

國立台北大學資訊工程學系專題報告

電腦麻將防放槍機制之研究

專題組員：陳昱堂、王昱鈞、李家緯、姜承佐

專題編號：PRJ-NTPUCSIE-110-001

執行期間：2021 年 09 月 至 2022 年 06 月

1. 摘要

近年來 AI 技術不斷地攀升，在模擬這方面也有相當的成果，電腦棋局自然不用說。圍棋、將棋、暗棋、黑白棋、麻將等，都已有許多人在著手進行研究。而我們以台灣 16 張麻將為對象，針對場上的公開情報和其他家的棄牌狀況來製作麻將程式的防禦模組。

2. 簡介

麻將為一種資訊不完全的遊戲，除了自身手牌外無法得知其他對手的牌型，也潛在許多不確定性，因此難以模擬其他玩家的行動。所有對自己不利的牌皆有可能是其他玩家勝利的元素。例如：自己打出的牌皆有可能被對手湊成牌組而進一步達成胡牌，為避免自身放槍，因此研究從進攻轉換成防守的時機，而防守中利用其他家的棄牌資訊及牌桌資訊，進而猜測他可能需要與不需要的牌型，觀察牌型的危險度。

3. 專題進行方式

為了製作出防放槍模組，我們研究了許多麻將打牌的策略及相關資料並統計各樣台桌數據從而得出算法及流程。

3.1 名詞解釋

(1)熱度(heat)，我們的模組是以熱度做為計算的基礎，並以其他參數來加權調整熱度值，以此決定在棄牌時該捨棄哪張牌，所謂熱度表示某張牌還剩下多少種組成順子的可能性



Fig.1. 熱度計算示意圖

3 萬可以組成 1 到 3 萬，2 到 4 萬 3 到 5 萬這三種順子，而每一組中的 3 張牌都各有 4 張，所以 3 萬一開始的熱度是 4 乘以 4 乘以 4 乘以 3，也就是 192，熱度越高表示此牌越危險。

(2)時間(t)，除了熱度我們還會使用到 t(時間)作為參數，t 用來紀錄某一張牌距離上次被打出後過了多少時間，t 越大表示越危險，且除了自己的 3 個玩家對於每一張牌都有自己獨立的 t 值。

$t_{x,i}$: 時間, $x \in \{\text{其他家}\}$, $i \in \{\text{所有牌種}\}$
"p 家在上次略過 c 牌後摸過 $t_{p,c}$ 張牌"

(3)剩餘張數(剩張)，記錄每一種牌還剩多少張有可能出現在場上，舉例來說，如果場上已經出現一張八萬，手

中有一張八萬，那此時八萬的剩餘張數為： $4-1-1=2$ (張)

3.2 流程及算法

以下為我們的棄牌流程

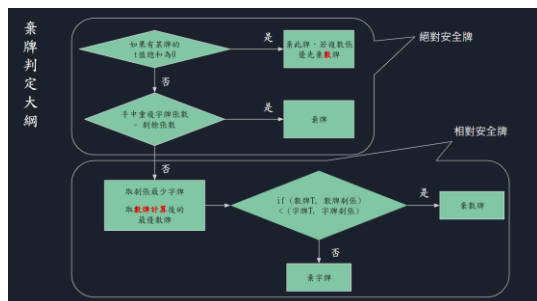


Fig.2.棄牌流程示意圖

首先尋找是否有牌的 t 值為零，如果有的話我們就直接跟打，如果符合條件的牌不只一種，我們優先捨棄數牌。因為一般來說數牌比字牌更危險，接受我們檢查手中是否有剩餘張數等於我們手中持有數量的字牌舉例來說就是假如場上已經有兩張東風，而我們手中也持有兩張東風的情況下，東風就是絕對安全的牌。如果也沒有的話我們接著比較剩餘張數最少的字牌和數牌的熱度值，然後打出熱度值最小的牌。

3.2.1 筋牌修正

筋牌是麻將中相差為 3 的一系列數牌，像 147 萬算一條筋，258 筒也是一條筋，我們可以透過筋牌來排除對手持有特定組合的可能。舉例來說，如果某人丟出 4 萬，那此時 1 萬和 7 會相對安全，因為如果此人手裡有 234 萬這個順子，因為已經成為一個完成組了，這時候把 4 萬丟出來是不合邏輯的。所以我們可以認定當某人丟出 4 萬時，此人手裡一定不會有 2 萬 3 萬這個組合，如果沒有 2 萬 3 萬，那此時 1

萬就是安全的，對 7 萬也是同樣的道理。



Fig.3. 筋牌關係示意圖

3.2.2 時間計算

下圖是我們在時間判斷的流程圖

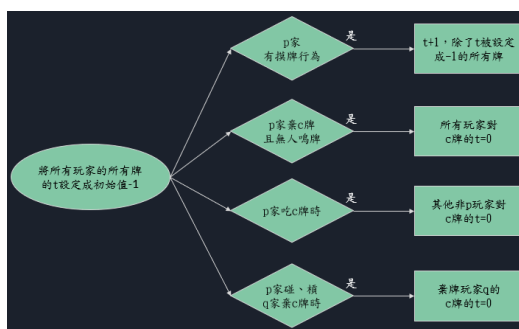


Fig.4. 時間計算示意圖

前面提到的時間計算，想法是透過其他玩家丟出的牌來判斷某張牌是否被需要。首先在開始時我們把所有玩家的所有牌的 t 設定成 -1，表示他們從來沒有被打出來過，接著我們會根據場上玩家的行為來調整 t 有玩家摸牌時，此玩家除了被設定成 -1 的牌 t 值全部加一，當有玩家捨棄某張牌且其他玩家沒有鳴牌時，我們把所有玩家對 C 牌的 t 值都設為 0，表示此時無人需要此牌。其他判定也是類似的概念，當 P 玩家吃了 C 牌時，我們把其他玩家對 C 牌的 T 值設為 0，最後當被玩家碰或槓 Q 玩家丟棄的 C 牌時，只有棄牌玩家 Q 對 C 牌的 t 值設為零，因為這種情況只能保證丟牌的玩家 Q 不需要這張牌。

3.2.3 數牌計算

下圖是我們決定丟棄數牌時的流程圖

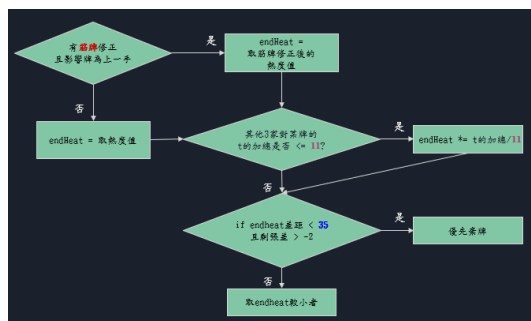


Fig.5. 數牌計算示意圖

首先檢查牌是否有筋牌修正的影響，如果有的話將熱度扣掉對應的數字，接著檢查其他三家對牌的 t 值加總是否低於我們設定的閾值，若低於閾值的話以一定比例減少此牌的熱度值，用以實現 t 值小時的熱度修正，此閾值為多次實驗後所得，如下圖

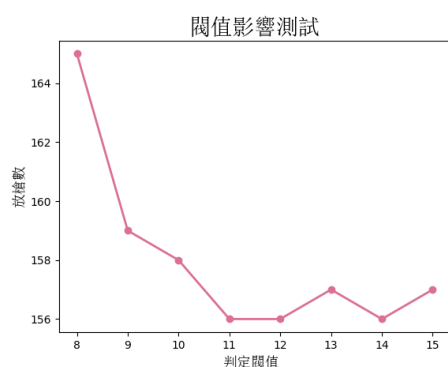


Fig.6.熱度大小閾值示意圖

接著找出熱度最小的牌，然後檢查是否有牌的熱度和此牌差距小於閾值，且剩餘張數比此牌還要少 2 張以上，如果有，則此牌將取派熱度最低的牌，成為即將丟棄的牌，這是因為熱度只考慮順子(123)沒有考慮到刻子(456)，此檢查的閾值同樣通過實驗得出，如下圖

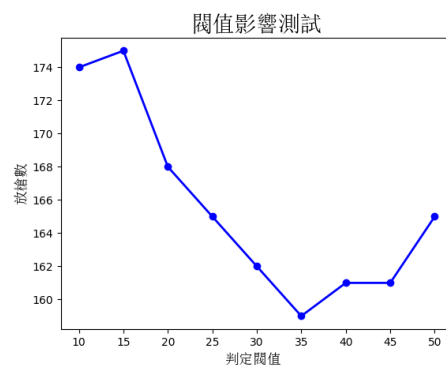


Fig.7.熱度差距閾值示意圖

3.2.4 啟動時機

我們藉由統計歷年比賽紀錄的資訊，分析得出適合防放槍啟動時機的區間

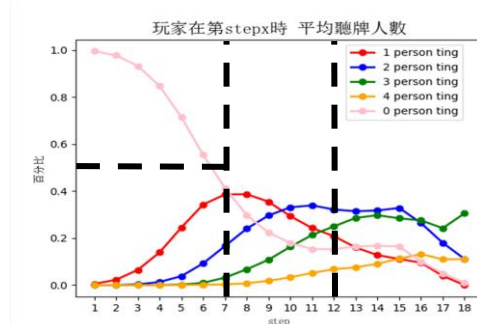


Fig.8.熱度差距閾值示意圖

統計次數共 13824 次，先經由統計各個 step 當中各玩家的 0 家聽到 4 家聽總次數統計，再把各個 step 中的 0~4 家聽分別做加總再平均。ex: step4 裡面 player1，其 1 家聽次數為 478 次，player2 為 469 次，player3 為 485 次，player4 為 474 次，4 個玩家做加總，並平均後結果為:476.5 次。

由上述圖可注意到，第一條虛線：無人聽牌情況開始少於 50% 第二條虛線：1 人聽牌情況開始少於 2、3 人聽牌。由此推測啟動防放槍的時機在 step7~step12 最為合適。

3.3 驗證方法

我們使用 unit test 來檢查防放槍程式的效果，我們把防禦模組作為

function 導入 unit test 的測試模組中然後輸入以前比賽的歷史盤面後把放槍的 AI 改成防放槍程式，接著檢查是否依然放槍，左邊這一張圖就是以前比賽的歷史紀錄，右邊這張圖是對應的答案。



Fig.9. Unit Test 示意圖

我們實際測試了 1208 場，只有 159 場仍然放槍，比率只占 13.2%



Fig.10. Unit Test 結果示意圖

4. 主要成果與評估

4.1 校外實戰結果

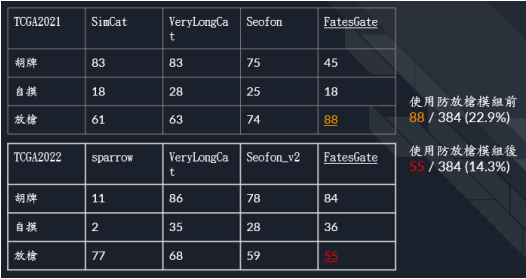
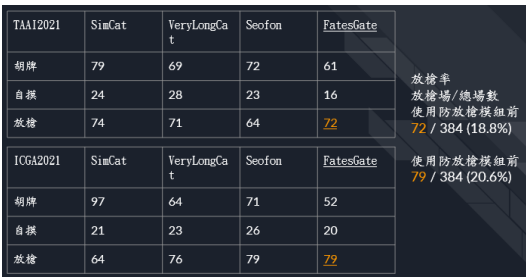


Fig.11、12. 校外實戰結果示意圖

在專題製作後期時，我們結合了實驗室的進攻型 AI 去參加校外比賽。由上述圖可觀察出，沒有結合防放槍模組的 AI 為前三個，只有最後一個 AI 有結合防放槍模組，而從數據可得知明顯地有結合防放槍模組的 AI，放槍率比以往沒結合的 AI 還要低一些。這代表我們所做的防放槍模組是有一定功效的，但同時也暗示著仍然還有很大的進步空間。

4.2 評估

此次研究開發出了一個防放槍模組，利用當下牌桌上的資訊及他家棄牌的資訊，來當作媒介進行時間計算、剩餘張數、筋牌修正、數牌計算、熱度計算判定，得出一個閾值來得知此回合哪張手牌棄出最不容易放槍。藉此避免不必要的放槍，再結合進攻的麻將 AI，達到可攻可守的麻將 AI。

對於麻將的電腦對局來說此研究算是一個新穎的領域，以往的麻將 AI 都是比較誰先胡牌就能快速得分快速刷場，而這個研究則是以保守的方式去打牌，避免自己放槍給予自己莫大的虧損，但是啟動防守的時機也是一大難題。

未來希望能夠將各個判定間的關係與閾值結合的更精確，以達到更高確率的防放槍能力。

5. 結語與展望

雖然我們的防禦模組已經完成，但仍然有許多不足之處，像是數據量不夠多，以及演算法內部可再強化等。因為內部閾值參數的調動需要再多做更多的配對，藉由更多數據去測試出最佳值。希望之後如果條件允許的話希望能把它做得更加完善，若能夠獲得更大量的數據的話甚至可以嘗試以深度學習的方式去實現。

6. 銘謝

在此感謝指導教授在開會時給予許多各種方面的看法，不管是在製作過程中的想法以及最終評分上的簡報製作，都修正了我們很多小錯誤，讓整體作品更加完善及周到。這邊也感謝實驗室的學長也在製作的過程中給予許多自身的看法以及研究方向，好讓我們得以順利進行。再來最終也感謝組員們在時間上都願意撥出許多時間互相幫忙查看對方每周的進度內容。

7. 參考文獻

- [1] 林宗翰, 2021, 〈利用棄牌資訊強化策略改良麻將程式〉, 國立臺灣師範大學資訊工程系碩士論文
- [2] 林世和, 〈電腦麻將溝程式 MJTalk 的設計與實作及電腦麻將程式 MahJongDaXia3 的演算法探討〉, 國立臺灣師範大學資訊工程系碩士論文
- [3] Yuan Cheng, Chi-Kwong Li, Sharon H. Li, Mathematical aspect of the combinatorial game "Mahjong". arXiv:1707.07345v2 [math.HO] 23 Jan 2019.取自 <https://doi.org/10.48550/arXiv.1707.07345>

