

國立臺灣師範大學

資訊工程學系研究所

碩士論文

指導教授：林順喜 博士



利用強化特定牌型得分方法改良麻將程式
Enhancing the Scores of Specific Hand Types
to Improve the Mahjong Program

研究生：陳均泓 撰

中華民國 111 年 8 月

致謝

首先要感謝的是我的老師林順喜教授的指導，在這兩年間，不管是修課、參加比賽、擔任助教、研究等，都給了我許多建議與指引。因為老師不厭其煩的指導，我才能夠從中學習到許多經驗與知識，讓我在這兩年間能夠培養許多基礎能力，讓本來能力就較差的我，也能夠順利完成研究與撰寫這份碩士論文。

感謝實驗室的志宏學長在許多方面的建議與幫忙，在我有疑問時總是願意替我解答，以及實驗室的各種事項都能夠順利進行與完成。感謝彥吉學長在許多專業知識上的傳授與實作技術上的教學。感謝宗翰學長在研究與程式上願意花時間與我討論，讓我能夠針對先前的相關研究與程式能夠有快速上手與深入瞭解的機會。

感謝交大的唐士傑學長與 Cycgame 的平台，以及臺北大學陳志昌教授實驗室所開發的平台，讓我能夠有方便的實驗環境，以及清楚準確的實驗數據能夠分析實驗結果，還有能夠正確的修復程式遇到的 BUG。

感謝身邊陪伴我度過這兩年的朋友們，在忙碌、壓力、以及各種負面情緒到來時，給了我許多正能量，提供我繼續向前的動力。

最後當然要感謝我的家人，時常關心我的狀況，給予我鼓勵，成為我修讀碩士期間的支柱，讓我不需要擔心經濟狀況，順利完成我的碩士學業。

摘要

近年來隨著人工智慧在電腦對局領域蓬勃發展，電腦對局領域的程式強度已經有了明顯成長。麻將為一個多玩家、機率性且不完全資訊的遊戲，由於麻將遊戲隨機且有限資訊的特性，也增加了遊戲的複雜度及困難度。

本篇論文將會針對臺灣麻將遊戲的規則，並且參考先前的相關研究，針對麻將程式進行改良。本篇論文將延續「利用棄牌資訊強化策略改良麻將程式」論文，使用規則導向與進胡數計算為主要的程式架構，並且針對原程式中的缺點，提出相對應的演算法進行改良。

本論文將會依照進攻與防守兩個方面進行改良。進攻方面以原先的進胡數計算為主要架構達到快速胡牌的目標，並且進一步注重在台分的獲取，能夠獲取更多的分數。防守方面進一步降低程式放槍率，降低分數的損失。

實驗數據顯示，改良後的程式 Seofon_v2，其不同版本與原版程式 Seofon 進行對戰，皆能夠獲得超過56%的勝率。

關鍵字：麻將、不完全資訊賽局、規則導向、台分

ABSTRACT

In recent years, with the vigorous development of the field of computer games in artificial intelligence, the strength of computer games programs has been promoted dramatically. Mahjong is a multiplayer, probabilistic, imperfect information game. These characteristics increase the complexity and difficulty of Mahjong game.

This thesis will focus on the rules of Taiwan Mahjong game, refer to the related research, and make improvements on the Mahjong program. We follow up on the previous thesis titled “Using the Enhancement Strategy from Discarded Tiles Information to Improve Mahjong Program” that used the rule-based approach and the computation of “deficiency number” to be the main framework. Aiming at the shortcomings of the original program, some corresponding algorithms are proposed for improvement.

The efforts are made in two aspects, offensive and defensive. For the offensive objective, it not only keeps the original framework of computing the “deficiency number” in order to achieve the goal of a quick win, but also wants to get more Tai (equivalent to Faan) scoring units if it is possible. For the defensive objective, it will furthermore focus on avoiding discarding a tile that will make an opponent win the game. It means that the program will try to decrease the loss of scores.

The experimental results show that our proposed algorithms implemented on the program Seofon_v2 have more than 56% win rate against the original program Seofon.

Keywords: Mahjong, Imperfect Information Game, Rule-Based, Faan

目錄

致謝.....	i
摘要.....	ii
ABSTRACT	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
一、緒論.....	1
1.1 研究背景.....	1
1.2 研究目的.....	3
二、文獻探討.....	5
2.1 名詞解釋與比賽規則.....	5
2.2 文獻回顧與相關程式.....	12
三、演算法設計與程式改良.....	15
3.1 字牌棄牌優先度.....	17
3.2 強化獲取特殊牌型台分.....	21
3.3 槓牌機制.....	26
3.4 跟打機制.....	27

四、實驗結果與比賽結果.....	31
4.1 實驗環境設置.....	32
4.2 字牌棄牌優先度版本.....	33
4.3 進攻模組版本.....	34
4.4 跟打機制版本.....	36
4.5 比賽結果.....	38
五、結論與未來展望.....	43
參考文獻.....	46



圖目錄

圖二-1	「萬」牌示意圖.....	8
圖二-2	「筒」牌示意圖.....	8
圖二-3	「條」牌示意圖.....	8
圖二-4	字牌示意圖.....	9
圖二-5	順子範例.....	9
圖二-6	刻子範例.....	9
圖二-7	眼範例.....	9
圖二-8	搭子範例.....	9
圖二-9	聽牌範例.....	10
圖三-1	進胡數範例.....	15
圖三-2	字牌棄牌優先度範例1.....	20
圖三-3	字牌棄牌優先度範例2.....	20
圖三-4	程式選擇棄牌順序.....	22
圖三-5	估值函式 Difficulty 計算範例.....	23

圖三-6	清一色牌型湊牌機制範例	25
圖三-7	字一色牌型湊牌機制範例	25
圖三-8	槓牌機制範例	27
圖三-9	加入跟打機制後程式判斷棄牌流程圖	28
圖四-1	Seofon_v2各版本程式修改順序	31
圖四-2	實驗中程式位置安排	32
圖四-3	相依牌檢查問題範例	39
圖四-4	單張字牌的棄牌問題範例	40
圖四-5	聽孤張問題範例	41



表目錄

表二-1	胡牌台數計算	11
表三-1	不同情形下之字牌額外權重分配	19
表四-1	字牌棄牌優先度版本對上 Seofon 的實驗數據.....	33
表四-2	進攻模組版本對上 Seofon 的實驗數據	34
表四-3	進攻模組版本對上字牌棄牌優先度版本的實驗數據	35
表四-4	跟打機制版本對上 Seofon 的實驗數據	36
表四-5	跟打機制版本對上進攻模組版本的實驗數據	37
表四-6	2022年 ICGA 比賽結果.....	38
表四-7	修正版本對上進攻模組版本的實驗數據.....	41
表四-8	修正版本對上 Zei 的實驗數據	42

一、緒論

1.1 研究背景

電腦對局領域在近年來的蓬勃發展，讓人工智慧一詞更加為人所知，也直接影響了近年來許多電腦對局領域的相關研究，許多具備專業知識的研究者與研究團隊投入了更多的時間與金錢在該領域研發。自從 AlphaGo[1]擊敗世界頂尖棋手李世石後，電腦對局對於人工智慧領域的發展已經具有代表性與重要性，其已經被視為人工智慧發展的標準，電腦對局與人工智慧發展的結果已經有著密不可分的關係。

接下來 DeepMind 團隊又陸續發表了 AlphaGo Zero[2]與 Alpha-Zero[3]，藉由深度學習使人工智慧由原先需要事先得知人類的知識來進行學習的狀態，發展到不需要依靠人類的知識即可進行訓練並學習的狀態。除了先前提到的圍棋外，在許多完全資訊的遊戲中，如西洋棋、象棋等，都已經達到超越人類頂尖玩家的地步，由此可以證明人工智慧在完全資訊的對局遊戲中，已經取得空前絕後的偉大成果。

電腦對局領域的人工智慧發展至今已經有著相當大的變化，從原先像是 Minimax search 與 Alpha-Beta pruning 等方法，參考人類下棋時的思考模式，以最大化自身優勢與最小化對方優勢為目標，模擬得出一個勝率最高的結果，到後來像是 DeepMind 團隊以神經網

路進行深度學習，甚至不需要依靠人類知識即可進行學習。而後甚至有已經開源的 Alpha Zero General[4]，讓許多完全資訊的電腦對局遊戲，都可利用該框架訓練出具有一定水準的人工智慧程式。

除了上述提到的完全資訊的對局遊戲以外，在不完全資訊的遊戲中，像是麻將，也有由微軟所開發的 Suphx[5]，其在本日本麻將平台中已經擊敗當時的人類最高段位玩家。其所利用的技術也是使用神經網路進行訓練，進而達到擊敗人類頂尖玩家的水準。

然而麻將在世界上不同地點有著完全不同的規則與風格，像是日本麻將有著較為嚴謹與複雜的規則，胡牌的條件較難達成，因此在本日本麻將中的進攻選擇便會受到一定程度的限制。但其優點是因為規則較為嚴謹與複雜，因此例外條件會較少。而臺灣麻將因為在規則上較不像日本麻將嚴謹與複雜，因此對於進攻方面的選擇較多，對於胡牌的難度也較低，因此大部分的玩家會著重在進攻這方面。

臺灣麻將發展歷史悠久，甚至有許多知名且專業的麻將比賽。而在電腦對局領域也不例外，對局領域中知名的組織如 TCGA (Taiwan Computer Game Association)、TAAI (Taiwanese Association for Artificial Intelligence)與 ICGA (International Computer Games Association)等，其舉辦的比賽中，麻將皆為其比賽項目之一。麻將受到大部分電腦對局組織所舉辦的比賽所重視，再加上先前提到大部分能夠達到擊敗人類頂尖玩家水準的人工智慧多為完全資訊遊戲，而麻將作為一個不完全資訊遊戲的代表，麻將在人工智慧的研發上有其無可替代的價值存在。

1.2 研究目的

在電腦對局領域發展中，大部分能夠擊敗人類頂尖玩家的人工智慧多數為雙人對戰且完全資訊的遊戲。然而麻將為一個四人且不完全資訊的遊戲，並且因為有著摸牌的特性而含有機率性，雖然日本麻將方面已經有微軟開發出的 Suphx，但因為臺灣麻將的風格與日本麻將差異大，且臺灣麻將尚未有像是 Suphx 一般強大到足以擊敗人類最頂尖玩家的人工智慧，因此關於臺灣麻將的人工智慧研究有其重要性與價值。

麻將作為一個不完全資訊的對局遊戲，除場上已被打出的牌以及自己本身的手牌外，其餘皆為隱藏的資訊，若要使用如 MCTS (Monte Carlo Tree Search) 的方法進行模擬，接著選擇最佳走步的方式，則會遇到非常大的困難。主要原因在於，MCTS 需要以雙方目前的盤面狀況，以麻將來說即為手牌，進行最佳走步的模擬。但問題在於麻將是不完全資訊遊戲，無法得知對方手牌為何。另外，麻將為一個四人遊戲，自己本身的行動會同時影響其他三家，因此很難確認何為自己較佳的行動。

還有因為麻將並非完全依照順序進行的遊戲，在遊戲中可能因為任何一位玩家的動作而改變出牌順序，如吃牌、碰牌與槓牌等動作，都會改變遊戲進行的順序。而該原因也導致如果想用 MCTS 的方法進行模擬，勢必需要將所有不同順序的可能性都進行模擬，如此一來分支數量會非常巨大，會導致運算效能不彰，即使硬體資源強大到有辦法順利計算，但效果可能也會不如預期。

Zeil[8]為林猷琛學長所撰寫的麻將程式，其概念與先前其他麻將程式不同在於：利用棄牌資訊讓程式在棄牌時能夠參考場上非隱藏的資訊進行棄牌選擇，用以達成防守策略的選擇，並且在防守的同時能夠同時兼顧自己的手牌狀態。Seofon[9]為林宗翰學長所撰寫的麻將程式，其是基於 Zeil 的架構下進行改良，改善了原先 Zeil 中存在的一些問題，如槓牌機制的瑕疵，藉此將進攻層面的能力更進一步提升。此外還將原先 Zeil 中所使用的棄牌資訊，加以運用到進攻選擇判斷上，達成在進行防守的同時，還能保有一定程度的進攻能力，令程式能夠在進攻與防守上達成平衡。

本論文中研發的程式 Seofon_v2則將進一步基於 Seofon 的架構進行改良，分別針對原先進攻與防守的部分進行修正，進一步增強其進攻與防守能力。例如強化獲取特殊牌型台分的決策，麻將比賽中是以分數計算勝負，而不同牌型勝利會獲得不同的分數，然而原先的程式中並未針對特殊牌型進行額外的處理。字牌棄牌優先度策略則改良了原先的程式中因為字牌棄牌的問題導致獲取較少台分的問題。此外，原先的程式中，防守策略與一般人類高手玩家的選擇判斷有蠻大程度的差異，Seofon_v2在此部分設法能夠達成參考人類專家防守策略的判斷，達成更佳的防守能力。例如在槓牌機制中，人類玩家在某些情況並不會選擇槓牌，此做法能夠有效降低自己的放槍率。以及像是跟打機制，都是人類專家的防守策略中常使用的部分。

二、文獻探討

2.1 名詞解釋與比賽規則

臺灣麻將的牌型主要分為萬、筒、條、字牌、花牌五種牌型，其中萬、筒、條三種類型的牌，如圖二-1至圖二-3所示。以數字一到九區分為九種牌，因此又稱做序數牌，每種數字皆含有四張牌，例如在「萬」的牌型中，一萬的牌共有四張。而字牌總共分為下列七種牌型，分別是東、西、南、北、中、發、白，如圖二-4所示，每一牌型皆含有四張。花牌的部分因不在比賽規則中採用，因此不在本論文中做討論與說明。

遊戲開始時共分為四個方向的位置，分別為東、南、西、北四個方向，從四名玩家中選取一名玩家做為莊家，並由莊家開始依逆時針方向進行遊戲。每位玩家起始手牌皆為16張，開局莊家由牌牆中取出一張牌加入手牌，而後由手牌挑選一張不需要的牌進行棄牌的動作。棄牌動作完成後，則依照逆時針方向的下一位玩家重複上述動作，在這之中若有玩家勝利則該回合遊戲結束。此外，若牌牆中只剩餘16張，則該回合流局，沒有任何玩家勝利。當回合中若莊家勝利，則下一回合繼續做為莊家進行遊戲，稱做「連莊」。若當回合由非莊家的玩家勝利，則由原先莊家逆時針方向的下一位玩家做為新一回合的莊家開始新遊戲。

除了上述提到的摸牌動作外，玩家還能夠從其餘三種情況獲得

牌，分別為吃牌、碰牌與槓牌，這三種動作皆與其餘玩家的棄牌動作有關。

首先是吃牌，能夠進行吃牌動作的必要條件之一為該張棄牌是由自己的上家所打出，若由自己的對家與下家打出，則無法進行吃牌動作。若要進行吃牌動作，則手牌中需要存在一組順子中的兩張牌，並且與上家打出的該張棄牌能夠組成順子的狀態，才能夠完成吃牌的動作。例如手中擁有「三萬」、「四萬」兩張牌，並且上家打出「五萬」，此時便能夠進行吃牌動作。在進行吃牌動作後，需要將組成順子的兩張牌與該張棄牌移出手牌組成完成組。並且從自己手牌中選擇一張牌進行棄牌動作，才算完成吃牌動作的流程，並由進行吃牌玩家的下家繼續進行遊戲。

碰牌則為當玩家手中擁有兩張相同的牌時，並且由其餘玩家打出相同的牌，便能夠進行碰牌的動作，形成刻子。碰牌的動作與吃牌不同，沒有限制需要由上家打出，其他任何一位玩家打出相同的第三張牌時，皆能夠進行碰牌的動作，並且若碰牌與吃牌動作同時發生，則碰牌優先度較高。碰牌的動作與吃牌相似，在拿取該張牌後，則將成組的刻子移出手牌組成完成組。並且從自己手牌中選擇一張牌進行棄牌動作，完成碰牌動作的流程，並由進行碰牌玩家的下家繼續進行遊戲。

槓牌的部分又分為三種情況，分為明槓、暗槓與補槓。明槓動作為當手牌中存在一組刻子，而對家或下家打出相同的牌時，便能夠進行明槓的動作，與碰牌差別在於打出該張棄牌的玩家不能為上家，並將成組的四張牌移出手牌。暗槓則為手牌中存在一組刻子時，

並且經由自己的摸牌動作獲得第四張相同的牌時，可以將成組的牌移出，並且正面朝下不讓其餘玩家得知暗槓的牌。補槓則為進行碰牌形成刻子，並且移出手牌的成組牌，在後續又經由摸牌動作摸到第四張相同牌時，能夠進行補槓的動作。無論是哪種槓牌形式，在形成槓牌的成組後，皆需要從牌牆的尾端進行摸牌的動作。並且從自己手牌中選擇一張牌進行棄牌動作，完成槓牌動作的流程，並由進行槓牌玩家的下家繼續進行遊戲。

整場遊戲最後的獲勝條件則為分數最高者，每一回合皆能夠通過胡牌的動作獲得分數，亦或是透過放槍的動作失去分數。在比賽規則中，要達成胡牌，則需要組成五個「完成組」加上一組「眼」，若玩家只差一張牌即可達成胡牌條件，此情況稱為「聽牌」(如圖二-9所示)。完成組由順子(如圖二-5所示)、刻子(如圖二-6所示)與槓所組成，換句話說，任一組三張連續的牌、三張相同的牌或是四張相同的牌，都能構成一組完成組。眼(如圖二-7所示)的部分則由兩張相同的牌組成，例如兩張「三條」。

在先前提到的麻將對局比賽中，目前是採用由國立臺北大學陳志昌教授的實驗室所開發的麻將對弈平台進行比賽，與先前所使用的 Cycgame 麻將平台[6]比賽規則相同，其分數計算與主要規則如下：

1. 比賽不使用花牌，其餘萬、筒、條、字牌皆有使用。
2. 每回合底分為1000分，另外每台為500分。
3. 比賽中不採用連莊，即莊家每回合皆會換人，莊家計一台。

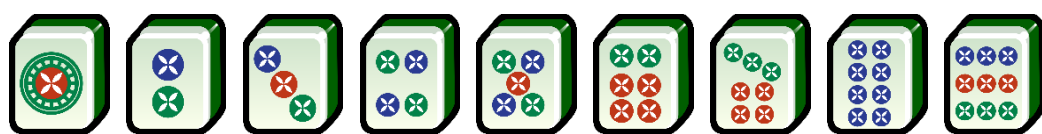
4. 牌山更新率為一圈，即每位玩家都能拿到相同的起始手牌。

5. 比賽中玩家會依照相對位置輪流交換位置。

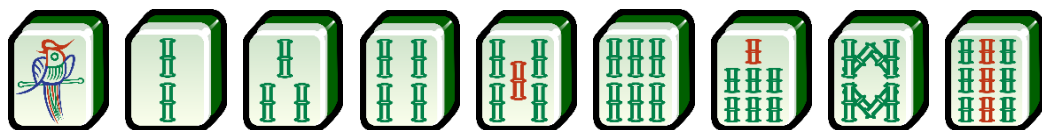
每場比賽共進行384局，每局的計分方式為，若是達成胡牌時是由其中一位玩家的棄牌達成，則該玩家的行為稱為放槍，並且由胡牌方取得該玩家方的分數，取得的分數為每局底分1000分加上台數乘上500分。若玩家是透過自己摸牌的動作達成胡牌，則稱為「自摸」。此時由自摸的玩家取得其餘三位玩家的分數，取得的分數為每局底分加上台數乘上500分。因此自摸會得到胡牌三倍的分數。此外，比賽規則中有說明幾種特殊的胡牌牌型，其各自代表不同台數。胡牌台數計算如表二-1所示。



圖二-1 「萬」牌示意圖



圖二-2 「筒」牌示意圖



圖二-3 「條」牌示意圖



圖二-4 字牌示意圖



圖二-5 順子範例



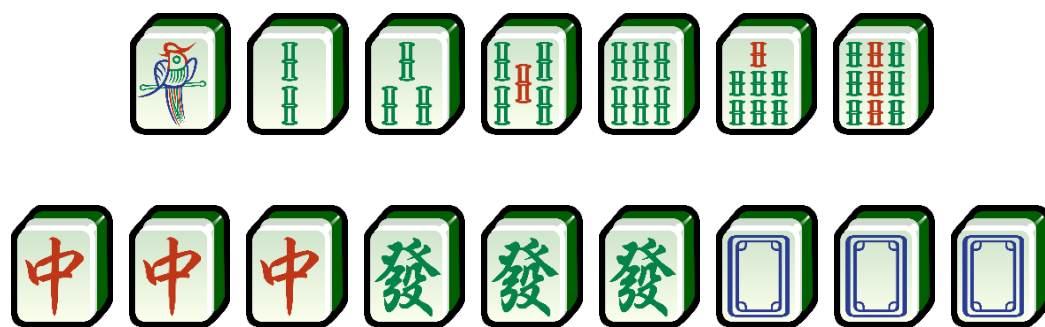
圖二-6 刻子範例



圖二-7 眼範例



圖二-8 搭子範例



圖二-9 聽牌範例

台類名稱	台數	說明
莊家	1	只適用莊家，莊家胡別人可以多算一台，被胡(包括平家自摸)也要多賠一台。如果是莊家自摸，則其他三家都要多算一台。
門清	1	胡牌時無吃碰槓等情形，全部牌都是自己摸到的。
自摸	1	最後胡牌的那一張牌為自己摸進的牌。
斷么九	1	胡牌時牌裡均無一、九牌或字牌。
海底撈月	1	在此局的最後一張牌自摸而胡牌。
門風刻	1	開牌時，位置反時針方向計算風位，依序為東南西北；胡牌時有屬於自己風位的刻子，多加一台。
三元牌	1	有紅中、白板、或青發的刻子或槓。例如：中中中。
雙龍抱	1	胡牌時，牌組為門清並牌裡有兩副同花色同數字的順子牌，例：112233萬。
槓上開花	1	槓牌後補牌或補花就胡牌可加算槓上開花。
眼牌	1	宣告眼牌後被胡牌會被扣一台，但是胡別人或自摸者並不加一台。
搶槓	1	別家喊槓所明槓之牌，恰為自己胡的牌，有權力可搶槓胡牌
全求	2	胡牌時，牌面只剩一張單吊牌，其餘全部為吃或碰。(自摸不計)
平胡	2	胡牌時牌裡只有順子而沒有刻子、槓子及字牌。(獨聽、自摸不計)。
混帶么	2	胡牌時，牌裡的對子、順子、刻子或槓子均含么九牌或由字牌組成。
三色同順	2	胡牌時，牌裡有三種花色的同一種順子，門清者此台數可乘上兩倍，等於計為四台。

台類名稱	台數	說明
一條龍	2	胡牌時，牌裡有同種花色1~9齊全的牌，門清者此台數可乘上兩倍，等於計為四台。
雙雙龍抱	2	胡牌時，牌組是門清且牌裡有兩組雙龍抱，例如：112233萬、556677索。
三暗刻	2	牌面有三組暗刻。
三色同刻	2	三個數相同的刻子(槓)。
門清自摸	3	牌面同時有門清和自摸。
碰碰胡	4	胡牌時，由刻子(或槓子)以及一個眼所組成。
混一色	4	胡牌時，牌由字牌及同一種花色序數牌組成。
純帶么	4	胡牌時，牌裡的對子、順子、刻子或槓子均含么九牌。
混老頭	4	胡牌時，均由一、九牌和字牌的刻子及眼所組成。不計碰碰胡。
小三元	4	胡牌時，牌裡中發白其中一個為眼，另兩個為刻子或槓子。
四暗刻	6	胡牌時，牌面有四個暗刻(槓)。
四槓子	6	胡牌時，牌面有四個槓子所組成。
大三元	8	胡牌時，牌面有中、發、白三組刻子。
小四喜	8	胡牌時，由風牌的三組刻子及將牌所組成。
清一色	8	胡牌時，牌均由同一種花色序數牌所組成。
字一色	8	胡牌時，由字牌的刻子(槓子)及將所組成。
五暗刻	8	胡牌時，牌面有五個暗刻(槓)。
清老頭	8	胡牌時，均由序數牌的一、九刻子所組成。
大四喜	16	胡牌時，牌面由東、南、西、北四種風牌的刻子(槓子)所組成。
天胡	16	莊家的特權，開始牌局摸牌就胡
地胡	16	平家的特權，開始牌局摸牌就自摸
人胡	16	在第一輪就有人放槍

表二-1 胡牌台數計算

2.2 文獻回顧與相關程式

麻將的勝利條件在2.1節中有說明，即為獲得五個「完成組」加上一個「眼」，在判斷完成組的組合情況中，電腦程式與人類玩家在判斷上有非常大的差異。為了解決程式在判斷上有可能發生的問題，過往的幾支程式，如 ThousandWind[10]、MahjongDaXia[11]、Zei 以及 Seofon 等都採用資料庫的方式解決上述可能發生的問題，直接從資料庫得知當前手牌狀況，即可避免程式在判斷上的錯誤情況。

因本論文章式主要針對 Zei 與 Seofon 進行改良，而 Zei 則是引用了部分 MahjongDaXia 的概念，因此下面大致探討這三支程式。

MahjongDaXia 在針對手牌處理主要是將手牌先進行拆分，並將拆分後的情況進行事先規定好的標記，並將標記好的情況輸入一完成組與未完成組的資料庫，即可得知當前手牌狀態。而棄牌的部分則是針對事先設定好的判斷條件，依照不同條件的重要性進行排序，即可得到對於當前手牌來說最佳的棄牌。舉例來說，丟棄某張牌能夠達成較高的完成組數，即為較佳的棄牌選擇。

Zei 在牌型判斷的概念上與 MahjongDaXia 有非常大的不同，Zei 主要將萬、筒、條三種牌型與字牌分開，針對前面三種牌型，利用資料庫即可得知當前完成組的狀態，並且在進牌與棄牌都參考了 Let's Play Mahjong[7]論文中進胡數的概念，以減少進胡數為目標進行進牌與棄牌的動作選擇參考。並且 Zei 進一步增加了防守模組，

利用場上已知的棄牌資訊，推論其他玩家手牌狀況進行防守。

Seofon 主要將 Zei 程式中所碰到的問題進一步進行改良，如字牌的處理、棄牌選擇的缺失與槓牌機制改善等。

在字牌處理的部分主要針對字牌的權重進行調整，在原先 Zei 中，字牌有著較高的機會被當作棄牌，可是在麻將比賽規則中某些字牌擁有台數，例如門風刻與三元牌，可能會因為較高的棄牌機率而損失部分分數，此外也令原先已經成「眼」的字牌被迫進行棄牌的動作。

在棄牌選擇缺失的部分，因為 Zei 主要以減少進胡數為目標進行棄牌動作，該動作沒有辦法事先判斷每一張牌的相依牌狀況，有可能造成即便減少了進胡數，但因為最後始終無法組成完成組造成當回合無法胡牌的情況出現。因此，Seofon 在這部分的做法為，在程式計算進胡數之前，優先判斷所有牌的相依牌剩餘狀況。若已經無法組成完成組，則不管進胡數計算結果如何，優先當作棄牌打出。

而槓牌機制改善的部分主要是因為在 Zei 中槓牌擁有較高的優先度，許多時候會因為優先執行槓牌動作而造成手牌被拆分的狀況，Seofon 在此的解決方法大致上為，槓牌前事先判斷是否有相依牌能與其成組，若有則不進行槓牌動作。

在防守模組的改良方面，主要是針對防守模組啟動的時機進行調整，以避免過早或過晚才啟動防守模組，導致防守效率不彰的問題。其方法為進行序數牌的判斷，一位玩家每打出一張序數牌，則提前對其啟動防守模組。並且在防守模組中，加入棄牌資訊的策略，

考量需求牌與非需求牌之間的相依牌，調整其權重以便能夠達成更完善的防守策略。

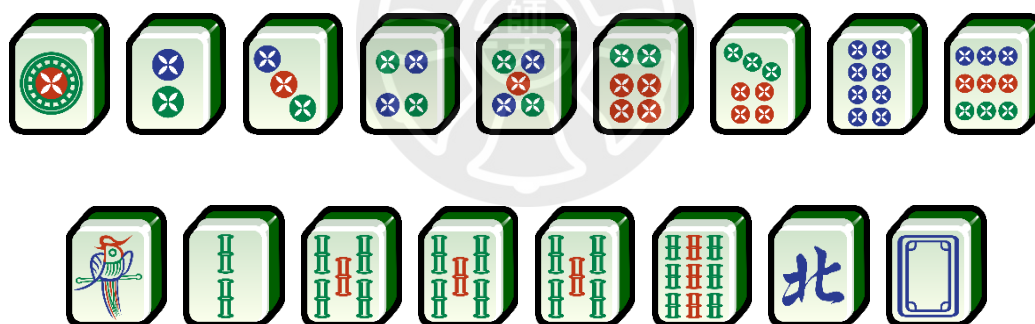
此外還有 VeryLongCat[12]麻將程式，不考慮對手丟牌，只計算自己的摸牌與捨牌狀態，以含有轉張概念的期望最大值搜尋法決定摸牌胡牌機率與捨牌胡牌機率，另外搭配估價函式判斷吃、碰、槓等情況，選擇出最佳的湊牌策略。



三、演算法設計與程式改良

在2.2節中提到，Seofon 主要針對 Zei 程式的缺點進行修正與改良，Seofon 在進攻與防守的概念上大致上與 Zei 程式差異不大，進攻部分兩者皆是以減少進胡數為目標進行湊牌動作，防守部分兩者都參考了場上棄牌資訊的狀態進行防守時機的選擇與防守動作的考量。

進胡數出自 Let's Play Mahjong 論文，進胡數表示該副手牌還需更換幾張牌才能達成胡牌狀態。以下圖三-1為例說明，該副手牌最少還需要更換兩張牌才能達成胡牌狀態，因此進胡數即為2。



圖三-1 進胡數範例

接著說明進胡數計算方式。計算進胡數的主要方法為計算該副手牌各花色的完成組數、眼是否存在以及搭子數，序數牌部分搭配資料庫進行計算處理，以上圖三-1為例，手牌會被依照花色先分別標記為：

'000000000'、'111111111'、'110030001'、'0001001'

前三組分別代表三種序數牌萬、筒、條，並將其分別輸入資料庫後，即可得到標記為：

‘000’、‘300’、‘101’

而字牌的部分因為其與序數牌不同，不需要判斷順子的完成組，因此只需要從當前手牌狀態，不需要輸入資料庫即可判斷。以上述字牌標記‘0001001’為例，經過計算後會被標記為‘000’。輸入資料庫後或經由計算後的三個數字代表意義分別為：「完成組數」、「眼是否存在」以及「搭子數」。

因為上圖三-1中手牌張數為17張，若要達成胡牌狀態總共需要五個完成組加上一個眼，由上述四組數字合併後即可得知當前手牌狀態。以上述圖三-1為例，將四組數字相加後會得到‘401’，代表該副手牌共有「四個完成組」、「不存在眼」以及「一個搭子」。因此只需要將三張孤張中的其中一張更換成能與搭子形成完成組的牌，以及將剩餘的兩張孤張組成一組眼即可達成胡牌狀態，最後得到進胡數為2。

在本論文中，Seofon_v2在程式架構上主要延續 Seofon 程式的作法，採用進胡數的計算為主要架構，並針對 Seofon 程式上的缺失進行修正與改良，進一步針對進攻與防守兩個部分進行改良。在進攻的部分分為兩個部分：字牌棄牌優先度與強化特殊牌型得分。其主要考量為因為麻將比賽規則中除了底分外，還有計算台分，而不同的牌型擁有不一樣的台分，Zei 與 Seofon 主要只參考了進胡數進行湊牌動作，雖然能夠快速達成胡牌條件，但可能會因此損失部分台分的取得。

在防守的部分分為兩個部分：槓牌機制改良與跟打機制加入，原先 Zei 與 Seofon 主要是針對場上棄牌資訊或是參考已打出的序數牌狀態進行防守動作，但這與一般人類頂尖玩家打麻將時的防守動作上有很大的出入，因此在防守的部分會在額外加入如同一般人類玩家在進行防守時的動作。

總結來說，Seofon_v2在進攻與防守方面都進行了改良，在快速胡牌的臺灣麻將風格中，期望能獲得更高的得分。而在防守方面，因為要加入模擬人類玩家防守時的動作，勢必需要選擇更精確的防守時機，而這部分也會造成防守時將會損失部分進攻的機會，在策略上需要作出取捨。

3.1 字牌棄牌優先度

字牌與序數牌不同，字牌若要形成完成組，只能由刻子或是槓子所組成。因此在胡牌狀態中，字牌通常以刻子或是眼的狀態保存在手牌中。「中」、「發」、「白」又稱三元牌，若其作為完成組保留在胡牌的牌型中，可以額外獲取台分。此外每位玩家在每局開始時都會有其對應的門風，與門風相同的風牌即為門風牌，門風牌也擁有額外的台分。

在 Seonfon 程式中，解決了先前 Zei 程式在字牌中所面臨的問題，在 Zei 程式中字牌並無法被特別保留下來，因此可能損失部分台分的獲得。而在 Seonfon 程式中，為了解決該問題，則賦予同一

種字牌存在兩張以上時額外的權重，讓該字牌能夠有較高的機會被保留下來形成完成組。若無法組成完成組，也能夠被當作眼保留在手牌中形成胡牌狀態。

上述做法雖然確保字牌有更高的機會被留在手牌中組成胡牌牌型，但若根據規則來看，擁有台分的字牌其實只有三元牌以及門風牌，在 Seofon 程式中並未針對不同字牌之間給予不同的優先度。導致在棄牌時，中、發、白三元牌有著較高的機率比四張風牌被優先打出，無法有效的獲取應該獲得的台分。

因此需要針對不同的字牌給予不同的權重，沒有台分的風牌應該賦予較高的額外權重，擁有台分的三元牌與門風牌應該賦予較低的額外權重。確保擁有台分的字牌能夠有較高優先度被保留在手中，而沒有台分的字牌能夠優先被選擇為棄牌打出。

在不同的情況下也需要調整其權重，例如字牌擁有不同的數量時，也需要分別考慮，由於字牌存在兩張至少能夠確保被作為眼達成胡牌條件，也相對較容易形成完成組，因此若不同字牌分別存在一張與兩張的情況下，單獨存在一張的字牌應該有著較高的優先度被選擇為棄牌。

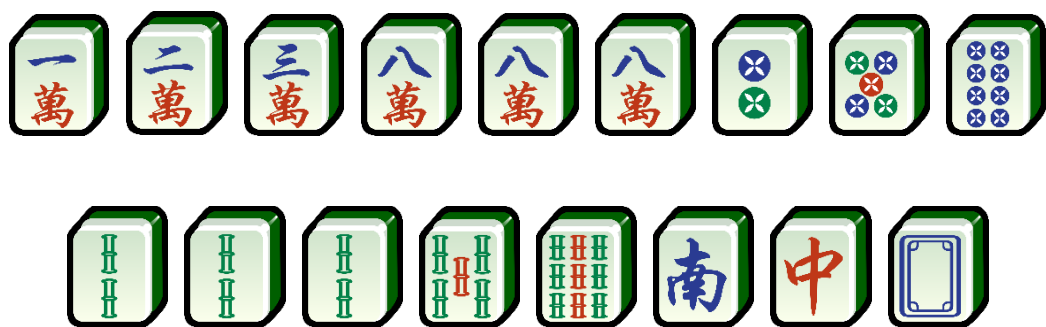
此外，在原先的作法中只有考慮如果某張字牌已經沒有剩餘張數時，該張字牌會被優先選擇為棄牌。但由於此改良作法考量的前提為確保字牌台分的獲取，因此也需要考慮擁有台分的三元牌與門風牌如果已經無法形成完成組的狀態下，也應該將其視為不擁有台分的字牌，賦予較高的額外權重，確保該張字牌能夠優先被選擇為棄牌。

根據上述提到的不同狀況，實作中詳細分為八種情況。八種情況被用來判斷應該賦予某張字牌多少的額外權重分，詳細的權重分配方式如表三-1中所示。

字牌種類	風牌數量	三元牌數量	三元牌 是否能成組	風牌 是否能成組	額外權重分
風牌	1	0	X	X	10
風牌	1	1	是	X	90
風牌	2	2	是	X	90
風牌	1 or 2	1 or 2	否	X	15
三元牌	0	1	X	X	10
三元牌	1	1	X	是	30
三元牌	2	2	X	是	30
三元牌	1 or 2	1 or 2	X	否	10

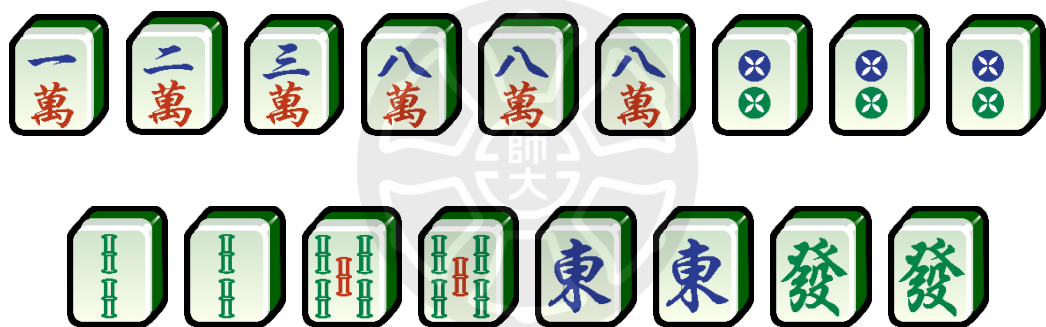
表三-1 不同情形下之字牌額外權重分配

以下圖三-2作為範例，原先程式將會選擇白作為棄牌，在改良後會根據不同情況選擇不同的棄牌。若中、白還有機會形成完成組，經過計算後，南的權重分為90，中、白的權重分為30，則程式會優先選擇南作為棄牌。若中的剩餘張數為1，南的剩餘張數為2，代表中無法形成完成組，但南有機會形成完成組，因此無法從中獲取台分，此時中會被選擇為棄牌，相同狀況也適用於白。



圖三-2 字牌棄牌優先度範例1

以圖三-3為範例，假設該副手牌中，發已經無法形成完成組，但東仍有機會時，東的權重分為15，發的權重分為30，程式此時會選擇棄牌為發。



圖三-3 字牌棄牌優先度範例2

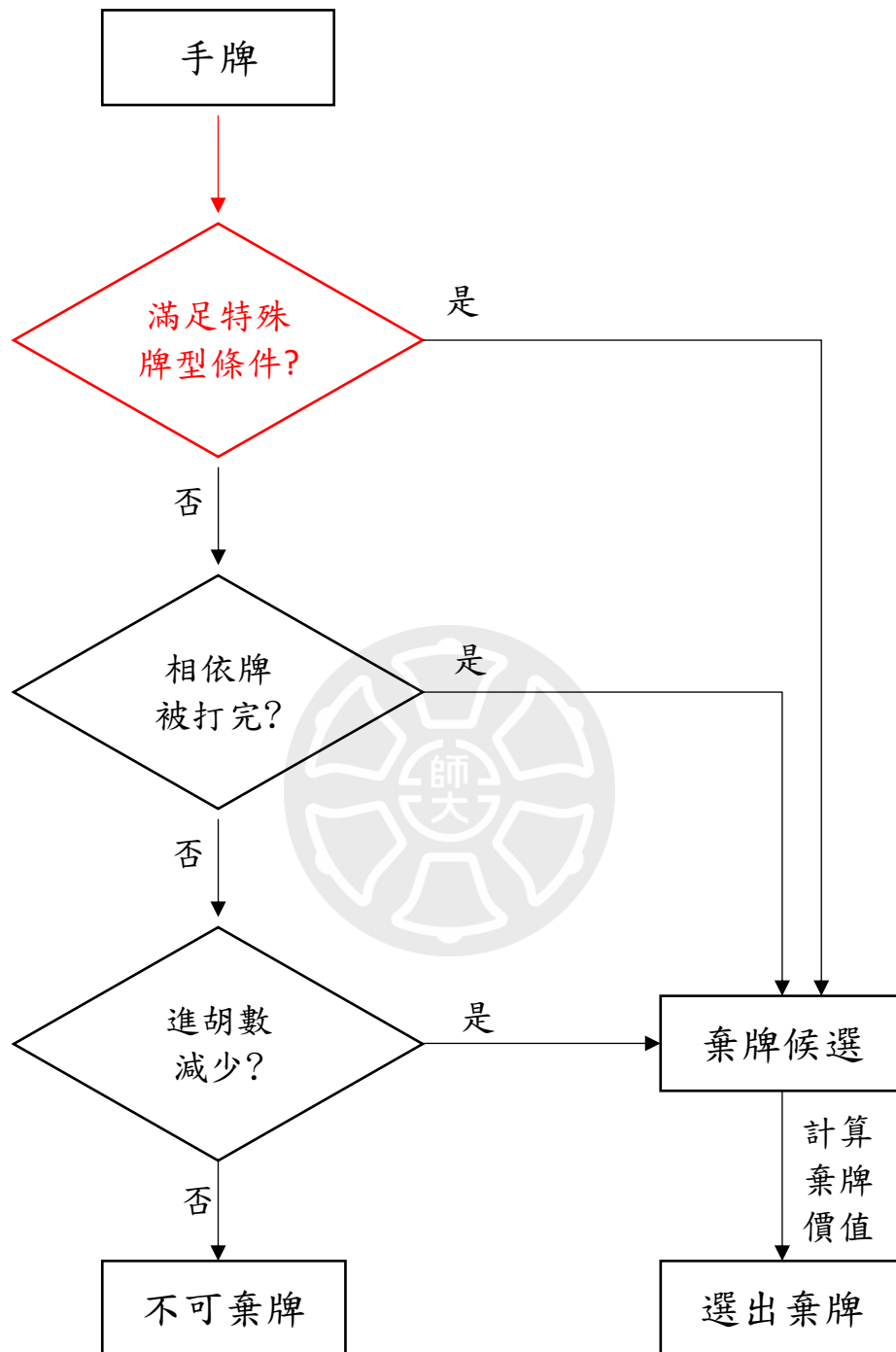
3.2 強化獲取特殊牌型台分

臺灣麻將與日本麻將最大的差別在於，臺灣麻將對於胡牌條件的規定較不嚴格，因此也造成臺灣麻將的風格通常都以快速胡牌作為主要的進攻目標。如同第三章中所述，Zei 與 Seofon 所考量的進胡數也是快速達成胡牌條件的進攻策略。但因為臺灣麻將比賽規則中有規定不同的胡牌牌型擁有不同的台數(表二-1)，而其中某些牌型雖然獲得的台數與其他牌型相同，但其條件達成的難易度卻相較其他牌型容易許多。因此這些相對容易達成條件的牌型，在滿足某些特別的情況時，若能夠將其視為優先湊牌的選擇將會非常有價值。

因為在原先的程式中，包括 Zei 與 Seofon 在進攻選擇上皆以進胡數為主要架構，在進攻湊牌的棄牌選擇上皆以計算進胡數為參考依據。而一般人類玩家在胡牌的牌型選擇上，在滿足某些條件時會選擇特殊牌型，最主要原因為能夠獲取較高的台分數。然而因為進胡數只單純計算當前手牌距離胡牌狀態還差幾張牌，若單純只以進胡數作為湊牌架構，在大部分的情況下絕對無法以特殊牌型達成胡牌條件。因此該方法的主要目標為當目前手牌若距離某些特殊牌型的胡牌條件已經非常接近時，能夠進一步確保在棄牌的選擇上以這些特殊牌型為目標進行湊牌動作，避免在有較大的機會達成特殊牌型胡牌條件時，卻無法把握該機會獲得較高的台數。

在原先的程式架構中，主要的棄牌選擇次序如圖三-4所示。程式會先判斷該副手牌中某張牌的相依牌是否已經被打完，接著考慮進胡數是否增加，得出可棄牌名單後進行棄牌價值的計算(3.1節中

提到的方法主要在此部分進行實現), 最後選出棄牌。



圖三-4 程式選擇棄牌順序

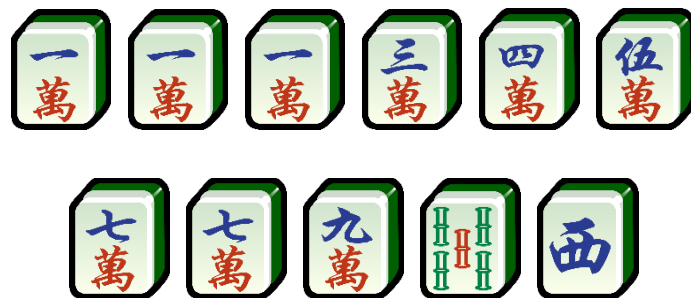
程式主要架構仍然延續進胡數計算的方式, 但需要額外的條件判斷來輔助該方法, 如圖三-4中紅色部分所示。主要做法為條件限

制與估值函式並行的方式。條件限制主要判斷該副手牌是否已經足夠接近特殊牌型的胡牌條件，用以控制湊牌的啟動時機，而估值函式則判斷是否擁有足夠價值進行湊牌動作。程式判斷是否進行特殊牌型湊牌為上述兩個條件皆須滿足。估值函式計算方式如下：

$$\text{Value} = \text{Difficulty} * \text{牌型台數}$$

$$\text{Difficulty} = \begin{cases} \text{Max}(\text{Remaining Card})/4, & \text{當滿足條件之花色湊牌完成} \\ \text{Min}(\text{Remaining Card})/4, & \text{當滿足條件之花色未湊牌完成} \end{cases}$$

牌型台數為該牌型可獲得台數，Difficulty 代表完成該牌型的難易程度。若滿足該牌型條件的花色湊牌完成(例如：萬子清一色判斷萬子)，也就是該花色只存在完成組與眼的狀況下，則取滿足條件的花色中，剩餘張數最大者作為 Difficulty 的分子。反之，若尚未湊牌完成，也就是該花色存在孤張或是搭子的狀況下，則取滿足條件的花色中，剩餘張數最小者作為 Difficulty 的分子。以下圖三-5為例，若已經啟動清一色牌型的湊牌動作時，由於萬牌尚未湊牌完成，因此 Difficulty 將會取所有萬牌剩餘張數中最小的數字，例如最小剩餘張數為5萬的1張，此處的 Difficulty 即為1/4。反之若萬牌在此處已經湊牌完成，則會取剩餘張數中最大的數字作為 Difficulty。

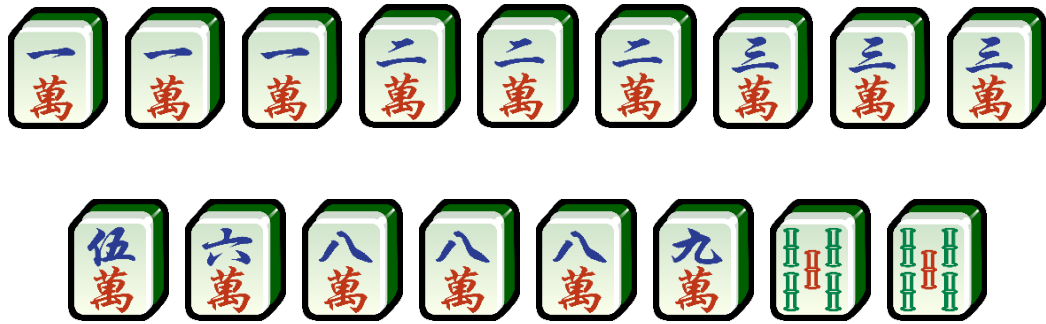


圖三-5 估值函式 Difficulty 計算範例

此外，由於胡牌時的牌型判斷也會將已經吃、碰、槓的牌納入計算，因此也需要將已經吃、碰、槓的牌加入條件限制的考量之中。在 Seofon_v2 中針對許多不同的特殊牌型進行強化得分的策略，例如清一色、字一色、大四喜等。接下來將以幾種牌型作為範例與代表，說明針對特殊牌型強化得分的策略是如何進行。

首先是「清一色」牌型，清一色牌型擁有8台，清一色牌型的胡牌條件為當胡牌時只能存在單一序數牌。因此在判斷是否足夠接近清一色牌型的胡牌條件時需要考慮的即是單一花色的序數牌數量，若當前手牌中某一花色的序數牌數量超過某一數量時，即滿足第一階段的條件限制。要特別注意的是，此處要納入考量的還有前面提到的已經吃、碰、槓的牌，因為這些牌也會納入胡牌條件的判斷中。

接著會進行估值函式的計算，以下圖三-6為範例，該處假設上述第一階段的條件限制已經滿足。清一色牌型的台數為8，因此牌型台數為8。而此處由於萬牌尚未湊牌完成，此處 Difficulty 將會取最小剩餘張數，此處假設最小剩餘張數即為一萬的1張，因此 Difficulty 即為1/4，將兩數相乘便會得到 $Value = 1/4 * 8 = 2$ 。最後便會將 Value 的值與先前設定好適用於各牌型的比較值進行比較，若滿足條件則此階段的條件限制也滿足，加上先前第一階段滿足的條件限制，便會啟動清一色牌型的湊牌動作。在圖三-6中，若沒有啟動該機制，選擇的棄牌將會是九萬，然而在啟動該機制後，選擇的棄牌將會是五條。此範例說明在非常接近清一色牌型胡牌條件的狀態下，程式會選擇朝向清一色牌型進行湊牌，而若在先前的程式中，則會失去可以獲得額外台數的機會。



圖三-6 清一色牌型湊牌機制範例

再來以「字一色」牌型作為範例說明，字一色牌型同樣擁有8台，字一色牌型的胡牌條件為當胡牌時只能存在字牌。因此在判斷是否足夠接近字一色牌型的胡牌條件時需要考慮的即是字牌的數量，若當前手牌中字牌數量超過某一數量時，即滿足第一階段的條件限制。同樣要特別注意的是，此處要納入考量的還有前面提到的已經吃、碰、槓的牌，因為這些牌也會納入胡牌條件的判斷中。

接著會進行估值函式的計算，以下圖三-7為範例，該處同樣假設上述第一階段的條件限制已經滿足。最後計算得出的 Value 值同樣也是2，在接著與先前設定好的比較值相比，最後判斷是否要啟動字一色牌型的湊牌動作。以圖三-7來說，若啟動該機制則棄牌為一萬，反之則為東。



圖三-7 字一色牌型湊牌機制範例

如同3.2節前半部分所述，該方法能夠確保在有機會做大牌時不會錯過較好的機會，雖然此方法會損失部分快速胡牌的機會，但由於麻將比賽最後勝負還是由分數進行判斷。因此除了快速胡牌的策略外，如何最大化獲取分數的策略也是非常重要的考量因素之一。

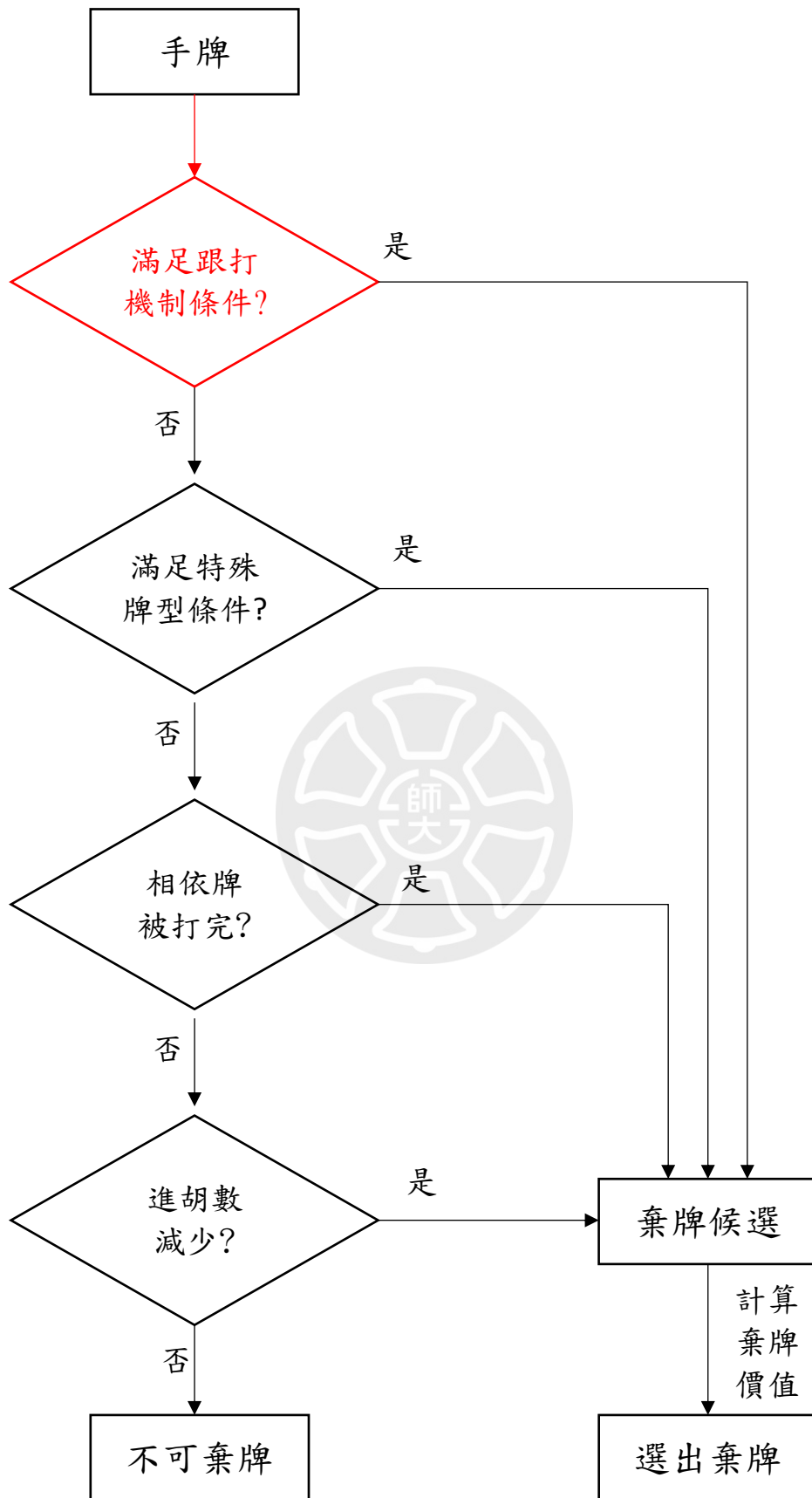
3.3 槓牌機制

先前的 Zei 程式在槓牌時有一定的機率會將手牌中已經成組的牌拆分，造成湊牌與胡牌效率不佳，而 Seofon 解決了該問題，主要採用的方法為吃牌與槓牌的混合機制。

由於規則上規定明槓後無法胡牌，因此若在聽牌時選擇明槓，非但無法胡牌還會增加自己的放槍機率，而先前的程式並未考慮該重點。雖然 Seofon 解決了槓牌會拆分手牌的問題，但因為槓牌的優先度較高，在不會拆分手牌的前提下仍然會選擇槓牌，此時若已經聽牌，並且槓牌的種類也是明槓，就會造成上述無法胡牌且額外增加放槍率的問題。

此處採用進胡數與槓牌並行的做法，由於槓牌仍為湊牌的手段之一，因此只需要利用進胡數計算當前手牌是否已經胡牌，並結合槓牌種類的判斷。當槓牌種類為明槓且已經達成聽牌狀態，則不進行槓牌，反之則正常進行槓牌動作確保湊牌手段。

圖三-8為槓牌機制範例，若此時對家打出二筒，在原先的程式



圖三-9 加入跟打機制後程式判斷棄牌流程圖

而在 Zei 與 Seofon 中並未有此判斷與動作，因此在原先的防守架構上，額外加入跟打機制，使得程式能夠做出類似於人類頂尖玩家防守時的行為。若無相同牌能進行跟打的動作，則進行原先程式的防守動作，繼續在防守的同時能保有一定程度的進攻。

若要額外加入跟打機制，需要考量的重要因素主要有兩個，第一個是啟動時機，由於若啟動跟打機制，則代表此時是完全的防守狀態，也就是有非常高的機率會拆分目前的手牌。因此若過早啟動跟打機制，會導致胡牌效率非常差，需要注意不能過早啟動跟打機制。另外，若當前手牌狀態已經是聽牌狀態，此時若啟動跟打機制，雖然能夠有較高的安全性與較低的放槍率，但也會完全放棄已經非常接近胡牌條件的該副手牌。

因此這個部分採用進胡數與條件判斷的並行機制，進胡數用來計算當前手牌是否已經聽牌，如果已經聽牌則不進行跟打機制。條件判斷用來判斷跟打機制的啟動時機，過早會導致湊牌效率不佳，而過晚也會導致防守效率不佳。此部分則採用與原先防守模組混合的方式，在原先的防守模組中會參考其他三位玩家的棄牌資訊，以及打出的序數牌數量，因此便能夠以此資訊加入跟打時機的判斷標準，調整出適合的跟打機制啟動時機。

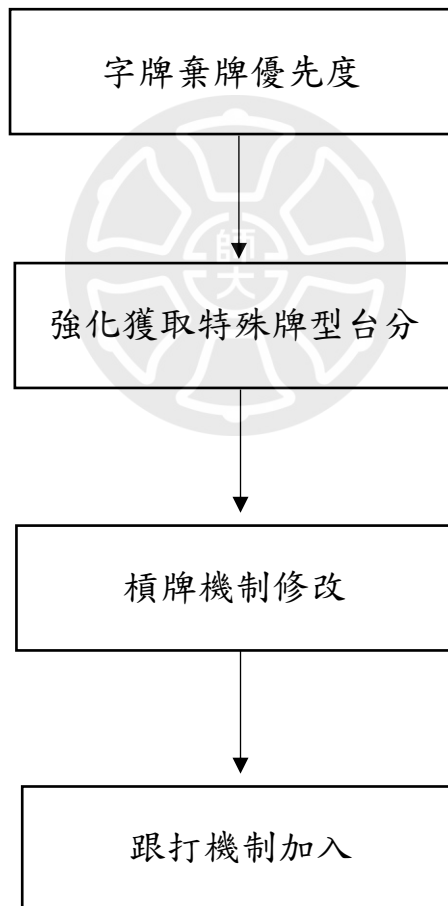
此外，在同一圈中的棄牌，其安全性相對較高，因此將跟打機制中的棄牌數量增加至前三張棄牌，並設有不同的優先度，距離棄牌選擇當下越近的棄牌，擁有較高的優先度。此部分需要特別注意的是，若將棄牌範圍的回合數往前涵蓋太遠，會有較高的機率造成放槍，原因主要為若某張牌在當時並未能夠讓某位玩家達成胡牌狀

態，但後來經由湊牌後，該位玩家聽了該張牌，此時便會造成放槍，因此這個部分的做法是將跟打機制選牌的範圍涵蓋到當圈，也就是往前推三手的棄牌才納入考量範圍。



四、實驗結果與比賽結果

本章將說明實驗環境設置，以及各版本程式的實驗結果。實驗以臺北大學陳志昌教授實驗室所開發的平台進行。各版本程式將以 Seofon 程式作為比較對象，展示各版本程式與 Seofon 程式之間的比賽結果。此外，各版本的程式也將與上一版本的程式做比較，下圖四-1為 Seofon_v2各版本程式修改順序。



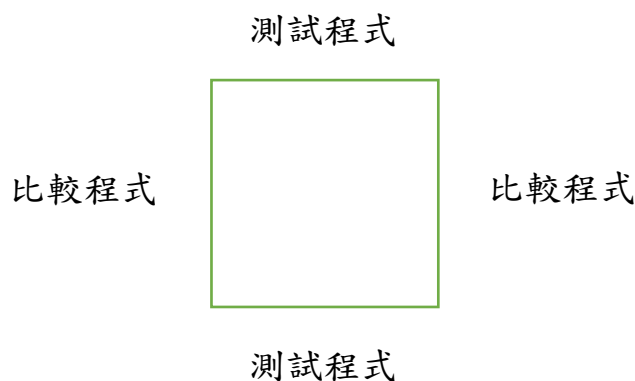
圖四-1 Seofon_v2各版本程式修改順序

本章分為五節，4.1節將會說明實驗環境設置，4.2、4.3與4.4節將說明各版本程式的實驗結果，4.5節將說明比賽結果與程式修正。

4.1 實驗環境設置

目前比賽採用每場對戰進行384局為單位，並以最終分數判定勝負。實驗中設定將與比賽規則大致相同，每個版本程式將與比較對象進行共30場比賽，每場比賽皆進行384局，每種對戰組合皆進行11520局。位置安排上，相同程式安排在對家位置，上下家皆為另一支程式，位置安排如圖四-2所示。牌堆更新率為每圈更新，確保每副起始手牌四位玩家皆能輪流獲得。並且不採用連莊，亦即四位玩家將會輪流擔任莊家。每局底分1000分，每台500分，台數計算參考 Cycgame 平台規則，如表二-1所示。

圖四-2 實驗中程式位置安排



實驗結果會將相同程式的分數、他家放槍數、自摸數加總，與

另一支程式做比較，每場比賽的勝負皆以分數高低判斷，最後分別比較兩支程式的他家放槍數、自摸數、胡牌率、分數以及勝率。

雖然相同程式之間可能相互胡牌、放槍，但因為相同程式的分數會經由加總後才與另一支程式比較，因此相同程式之間的胡牌、放槍動作並不會影響最後的比較結果，並且在程式強弱比較的表現上也比較容易解讀。此外也有加入他家放槍數、自摸數、勝率等數據進行比較，這些數據也都能夠作為判斷一支程式強弱的依據。

4.2 字牌棄牌優先度版本

首先第一個版本為3.1節中提到的字牌棄牌優先度版本，命名為Seofon_v2_{Honor}，由原先的Seofon程式中改良了字牌處理方面的問題。實驗比較對象為原版Seofon程式，總共進行30場比賽，每場比賽進行384局，總共進行11520局。實驗結果如表四-1所示。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2 _{Honor}	4414	1370	50.2%	682000	60%
Seofon	4040	1294	46.3%	-682000	40%

表四-1 字牌棄牌優先度版本對上Seofon的實驗數據

實驗結果數據顯示，字牌棄牌優先度版本在他家放槍數、自摸

數皆比原版 Seofon 多。表示在原先的字牌處理機制上，可能存在損失部分湊牌機會的問題。也就是字牌棄牌的順序導致原本可能存在湊牌的機會卻被當作棄牌打出，進而影響最後的湊牌效率，導致他家放槍數與自摸數較低。此外在分數的獲取上，也顯示出原版 Seofon 損失了部分分數的獲取，即使每種三元牌只有1台，隨著局數的增加，三元牌台數的損失便會進一步擴大。

4.3 進攻模組版本

第二個版本為3.2、3.3兩節中提到的兩個方法：強化獲取特殊牌型台分與槓牌機制混合的版本，命名為 Seofon_v2_{Offensive}，由4.2節中字牌棄牌優先度版本，進一步加入強化獲取特殊牌型台分以及槓牌機制改善的方法。實驗比較對象為原版 Seofon 與 Seofon_v2_{Honor}，每種對戰組合總共進行30場比賽，每場比賽進行384局，總共進行11520局。實驗結果分別如表四-2與表四-3所示。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2 _{offensive}	4382	1400	50.2%	875500	73.3%
Seofon	4022	1288	46%	-875500	26.7%

表四-2 進攻模組版本對上 Seofon 的實驗數據

表四-2顯示了進攻模組版本仍然比原版 Seofon 有更佳的胡牌率，

整個對戰組合大致上他家放槍數與自摸數與第一個字牌棄牌優先度版本差不多，但從最終獲得的分數可以看出，進攻模組版本在分數的獲取上相較於前一個字牌棄牌優先度版本有著更佳表現。

此外參考勝率的變化，勝場數從字牌棄牌優先度版本的18場提升到了進攻模組版本的22場，也顯示在此版本能夠在更多的場次獲取更多的分數，以確保勝利。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2 _{offensive}	4357	1365	49.6%	687500	63.3%
Seofon_v2 _{Honor}	4180	1259	47.2%	-687500	36.7%

表四-3 進攻模組版本對上字牌棄牌優先度版本的實驗數據

表四-3顯示了進攻模組版本在對上字牌棄牌優先度版本仍然有六成三左右的勝率。字牌棄牌優先度版本的胡牌率從原本的50.2%下降到了47.2%，而進攻模組版本仍然保持著將近五成左右的胡牌率，可以看出進攻模組版本有著較佳的湊牌效率。另外在分數的獲取上，參考這兩個版本與原版 Seofon 對戰的結果，正好顯示了進攻模組版本在分數的獲取上有著較佳的表現。

4.4 跟打機制版本

第三個版本為3.4節中提到的跟打機制版本，命名為 Seofon_v2_{follow}，由4.3節中的進攻模組版本，額外加入跟打機制。實驗比較對象為原版 Seofon 與 Seofon_v2_{Offensive}，每種對戰組合總共進行30場比賽，每場比賽進行384局，總共進行11520局。實驗結果分別如表四-4與表四-5所示。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2 _{follow}	4318	1369	49.4%	648000	56.7%
Seofon	4103	1292	46.8%	-648000	43.3%

表四-4 跟打機制版本對上 Seofon 的實驗數據

表四-4顯示了跟打機制版本雖然在對戰原版 Seofon 上仍然取得了56.7%的勝率，但在胡牌率上比起前兩個版本有小幅度的降低，在分數的部分，獲得的總分也不如前兩個版本。

推測可能的原因為跟打機制的加入，勢必需要放棄部分的湊牌與胡牌機會，也就是如果需要採取防守動作，一定會損失進攻機會。這部分的解決方法大致可以採取兩個部分，第一個方法是修正啟動跟打機制的時機，在進行防守動作時，若過早或過晚皆會導致效果不盡理想。太早啟動會放棄太多進攻機會，而臺灣麻將本身還是快速胡牌風格的遊戲，若太晚啟動則會導致防守動作失去作用，無法有效降低放槍率。第二個方法是針對跟打機制中的棄牌作出權重調

整，簡單來說，目前程式在進行跟打機制時，會挑選出前三張棄牌資訊，這部分未來應該要額外再加入更多棄牌選擇，提高跟打機制的效率，此部分詳細作法將會在第五章中作說明。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2follow	4200	1321	47.9%	-189000	43.3%
Seofon_v2Offensive	4219	1401	48.7%	189000	56.7%

表四-5 跟打機制版本對上進攻模組版本的實驗數據

表四-5顯示了跟打機制版本在對戰先前的進攻模組版本時，沒有辦法取得優勢。此結果可以進一步驗證先前幾個對戰組合的結論。首先跟打機制版本相比先前兩個改良版本在對戰原版 Seofon 時，胡牌率有下降的趨勢(50.2%→50.2%→49.4%)，也就是說跟打機制版本在湊牌與得分效率上確實不及字牌棄牌優先度版本與進攻模組版本，而表四-5實驗數據中顯示跟打機制版本強度確實不如進攻模組版本，剛好能夠驗證上述的結論。

在另一方面，雖然跟打機制版本在對戰進攻模組版本時沒有辦法取得優勢，但不管在胡牌率還是分數上，差距都沒有比原版 Seofon 對戰進攻模組版本時來的大(表四-2)，這部分也可以再次驗證上一個對戰組合的結論(表四-4)，跟打機制版本的程式強度仍然優於原版 Seofon。

4.5 比賽結果

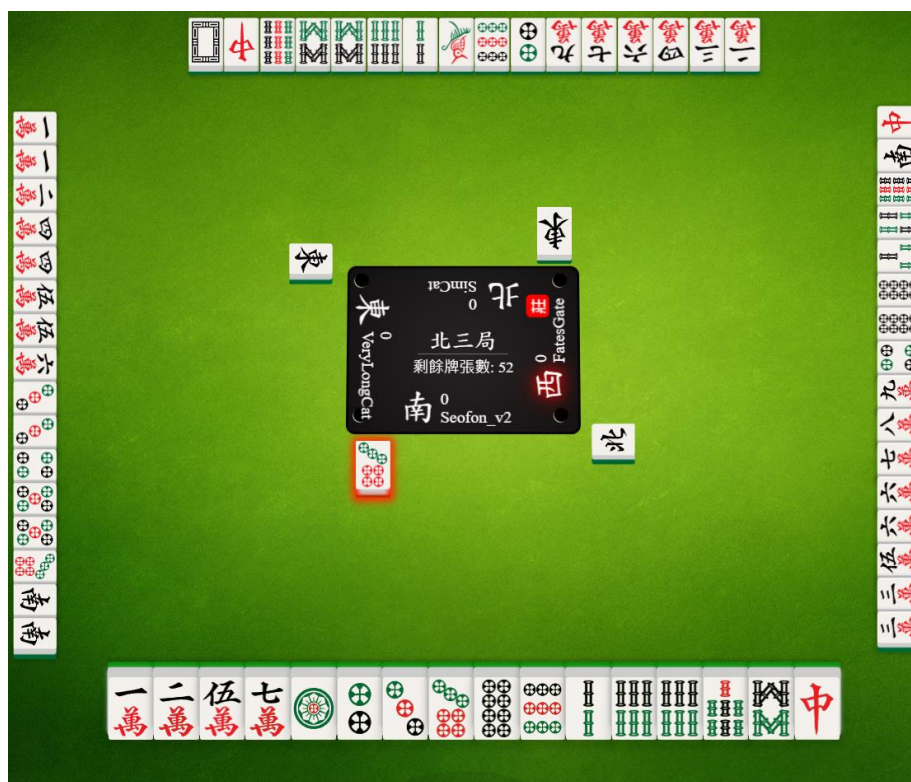
本研究所研發的程式 Seofon_v2 參與了 2022 年的 ICGA 電腦對局比賽。出賽時使用的程式版本為進攻模組版本，該次比賽共有五支程式參與，Seofon_v2 最後獲得了第四名的成績。

以下將進一步說明比賽數據與結果，接著進一步提出程式在比賽中表現不如預期的幾個原因。針對不同的問題進行討論與程式修正，最後再將修正問題後的版本與比賽時使用的進攻模組版本進行實驗比較。此外，最後會再將此修正後的版本與另一支程式 Zei 進行對戰，分析與其他程式進行對戰的結果。

程式	總胡牌數		胡牌率	排名
	他家放槍數	自摸數		
Seofon_v2	279	84	23.6%	4
VeryLongCat	309	102	26.7%	1
SimCat	313	105	27.2%	2
FateGate	319	87	26.4	3
Sparrow_v2	195	73	17.4	5

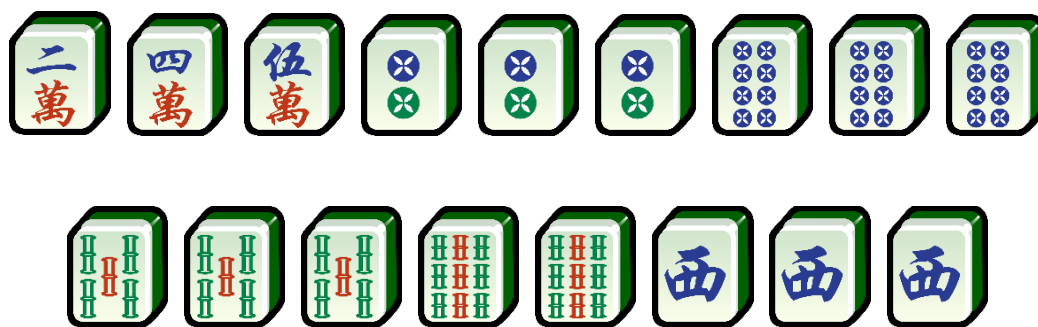
表四-6 2022年 ICGA 比賽結果

比賽數據如表四-6所示，比賽結果最終只獲得了第四名。可以觀察到 Seofon_v2 的湊牌速度明顯不如其他幾支較強的程式，主要的問題有：相依牌檢查問題、單張字牌的棄牌問題、聽孤張問題。以下將分別說明這些問題與進一步提出修正方法。



圖四-4 單張字牌的棄牌問題範例

從圖四-4中可以看到，程式在摸了七條後，最後選擇了七筒作為棄牌。此問題與先前在3.1節中所提到的方法有關，該方法在上圖的情況中，應該優先選擇紅中作為棄牌。此部分問題是因為在程式實做上的疏漏，並未在該情況中正確賦予額外權重，造成程式無法將紅中優先選擇為棄牌。程式在經由修正後已經能在此情況中正確選擇紅中作為棄牌。



圖四-5 聽孤張問題範例

以圖四-5為例，程式原先會選擇五萬作為棄牌，此選擇會造成程式只能聽三萬這張牌。原先程式只會將三萬與六萬的剩餘數量分別進行棄牌價值計算，但正確選擇應該將三萬與六萬兩張牌的剩餘數量一同計算。以上圖為例子，此處假設三萬、六萬剩餘張數皆為三張。程式經由修正後，在計算二萬的棄牌價值時，會得到三萬的三張，而在計算五萬的棄牌價值時，會得到三萬+六萬的六張。最後得到五萬的成組機率較高，最後正確地選擇出二萬作為棄牌。

將上述的問題修正後，將修正後的程式與比賽使用的進攻模組版本進行實驗。因為在比賽中遇到的問題主要為湊牌效率不佳，因此在實驗數據的呈現上將只會針對胡牌率進行比較。確認程式的湊牌效率經由修正後是否有提升。

程式	總胡牌數		胡牌率
	他家放槍數	自摸數	
Seofon_v2 _{fix}	4237	1362	48.6%
Seofon_v2 _{Offensive}	4148	1345	47.6%

表四-7 修正版本對上進攻模組版本的實驗數據

由表四-7中的實驗數據可以得知，修正問題後的版本在胡牌率在1%的提升，表示先前所提到的問題對於 Seofon_v2的湊牌效率有一定程度的影響。

此外再將修正版本與另一支程式 Zei 進行實驗，確認 Seofon_v2在對戰除了 Seofon 以外的程式是否也能取得一定程度的優勢。從表四-8的實驗數據中可以得知，Seofon_v2在對戰 Zei 時，在胡牌率、分數與勝率皆有較佳的表現。

程式	總胡牌數		胡牌率	分數	勝率
	他家放槍數	自摸數			
Seofon_v2 _{fix}	4244	1388	48.9%	737500	73.3%
Zei	4206	1319	47.9%	-737500	26.7%

表四-8 修正版本對上 Zei 的實驗數據

五、結論與未來展望

如同前面章節中所述，Seofon_v2是一支由Seofon改良而來的程式，但從實驗結果中觀察與分析可以得知Seofon_v2仍然存在一些缺點能夠改進。此章將會討論這些缺點，並且針對這些缺點提出未來在程式修改上可行的做法，以提供後續研究的方向與目標。

首先不管是先前的程式Zei、Seofon，還是本研究中的Seofon_v2，整體上都是採用規則導向為主要架構，也因此有許多實做的技術與細節上需要更加小心謹慎。如同Seofon所面臨的問題，Seofon_v2在改良Seofon後，仍然產生在實作上忽略了部分規則的缺點，而這也是規則導向為主的程式所要面臨的問題。舉例來說，像是Seofon在聽牌時仍然會選擇明槓的問題。

本研究改良後的程式在胡牌率、自摸率與得分上都有一定程度的提升，但如同前面章節中所述，這些影響勝負關鍵的因素會隨著不同策略的使用而有所改變。若要在後續研究中繼續加入不同的策略來改良程式，也會進一步增加影響湊牌效率與得分效率的變因，此部分也是我認為在後續研究中需要特別注重的部分。

除了上述提到加入不同策略的情形外，以下將針對本研究中所使用的策略，提出改進的方向。首先是強化獲取特殊牌型的策略，在3.2節中有提到，此方法的主要目的是確保不要錯過做大牌的機會，但若要做大牌，勢必會損失部分快速胡牌的機會。因此在這個部分，如何在分數與快速胡牌間做出合適的選擇，便是改良此策略

的主要方向，目前在實做該策略上採用規則導向與估值函式並行的方式。

規則導向方面，在啟動條件的判斷上能夠進一步透過實驗得知將參數值設定為多少會取得更好的效果。估值函式方面，應該能以模擬胡牌牌型的方式來計算估計值。但此方法在實作上會有許多需要特別考慮的因素，例如距離胡牌牌型所需要的張數，會影響估計的準確度，因為距離胡牌所需要的張數越多，其影響因素也越多。此外，由於程式會記錄棄牌資訊來選擇棄牌，因此在模擬胡牌牌型時也會因為其他對手的棄牌而有所影響。在計算估計值時可能會因為對手的棄牌而有很大的落差，導致程式在選擇策略上可能出現不穩定的情況，這也是這部分在實作上需要特別考慮的部分。

再來是跟打機制的部分，如同實驗結果所示，跟打機制版本的加入並沒有辦法有效增加程式的強度。但跟打機制確實是能有效降低放槍率的策略之一，因此如何改良這部分也是改進方向與目標之一，若要改良此策略，我認為可以針對下面兩個部分。

首先是啟動跟打機制的時機，過早或過晚啟動跟打機制都會導致防守效率不佳，甚至會影響湊牌效率，進而導致進攻效率不佳，因此後續可以針對不同的啟動時機進行實驗，從中找出最適合的啟動時機。

接著是跟打機制中棄牌的選擇，目前加入的棄牌選擇數量較少，導致跟打機制可能無法有效執行。因此可以增加棄牌的選擇數量，但此部分需要注意的是，需要針對每張棄牌給予不同的權重，因為某張棄牌被打出時可能距離當前回合已經非常遠，在當前回合中或

許該張棄牌已經不是安全牌，因此針對距離當前回合更近的牌應該給予更高的優先度。另外在跟打機制中時機的選擇，應該將聽牌狀態給予更嚴謹的判斷條件，未必聽牌時就一定放棄跟打動作。應該進一步考量聽牌時的胡牌機率，若聽牌時的胡牌機率很低，此時進行跟打動作應該是較好的選擇。胡牌機率的高低應該能夠經由進一步的實驗進行佐證，達到更準確的跟打機制啟動時機。

還有像是預測對手手牌的策略，人類高手玩家能夠依靠棄牌與場上資訊來推測其他對手的手牌。因此，若能透過程式目前已經使用的棄牌資訊策略與場上已知的資訊，來預測其他對手的手牌與對手的聽牌狀況，便能夠更有效降低放槍率，也能夠通過計算得知對手的手牌中哪張牌被打出的機率最大，進而增加胡牌效率。

此外，在本論文中的實驗數據中，只有單純參考胡牌率、分數與勝率的高低與變化，尚未有辦法直接驗證演算法能有效的解決原先程式所碰到的問題，例如分數的增加是否真的是因為獲得了額外的台數。此部分在未來進行相關研究時，也能夠進一步在實驗時加以考量，例如使用 log 檔紀錄胡牌時的牌型判斷。或是能夠統計胡牌時所帶來的正面收益與負面收益，進而判斷出演算法是否真的正確並有效地改良程式。

在進行實驗時，也可以挑選不同的程式進行實驗。除了能夠增加額外的比較對象，讓數據更有參考價值外。也能夠避免若程式版本太過相近，可能會產生實驗結果不錯但比賽結果不如預期的現象。

參考文獻

- [1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Mastering the Game of Go with Deep Neural Networks and Tree Search, *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. V. D. Driessche, T. Graepel, and D. Hassabis, Mastering the Game of Go without Human Knowledge, *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan and D. Hassabis, Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, arXiv:1712.01815v1.
- [4] S. Nair, Alpha Zero General, <https://github.com/suragnair/alpha-zero-general>.
- [5] J. Li, S. Koyamada, Suphx: Mastering Mahjong with Deep Reinforcement Learning, arXiv:2003.13590, 2020.
- [6] Cycgame 麻將平台, <https://cycgame.com/cyc/>。

[7] S. Li and X. Yan, Let's Play Mahjong!, Computing Research Repository(CoRR abs/1903.03294), 2019.

[8] 林猷琛，利用他家資訊模組來改良麻將程式，2020，國立台灣師範大學資工所碩士論文。

[9] 林宗翰，利用棄牌資訊強化策略改良麻將程式，2021，國立台灣師範大學資工所碩士論文。

[10] 陳新颺，電腦麻將程式 ThousandWind 的設計與實作，2013，國立台灣師範大學資工所碩士論文。

[11] 吳俊緯，電腦麻將程式 MahJongDaXia 的設計與實作，2015，國立台灣師範大學資工所碩士論文。

[12] 莊立楷，麻將人工智慧之研究，2015，國立交通大學資訊科學與工程研究所碩士論文。