9.37
4		0	7.03

Table 4.7: Confidence Interval of Average Number of Rounds Takes between States

To / From	3	4	5
2	[1.14, 1.26]	[2.69, 3.08]	[9.26, 11.83]
3	0	[1.58, 1.68]	[9.12, 9.62]
4		0	[6.72, 7.34]

4.4.2. General Hands Simulation

The rules of Sichuan mahjong require that a player can only keep at most 2 types of tile in hand, which is called the general hand. In the general two-player mahjong, the players both consider forming a general winning hand.

A general hand has at least 1 word. This can be easily proved: If a hand with 14 tiles has no word, this means at least 7 tiles in of the same type cannot make a word. In this case, the 7 tiles are numbered differently (otherwise there would be a pair). The problem can be also understood as choosing 7 numbers among 1 to 9, and there would surely be 3 consecutive number. Thus at least 3 of the 7 tiles are numbered consecutively, which makes the word.

The simulation is designed as follows. We generated 1000 general hands with the word number equals to 1, 2, 3, and 4 respectively. The player in the simulation is aggressive, who only considers optimizing its own hand by changing tiles from the non-word part. Thus for each hand, we randomly removed one tile from the non-word part, and added one. The probability of adding a dot tile is proportional to its remaining number, which equals to 4 minus its number contained in the hand. We calculated the probability that the word number increases as a result of the tile change. We call this probability the transition probability.

We launched 10 simulations. The result of the success rate and the expected number of rounds are shown in the tables below.

Table 4.8: Success Rate of Word Number Transition

	Transition from 1 to 2	Transition from 2 to 3	Transition from 3 to 4	Transition from 4 to 5
Average Success Rate	0.45	0.33	0.19	0.023
95% Confidence Interval	[0.43, 0.47]	[0.31, 0.35]	[0.18, 0.20]	[0.017, 0.028]

Table 4.9: Mean of Average Number of Rounds Takes to The Next State of Number of Words

From \ To	2	3	4
1	2.23	5.35	11.52
2	0	2.97	9.24
3		0	5.64

Table 4.10: Confidence Interval of Average Number of Rounds Takes to The Next State of Number of Words

From \ To	2	3	4
1	[2.16, 2.30]	[5.19, 5.50]	[11.15, 11.90]
2	0	[2.84, 3.10]	[8.89, 9.60]
3		0	[5.40, 5.88]

The simulation results indicate that the transition from 4 to 5 is extremely hard. It takes a lot of effort and (almost) impossible to simulate the number of rounds.

In reality, the transition is much easier. The players are able to deal with the non-word parts better. Instead of discarding a random tile in the non-word part, they can choose the best tile among the non-word tiles to discard, based on their experience.

For example, if the non-word part is $D3D4D9$, the experienced player would choose $D9$ to discard. This is easy to understand by intuition: if $D9$ is discarded, the remaining $D3D4$ can be made into a word by either adding $D2$ or $D5$. If $D3$ is discarded, no tile can make $D4D9$ into a word. Also according to [6], $D9$ has a higher discarding value. So it is reasonable to discard $D9$. $D9$ is called the best tile to discard.

Instead of conducting another simulation, we give a theoretical approach to calculate the rounds after the experienced player discards the best tile.

Theoretical Expected Number of Rounds

We first designed the "tiles needed to complete a winning hand" algorithm, which outputs the tiles needed to complete the winning hand after the experienced player discards the best tile when (s)he reaches the state of 4 words.

The code for the algorithm can be found in appendix A.2. Given a 13-tile phand which contains 4 words, this algorithm outputs all possible tiles that can complete the phand. After implementing the algorithm, the number of tiles we need is clear.

Suppose that there are n tiles left in the wall. The number of tiles needed and still available is k. The expected number of rounds to get the first needed tile out of the k tiles is:

$$E[N_1] = 1 \times P_1 + 2 \times P_2 + 3 \times P_3 + \dots$$

where P_i is the probability of getting the first needed tile in the ith round.

$E[N_1]$ could also be written in this way:

$$E[N_1] = \frac{1\binom{n-1}{k-1} + 2\binom{n-2}{k-1} + 3\binom{n-3}{k-1} + \cdots + (n-(k-1))\binom{n-(n-(k-1))}{k-1}}{\binom{n}{k}} \tag{4.3}$$

$\binom{n}{k}$ represents the total possible positions of k good tiles in n tiles. It can also be understood as there are n tiles in a row, and you choose k of them to be good. So in total there are $\binom{n}{k}$ ways to choose them.

Following the same logic, $\binom{n-i}{k-1}$ means that you choose the first $i-1$ tiles to be bad and the ith tile to be good. For the rest $n-i$ tiles where $k-1$ tiles are good, there are $\binom{n-i}{k-1}$ ways to arrange them.

Equation 4.3 can also be written as:

$$E[N_1] = \frac{\sum_{i=1}^{n-(k-1)} i \times \binom{n-i}{k-1}}{\binom{n}{k}} \tag{4.4}$$

Using symbolic computation, we get the expression for $\sum_{i=1}^{n-(k-1)} i \times \binom{n-i}{k-1}$ is:

$$\binom{n+1}{n-k} \tag{4.5}$$

So the final expression for $E[N_1]$ is

$$E[N_1] = \frac{\binom{n+1}{n-k}}{\binom{n}{k}} = \frac{n+1}{k+1} \tag{4.6}$$

However, the first tile from the k tiles may not be the best tile to complete the winning hand. For example, we have a phand which consists of $D1D1D1D2D3$ and other 3 words. In this situation both $D1$ and $D4$ can make $D1D1D1D2D3$ into 2 words and thus complete the winning hand. According to the rules of Sichuan mahjong, the winning hand with $D1$ has twice the reward as the winning hand with D4. This is because there are 4 $D1$ tiles in the winning hand, which doubles the reward.

So the experienced players may not always choose the first good tile they get to complete the winning hand. Sometimes they will wait until they get a few good tiles. The problem can be generalized as: for n tiles in the wall, k of them are good to complete a winning hand. What is the expected number of rounds for a player to get the first t good tiles?

We first discuss the situation when t equals 2. Now we want to know the number of rounds to get the first 2 good tiles. The calculation can be understood in the following way:

The expected number of rounds to get the first 2 needed tiles out of the k tiles is:

$$E[N_2] = 2 \times P_2 + 3 \times P_3 + 4 \times P_4 + \ldots$$

Consider a row of n tiles, k of them are good. There are $\binom{n}{k}$ ways to choose k tiles to be good. If we want to get the first 2 good tiles exactly in the ith round ($i \geq 2$), we can choose the ith tile to be good, and for the first $i-1$ tiles we choose one of them to be good. Thus the total number way of choosing is $\binom{i-1}{1} \times \binom{n-i}{k-2}$. Thus $P_i = \frac{\binom{i-1}{1} \times \binom{n-i}{k-2}}{\binom{n}{k}}$.

The expected number of rounds to get the first 2 needed tiles can be calculated as:

$$E[N_2] = \frac{\sum_{i=2}^{n-(k-2)} i \times (i-1) \times \binom{n-i}{k-2}}{\binom{n}{k}}$$

Using symbolic computation, we get the expression for $\sum_{i=2}^{n-(k-2)} i \times (i-1) \times \binom{n-i}{k-2}$ is:

$$2 \times \binom{n+1}{n-k}$$

So the final expression for $E[N_2]$ is:

$$E[N_2] = 2 \times \frac{\binom{n+1}{n-k}}{\binom{n}{k}} = \frac{2(n+1)}{k+1} \tag{4.7}$$

Following the same logic, when t=3, consider a row of n tiles, k of them are good. There are $\binom{n}{k}$ ways to choose k tiles to be good. If we want to get the first 3 good tiles exactly in the ith round ($i \geq 3$), we can choose the ith tile to be good, and for the first $i-1$ tiles we choose $3-1=2$ of them to be good. Thus the total number way of choosing is $\binom{i-1}{2} \times \binom{n-i}{k-3}$. Thus $P_i = \frac{\binom{i-1}{2} \times \binom{n-i}{k-3}}{\binom{n}{k}}$.
The expected number of rounds to get the first 3 needed tiles can be calculated as:

$$E[N_3] = \frac{\sum_{i=3}^{n-(k-3)} i \times \binom{i-1}{2} \times \binom{n-i}{k-3}}{\binom{n}{k}}$$

Using symbolic computation, we get the expression for $\sum_{i=3}^{n-(k-3)} i \times \binom{i-1}{2} \times \binom{n-i}{k-3}$ is:

$$3 \times \binom{n+1}{n-k}$$

So the final expression for $E[N_3]$ is:

$$E[N_3] = 3 \times \frac{\binom{n+1}{n-k}}{\binom{n}{k}} = \frac{3(n+1)}{k+1} \tag{4.8}$$

To be more general, the expected number of rounds to get the first t ($t \leq k$) good tiles is:

$$E[N_t] = t \times \frac{\binom{n+1}{n-k}}{\binom{n}{k}} = \frac{t(n+1)}{k+1} \tag{4.9}$$

We can prove this by induction: Suppose that we want to get the first a good tiles. From equation (4.9) we get the expected number of rounds is $\frac{a(n+1)}{k+1}$, which is denoted as $F(a, n, k)$. Another way to draw the first a good tiles is to firstly draw one tile and then the rest tiles. Since there are k good tiles, the probability for the first tile to be good is $\frac{k}{n}$. After the first draw the expected number of rounds is $F(a-1, n-1, k-1)$. The probability for the first tile to be not good is $\frac{n-k}{n}$. After the first draw the expected number of rounds is $F(a, n-1, k)$. So the expected number of rounds can also be calculated as:

$$\frac{k}{n}(1 + F(a-1, n-1, k-1)) + \frac{n-k}{n}(1 + F(a, n-1, k))$$

Replace $F(a-1, n-1, k-1)$ with $\frac{(a-1)n}{k}$ and $F(a, n-1, k)$ with $\frac{an}{k+1}$ we get:

$$\frac{k}{n}(1 + \frac{(a-1)n}{k}) + \frac{n-k}{n}(1 + \frac{an}{k+1}) = \frac{a(n+1)}{k+1}$$

which is the same as what we calculated before. Thus equation (4.9) is proved.

Winning Probability

The game could also be represented in figure 4.12. In the figure, the probabilities of staying in the same state and transitioning to the next state are shown. The number in each state means the word number.

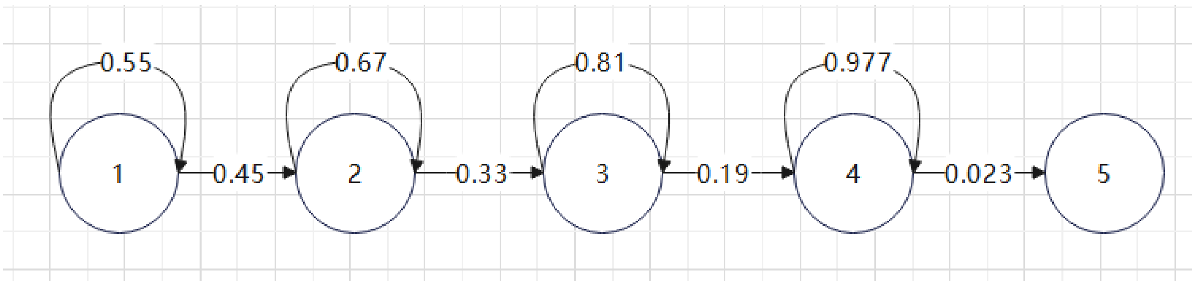


Figure 4.12: Success Rate

Suppose that player Alice and Bob are the players that are left. Consider situation 1: player Alice is in state 3 and player Bob is in state 4. The probability that Alice wins (Alice reaches state 5) can be calculated as follows: Let P_1 denote the probability that Alice wins. With a probability of $0.81 \times 0.977 = 0.79137$ Alice and Bob stay in their original state. With a probability of $0.19 \times 0.977 = 0.18563$ Alice moves from state 3 to state 4 while Bob remains at state 4. Thus we have the following equation for P_1:

$$P_1 = 0.81 \times 0.977 \times P_1 + 0.19 \times 0.977 \times \frac{1}{2}$$

Solving the above equation we have $P_1 = 0.44$.

Consider situation 2: player Alice is in state 2 and player Bob is in state 4. The probability that Alice wins can be calculated as follows: Let P_2 denote the probability that Alice wins. With a probability of $0.67 \times 0.977 = 0.6546$ Alice and Bob stay in their original state. With a probability of $0.33 \times 0.977 = 0.32241$ Alice moves from state 2 to state 3 while Bob remains at state 4. Thus we have the following equation for P_2:

$$P_2 = 0.67 \times 0.977 \times P_2 + 0.33 \times 0.977 \times P_1$$

Solving the above equation we have $P_2 = 0.41$.

Consider situation 3: player Alice is in state 1 and player Bob is in state 4. The probability that Alice wins can be calculated as follows: Let P_3 denote the probability that Alice wins. With a probability of $0.55 \times 0.977 = 0.53735$ Alice and Bob stay in their original state. With a probability of $0.45 \times 0.977 = 0.43965$ Alice moves from state 1 to state 2 while Bob remains at state 4. Thus we have the following equation for P_3:

$$P_3 = 0.55 \times 0.977 \times P_3 + 0.45 \times 0.977 \times P_2$$

Solving the above equation we have $P_3 = 0.39$.

Consider situation 4: player Alice is in state 2 and player Bob is in state 3. The probability that Alice wins can be calculated as follows: Let P_4 denote the probability that Alice wins. With a probability of $0.67 \times 0.81 = 0.5427$ Alice and Bob stay in their original state. With a probability of $0.33 \times 0.81 = 0.2673$ Alice moves from state 2 to state 3 while Bob remains at state 3. With probability of $0.67 \times 0.19 = 0.1273$ Alice stays at her original state while Bob moves from state 3 to state 4. With probability of $0.33 \times 0.19 = 0.0627$ Alice moves from state 2 to state 3 while Bob moves from state 3 to state 4. Thus we have the following equation for P_4:

$$P_4 = 0.67 \times 0.81 \times P_4 + 0.33 \times 0.81 \times \frac{1}{2} + 0.33 \times 0,19 \times P_1 + 0.67 \times 0.19 \times P_2$$

Solving the above equation we have $P_4 = 0.47$.

Consider situation 5: player Alice is in state 1 and player Bob is in state 3. The probability that Alice wins can be calculated as follows: Let P_5 denote the probability that Alice wins. With a probability of $0.55 \times 0.81 = 0.4455$ Alice and Bob stay in their original state. With a probability of $0.45 \times 0.81 = 0.3645$ Alice moves from state 1 to state 2 while Bob remains at state 3. With a probability of $0.45 \times 0.19 = 0.0855$ Alice moves from state 1 to state 2 while Bob moves from state 3 to state 4. With a probability of $0.55 \times 0.19 = 0.1045$ Alice remains at state 1 while Bob moves from state 3 to state 4. Thus we have the following equation for P_5:

$$P_5 = 0.55 \times 0.81 \times P_5 + 0.45 \times 0.81 \times P_4 + 0.45 \times 0.19 \times P_2 + 0.55 \times 0.19 \times P_3$$

Solving the above equation we have $P_5 = 0.45$.

Consider situation 6: player Alice is in state 1 and player Bob is in state 2. The probability that Alice wins can be calculated as follows: Let P_6 denote the probability that Alice wins. With a probability of $0.55 \times 0.67 = 0.3685$ Alice and Bob stay in their original state. With a probability of $0.45 \times 0.67 = 0.3015$ Alice moves from state 1 to state 2 while Bob remains at state 2. With a probability of $0.45 \times 0.33 = 0.1485$ Alice moves from state 1 to state 2 while Bob moves from state 2 to state 3. With a probability of $0.55 \times 0.33 = 0.1815$ Alice stays at state 1 while Bob moves from state 2 to state 3. Thus we have the following equation for P_6:

$$P_6 = 0.55 \times 0.67 \times P_6 + 0.45 \times 0.67 \times \frac{1}{2} + 0.45 \times 0.33 \times P_4 + 0.55 \times 0.33 \times P_5$$

Solving the above equation we have $P_6 = 0.48$.

The winning probabilities are summarized in the table 4.11.

Table 4.11: Winning Probability of Alice

State of Alice \ State of Bob	1	2	3	4
1	0.5	0.48	0.45	0.39
2	0.52	0.5	0.47	0.41
3	0.55	0.53	0.5	0.44
4	0.61	0.59	0.56	0.5

The winning probability shows that the chance for a disadvantageous player to win is still high. This is because the transition between different states becomes harder as the states get closer to winning. As are shown both in table 4.4 and table 4.8, the success rate decreases as the word number increases. The fact that both players are likely to win reveals the high playability of Sichuan mahjong.

4.5. Experimental Game Theory of Mahjong

To compare the simulation results in table 4.10 and table 4.11 with the real-world result, we asked four of the aggressive Sichuan mahjong players mentioned in chapter 3 to participate in an experiment. The four aggressive players are all students from TU Delft, with at least 5 years' professional experience of playing Sichuan mahjong on a competitive level. Figure 4.13 shows the beginning of a game in our experiment.

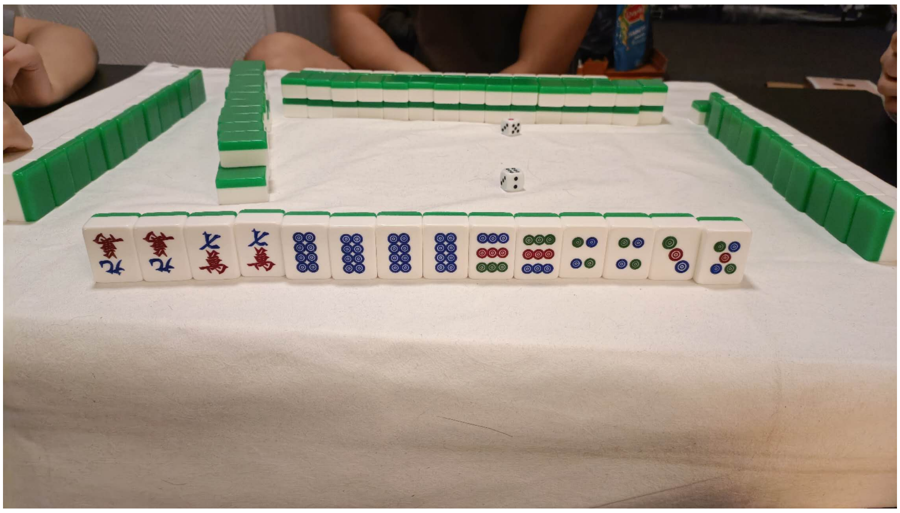


Figure 4.13: The Beginning of A Game in The Experiment

In the experiment we asked the four players to play 100 games, which took around 8 hours. During each game we started to record the game state when there were only two players left. We denoted the player who was in a disadvantageous state as Alice, and another player as Bob. If the two players were in the same state, we stopped the game and started the next game immediately.

The game states in which we start to record the game are shown in table 4.12.

Table 4.12: Game States When Start to Record

Starting State of Alice \ Starting State of Bob	2	3	4
1	14	19	8
2		14	22
3			23

We also recorded the number of games in which Alice won:

Table 4.13: Number of Games in Which Alice Won

Starting State of Alice \ Starting State of Bob	2	3	4
1	6	8	3
2		6	7
3			9

So the winning probability of Alice is:

Table 4.14: Winning Probability of Alice

Starting State of Alice \ Starting State of Bob	2	3	4
1	0.43	0.42	0.38
2		0.43	0.32
3			0.39

After conducting 1000 bootstraps, the confidence interval of Alice's winning probability is:

Table 4.15: Confidence Interval of Alice's Winning Probability

Starting State of Alice \ Starting State of Bob	2	3	4
1	[0.33, 0.53]	[0.32, 0.52]	[0.28, 0.47]
2		[0.33, 0.53]	[0.23, 0.41]
3			[0.30, 0.49]

The experimental result is systematically smaller than the simulation result in table 4.11, which indicates that our simulation is a little bit optimistic. During the experiment, the players were informed about the opponent's states, while in the simulation the players were not. Such transparency in state information offers an edge to the advantageous player.

Besides the winning probability, we calculated the average number of rounds in a game. The average number of rounds for transition from one state to the next is shown in table 4.16.

Table 4.16: Average Number of Rounds for Transition to The Next State

State Transition	from 1 to 2	from 2 to 3	from 3 to 4	from 4 to 5
Average Number of Rounds	1.91	2.85	4.74	6.82

Compared to the result in table 4.10, the experimental result is also systematically smaller. This is because the players drew from a smaller set of tiles during the actual gameplay.

For the winner, the average number of rounds to win, for different beginning states, is shown in table 4.17. The result is also systematically smaller than the simulation.

Table 4.17: Average Number of Rounds for Transition to State 5

Starting State	1	2	3	4
Average Number of Rounds	15.70	13.96	11.24	6.82

In conclusion, although systematically smaller, the experimental results are close to the simulation results and thus validate the simulation results.

4.6. Conclusion

In conclusion, this chapter investigates the player's game state by different measures. Based on the word number measure, we also calculated the winning probability and expected number of rounds from winning. The result shows the high playability of Sichuan mahjong. In the end, the result is validated by the experiment.

5

Conclusion

This report conducted an in-depth exploration into the game state measurement of a particular kind of mahjong: Sichuan mahjong. Sichuan mahjong is one of the most famous mahjongs all over the world. Sichuan mahjong has some special rules. The game continues until 3 players have completed the winning hand. A player can only keep at most 2 types of tiles in the winning hand. The first rule results in the unique game situation: two-player mahjong. In this situation only 2 players are left in the game and they have to compete with each other. The two-player mahjong research remains a gap, and this report focuses on measuring the game state within the two-player mahjong setting.

In chapter 3 we started a research on the Sichuan mahjong experts. We asked questions about their playing type. The results show that they can be divided into aggressive and conservative. The aggressive player only optimizes his(her) own hand while the conservative player considers different strategies according to the opponent's behavior.
Since researches in mahjong cover a lot of conservative players but few aggressive players, we launched simulations and analysis based on the aggressive player assumption in chapter 4.

In chapter 4 we designed quantitative methods to measure the game state. In section 4.1 we measured the opponent's state by estimating the number of uniform tiles in his(her) hand. In section 4.2 we used the word number as the game state measure and designed the word number algorithm. In section 4.3 we calculated the tile away number and compared it to the deficiency number. In section 4.4 we simulated the two-player mahjong. We calculated the winning probability and the number of rounds to win for different game states. We also conducted an experiment to validate our results.

The results in section 4.4 reveal that transitions between game states are highly dependent on the player's current state. The closer the state is to winning, the harder the transition is. Because of this, the winning probability for the player at a disadvantageous state is still high enough. This finding reveals the high playability of Sichuan mahjong.

There are still limitations in our study. We only focus on the two-player mahjong, and we assume all the players to be aggressive. More research could be done by considering the multi-player mahjong or assuming one of the two players is conservative.

References

[1] Yuan Cheng, Chi-Kwong Li, and Sharon H Li. “Mathematical aspect of the combinatorial game” Mahjong””. In: *arXiv preprint arXiv:1707.07345* (2017).

[2] IJCAI 2020 Mahjong AI Competition. *IJCAI 2020 Mahjong AI Competition*. `https://botzone.org.cn/static/gamecontest2020a.html`. Accessed: 2023-02-08.

[3] IJCAI 2022 Mahjong AI Competition. *IJCAI 2022 Mahjong AI Competition*. `https://botzone.org.cn/static/gamecontest2022a.html`. Accessed: 2023-02-08.

[4] Thierry Depaulis. “Embarrassing tiles: Mahjong and the taipings”. In: *Playing-Card* 35.3 (2007), pp. 148–153.

[5] Maggie Greene. “The game people played: Mahjong in modern Chinese society and culture”. In: *Cross-Currents* 17 (2015), pp. 1–25.

[6] Sanjiang Li and Xueqing Yan. “Let’s Play Mahjong!” In: *arXiv preprint arXiv:1903.03294* (2019).

[7] Christian E Lopez and Conrad S Tucker. “The effects of player type on performance: A gamification case study”. In: *Computers in Human Behavior* 91 (2019), pp. 333–345.

[8] Yunlong Lu, Wenxin Li, and Wenlong Li. “Official International Mahjong: A New Playground for AI Research”. In: *Algorithms* 16.5 (2023), p. 235.

[9] Dhendra Marutho, Sunarna Hendra Handaka, Ekaprana Wijaya, et al. “The determination of cluster number at k-mean using elbow method and purity evaluation on headline news”. In: *2018 international seminar on application for technology of information and communication*. IEEE. 2018, pp. 533–538.

[10] Naoki Mizukami and Yoshimasa Tsuruoka. “Building a computer Mahjong player based on Monte Carlo simulation and opponent models”. In: *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE. 2015, pp. 275–283.

[11] Hiroshi Sato et al. “An analysis of play style of advanced mahjong players toward the implementation of strong AI player”. In: *International Journal of Parallel, Emergent and Distributed Systems* 32.2 (2017), pp. 195–205.

[12] Mohsen Tavakol and Reg Dennick. “Making sense of Cronbach’s alpha”. In: *International journal of medical education* 2 (2011), p. 53.

[13] Gustavo F Tondello and Lennart E Nacke. “Player characteristics and video game preferences”. In: *Proceedings of the annual symposium on computer-human interaction in play*. 2019, pp. 365–378.

[14] Mingyan Wang et al. “A novel deep residual network-based incomplete information competition strategy for four-players Mahjong games”. In: *Multimedia Tools and Applications* 78 (2019), pp. 23443–23467.

[15] Xueqing Yan, Yongming Li, and Sanjiang Li. "A Fast Algorithm for Computing the Deficiency Number of a Mahjong Hand". In: *arXiv preprint arXiv:2108.06832* (2021).

[16] Yajun Zheng and Shuqin Li. "A review of Mahjong AI research". In: *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*. 2020, pp. 345–349.

[17] Yajun Zheng and Shuqin Li. "Mass Mahjong Decision System Based on Transfer Learning". In: *Proceedings of the 2022 6th International Conference on Innovation in Artificial Intelligence*. 2022, pp. 228–233.

A
Python Codes

A.1. Word Number Algorithm

```
class Node:
    def __init__(self, hand, depth=0, pair_removed=False):
        self.hand = hand
        self.children = []
        self.depth = depth
        self.pair_removed = pair_removed

def build_quadtree(node):
    hand = node.hand
    num_suits, num_ranks = hand.shape
    if node.depth >= 5:
        return
    for i in range(num_suits):
        for j in range(num_ranks):
            if hand[i][j] >= 3:
                modified_hand = hand.copy()
                modified_hand[i][j] -= 3
                child = Node(modified_hand, node.depth + 1, node.pair_removed)
                node.children.append(child)
                build_quadtree(child)
            elif j <= num_ranks - 3 and hand[i][j] >= 1 and hand[i][j+1] >= 1 and hand[i][j+2] >= 1:
                modified_hand = hand.copy()
                modified_hand[i][j] -= 1
                modified_hand[i][j+1] -= 1
                modified_hand[i][j+2] -= 1
                child = Node(modified_hand, node.depth + 1, node.pair_removed)
                node.children.append(child)
                build_quadtree(child)
            elif not node.pair_removed and hand[i][j] >= 2:
                modified_hand = hand.copy()
                modified_hand[i][j] -= 2
                child = Node(modified_hand, node.depth + 1, True)
                node.children.append(child)
                build_quadtree(child)
```

```

def calculate_deficiency_number(node, cache):
    key = str(node.hand)
    if key in cache:
        return cache[key]
    if node.depth == 5:
        result = 0
    else:
        result = min([calculate_deficiency_number(child, cache) for child in node.children]) if node.children else 5 - node.depth
    cache[key] = result
    return result

def compute_deficiency_number(hand):
    root = Node(hand)
    build_quadtree(root)
    return calculate_deficiency_number(root, {})
```

A.2. Tiles Needed to Complete A Winning Hand Algorithm

```
character = [int(i) for i in input('Please␣input␣character␣tiles␣(separate␣by␣commas,␣if␣you␣don␣not␣have␣please␣enter␣0):').split(',')]
bamboo = [int(i) + 10 for i in input('Please␣input␣bamboo␣tiles␣(separate␣by␣commas,␣if␣you␣don␣not␣have␣please␣enter␣0):').split(',')]
dot = [int(i) + 20 for i in input('Please␣input␣dot␣tiles␣(separate␣by␣commas,␣if␣you␣don␣not␣have␣please␣enter␣0):').split(',')]

mahjong_1 = wan + tiao + tong
mahjong = [i for i in mahjong_1 if i not in [0,10,20]]
mahjong.sort()

class Mahjong:
    def __init__(self,mahjong,mahjong_1):
        self.mahjong = mahjong
        self.mahjong_1 = mahjong_1

    def Is_valid(self):
        if len(self.mahjong) % 3 != 1:
            return 'How␣did␣you␣manage␣to␣have␣{}␣cards␣left␣in␣your␣hand?'.format(len(self.mahjong))
        elif all([(0 not in self.mahjong_1), (10 not in self.mahjong_1),(20 not in self.mahjong_1)]):
            return "Violation␣of␣Sichuan␣mahjong␣rules"
        else:
            for x in set(self.mahjong):
                if self.mahjong.count(x) > 4:
                    return 'You␣cannot␣have{}␣identical␣tiles'.format(self.mahjong.count(x))
            else:
                pass
        return 'Good!'

    def Judge(self,mahjong):
        double = [x for x in set(mahjong) if mahjong.count(x) >= 2]
        if len(double) == 0:
```

```
            return False
        if len(double) == 7:
            return True
        for i in double:
            mahjong_copy = mahjong.copy()
            mahjong_copy.remove(i)
            mahjong_copy.remove(i)
            for j in mahjong_copy:
                if j != -1 and mahjong_copy.count(j) >= 3:
                    mahjong_copy[mahjong_copy.index(j)] = -1
                    mahjong_copy[mahjong_copy.index(j)] = -1
                    mahjong_copy[mahjong_copy.index(j)] = -1
                elif ((j + 1) in mahjong_copy) and ((j + 2) in mahjong_copy):
                    mahjong_copy[mahjong_copy.index(j)] = -1
                    mahjong_copy[mahjong_copy.index(j + 1)] = -1
                    mahjong_copy[mahjong_copy.index(j + 2)] = -1
                else:
                    pass
            mahjong_copy = [i for i in mahjong_copy if i != -1]
            if mahjong_copy == []:
                return True
        return False

    def Find_solution(self):
        self.solutions = []
        for i in range(1,30):
            if i not in [0,10,20,30]:
                mahjong_copy = self.mahjong.copy()
                mahjong_copy.append(i)
                if self.Judge(mahjong_copy) == True:
                    self.solutions.append(i)

    def Translate(self,mahjong):
        translate = []
        for i in mahjong:
            if 0 < i < 10:
                translate.append('{}C'.format(i))
            elif 10 < i < 20:
                translate.append('{}B'.format(i - 10))
            else:
                translate.append('{}D'.format(i - 20))
        return translate

demo = Mahjong(mahjong, mahjong_1)
if demo.Is_valid() == 'Good!':
    demo.Find_solution()
    print('The tiles you have are:')
    print(demo.Translate(mahjong))
    print('='*80)
    if len(demo.solutions) > 0:
        print('The tiles you need to complete your winning hand is:')
        print(demo.Translate(demo.solutions))
    else:
        print('You do not have 4 words  yet')
else:
```

```
    print('The tiles you have:')
    print(demo.Translate(mahjong))
    print('='*80)
    print(demo.Is_valid())
```