

國立台北大學資訊工程學系專題報告

三國翻轉棋

A multi-player, deterministic board game

專題組員：沈彥廷、莊順安、陳昭宏、徐豪聯

專題編號：PRJ-NTPUCSIE-111-002

執行期間：2022年9月 至 2023年5月

1. 摘要

世界上有各式各樣的棋類遊戲，其中比較熱門的那些大多是兩人對弈的模式，而我們希望可以參考這些熱門的棋類遊戲，不削減玩法樂趣的同時還能夠讓更多的玩家參與其中，所以我們從黑白棋(Othello)的遊戲概念出發，設計出了三人對弈的三國翻轉棋。

2. 簡介

(1) 專題研製背景

在確定題目之前我們考慮了很多種選項，原本我們是打算直接針對某一種發展已經成熟的棋去設計電腦對局的程式，但是之後我們還是決定想要自己設計一款新的棋類遊戲看看，從規則開始一直到電腦對局程式的設計都由自己來做，也算是一個很有趣的過程。

(2) 專題研究目標

設計出公平且有趣的棋類遊戲，並設計電腦對局程式，透過觀察電腦間對戰的棋局以及實際與電腦對戰來盡可能提升對局程式的棋力。

(3) 主要預期效益

遊戲規則有趣易上手、對三方玩家公平、對局程式之棋力達到一定程

度並提供GUI讓玩家與玩家或是與電腦之間進行對戰。

3. 專題進行方式

(1) 三國翻轉棋規則介紹

此翻轉棋為正六邊形棋盤，每邊有6格，全棋盤共91格。玩家有紅、綠、藍三方，下子順序固定為紅方→綠方→藍方，依序輪流落子。

初始盤面：固定中央棋格F6為綠方棋，周遭6格為紅、綠、藍三方各佔兩子。

合法走步：下棋時可與其他顏色的棋子連續連成三個棋子以上所構成的橫線、正斜線或是反斜線，且除了連線兩端的棋子外皆為非己方的同顏色棋子，此時會將非連線兩端的棋子轉變為我方棋子。

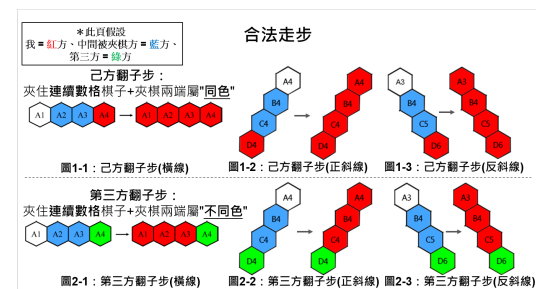


Fig.1 合法走步

獲勝方式：任一方皆無法下合法走步時即遊戲結束。在終局盤面計算紅、綠、藍三方的佔子數，但因為先手容易被後手翻子(後手佔有優勢)，

會將計算結果進行適當貼目。

貼目規則如下：

紅方貼0目：紅方佔子數不變。

綠方貼1目：綠方佔子數 - 1。

藍方貼2目：藍方佔子數 - 2。

(貼X目 = 佔子數 - X)

若貼目後佔子數同則先手方獲勝(紅

→ 綠 → 藍)。

(2) 電腦對局程式實作

(2.1) 電腦回合流程

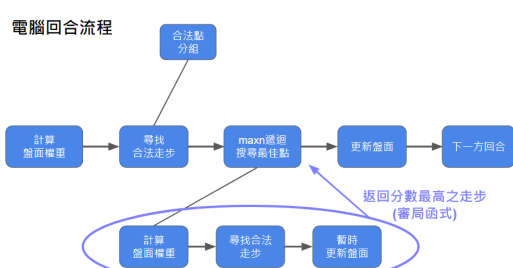


Fig.2 電腦回合流程

上圖為每一回合中電腦會做的事情，依序是計算盤面權重、從每個空格出發往六個方向去尋找及判斷該空格是否為合法走步，其中包括合法點分組、Maxn遞迴搜尋最佳走步(計算盤面權重、尋找合法走步、暫時更新盤面)、更新盤面並輪到下一方行動，使用到的演算法及判斷都會在後面詳細地介紹。

(2.2) Maxn遞迴搜尋演算法

maxn的主要目的是在當前回合中找到對自己最有利的走步，方法是透過預測數回合後的盤面變化並計算、比較各家分數來進行選擇。

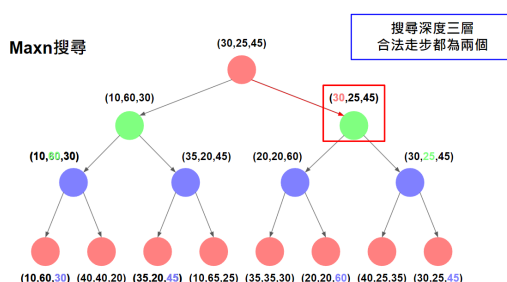


Fig.3 Maxn搜尋範例

若搜尋節點達成終止條件，須以葉節點之盤面進行審局，並給予三位玩家審局分數。每次遞迴將搜尋當前節點之所有子節點，如上圖所示。圖中每個節點有三個分量，第 i 個分量代表玩家 i 對於此節點之評分，其中 $i = 1, 2, 3$ 。當輪到第 i 個玩家要進行走步時，僅從當前節點下之子節點比較第 i 個分量，並選擇第 i 個分量之最大值。

(2.3) Shallow剪枝

剪枝的主要目的是不影響電腦表現的同時還可以增加搜尋效率，方法是透過設定一上限分數(Bound)使得分數可預測，藉此減去不必要的節點計算。

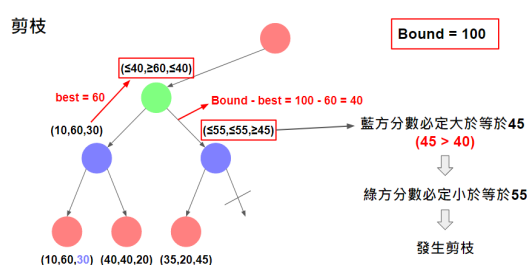


Fig.4 Shallow剪枝範例

設定一分數上限Bound=100，利用審局函式計算出三方的分數後把這一百分照原分數比例分配給三方，使得三方分數總和固定，這樣就能夠進行分數的預測。

當分數由子節點回傳時，會多一變數best來記錄當前所有子節點中的最佳分數，第一個子節點回傳分數後紀錄best後就會進入第二個子節點的搜尋，同時把Bound - best傳下去作為該子節點的Bound，若此時該子節點對應方之分數大於Bound的話則發生剪枝，若小於Bound則接著進入第三個子節點的搜尋。

(2.4) 審局函式

審局函式的功能是計算當前盤面三方之分數，我們審局函式總共使用了三個表來進行分數的計算。

(2.4.1) 路徑表

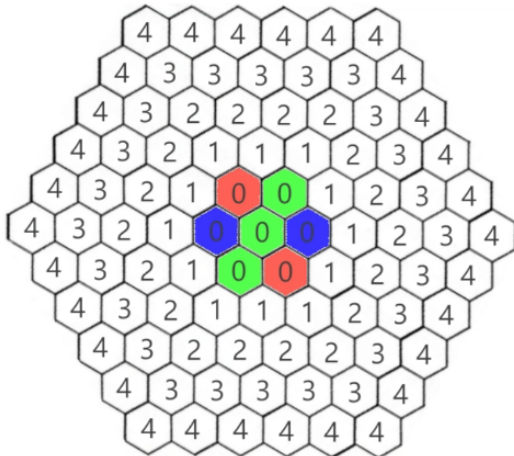


Fig.5 路徑表(Path Table)

路徑表的值代表各格與最近的被佔領格的最短直線距離，數值大表示該空格離戰場越遠，影響比較小；數值小則表示下該點會直接影響到當前戰況，因此路徑表數值越小越好。

(2.4.2) 翻轉次數表

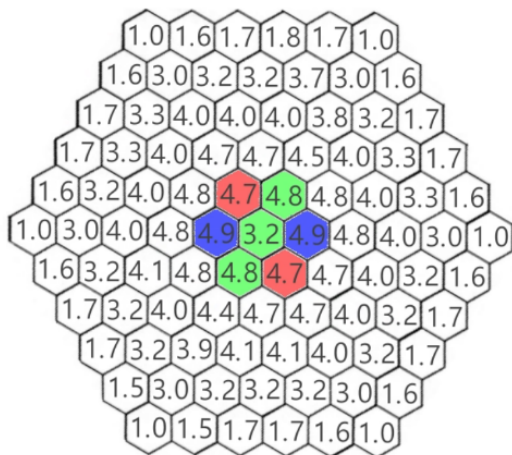


Fig.6 翻轉次數表(Flip Table)

翻轉次數表的值代表一整場遊戲中每格的平均翻轉次數，每格的平均翻轉次數代表該格的穩定性，數值高的點代表容易被翻轉；數值小的點則代表該格相對穩定，像是角落這個點，因為下了就不會被翻因此翻轉次數表的值為1.0。翻轉次數表的數值同樣是越小越好。

(2.4.3) 權重表

權重表主要是透過給各個位置不同的數值來判斷當前盤面中各點的重要性，我們主要分為高、低及基本權重，使用10:0.1:1的比例。首先先介紹一下高權重的概念，如果佔領了某個點就不會再被翻且有助於擴張、防守盤面的話就會設為高權重；而如果佔領了該點會把穩定點讓給另外兩方或是會幫助另外兩方擴張盤面的話則為低權重。另外，對三方來說重要的點與不好的點不完全一樣，因此三方會使用不同的權重表。

我們對高低權重的判斷總共有六項：

(一) 未被佔領之角落高權重，此角落會隨著盤面更新做延伸，若鄰近角落為己方的話則會出現新的高權重點，稱為角落延伸點。

(二) 角落相鄰點低權重。

(三) 某邊唯一空位高權重。

(四) 邊的中間兩點若佔領該點會導致【讓出角落及邊之佔領權】則設為低權重。

(五) 不位於邊上但為穩定的點設為高權重。

(六) 中央7格高權重。

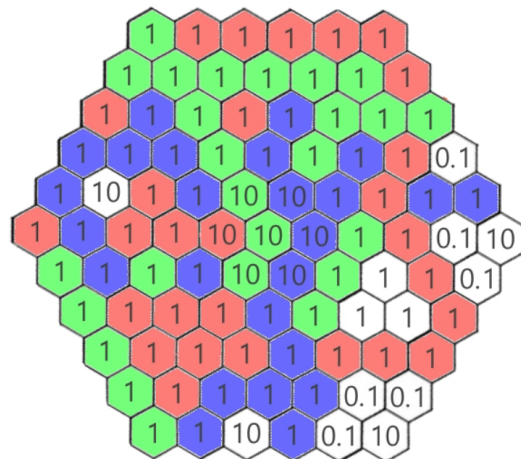


Fig.7 權重表(綠方)

基於以上六項對權重的判斷就可以建立出如上圖所示的權重表。

(2.4.4) 審局函式總結

$$1. F(P) = \sum_{i=0}^{90} \frac{\text{WeightTable}[i]}{\text{PathTable}[i] + \text{FlipTable}[i]}$$
$$2. F(P) = \frac{F(P) \times \text{Bound}}{F(P1) + F(P2) + F(P3)} \quad (\text{Bound} = 100)$$

Fig.8 審局函式

建立好三個表後就可以定義審局函式F(P), P表示某方, 共有123三種值分別代表紅綠藍方、i有值0~90代表91個棋格, 如果第i格為某方佔領的話就把分數加至那一方, 如果是空格就會跳過, 路徑表與翻轉次數表都是越小越好, 因此把它們放在分母的位置; 權重值越大越好, 因此放在分子位置。把91個點計算完後就會得到三方的分數, 然後進入式子2, 把100分分配給三位玩家, 這樣就完成了一次分數的計算。

(2.5) 合法點分組

若Maxn使用六層進行搜尋的話實驗中有出現過需要搜尋二十分鐘以上的回合存在, 因此我們想利用將合法點進行分組的策略來減少搜尋時間。

我們把合法點分成四組, 其中第二組有三項判斷, 其餘則為一種。另外, 分組順序是以1、4、2、3的順序進行分組, 這是因為Pool 1與4 是 利用權重表進行分組, 而權重表已經建立好了, 因此可以直接使用:

(Pool 1) 該合法點為高權重。

(Pool 2) 該合法點能夠在兩個方向以上形成翻子 或是 夾棋兩端屬於同方的子 或是 位於邊邊且能夠翻三子以上。

(Pool 3) 剩餘的所有合法點。

(Pool 4) 該合法點為低權重。

分完組後就可以進行搜尋, 搜尋的順序就是按照1、2、3、4的順序, 從好的點到差的點, 如果在前面的Pool

中找到合法點了就不會再搜尋下一個Pool了, 我們就是利用這樣的搜尋策略來減少搜尋的時間。

(3) 遊戲數據儲存格式

參考西洋棋的PGN棋局記錄格式, 以文字檔紀錄遊戲的比賽名稱、地點、日期、局數、玩家名稱、勝場、起始盤面以及棋步。

```
[Event "first play"]
[Site "home"]
[Date "2023.5.26"]
[Round "1"]
[Red "Andy"]
[Green "Brian"]
[Blue "Cindy"]
[Result "1-0-0"]
000000/000000/00000000/00000000/0000120000/000032300
00/0000210000/000000000/00000000/00000000/000000
1.E04 2.D03 3.E03 4.G04 5.E02 6.D05 7.F08 8.E07 9.D07
10.C05 11.F04 12.D04 13.D02 14.H03 15.H05 16.E08
17.F03 18.F02 19.G02 20.C07 21.G03 22.C06 23.H02
24.E09 25.D06 26.D08 27.F09 28.F01 29.D01 30.D09
31.C04 32.C03 33.C02 34.C01 35.B02 36.J01 37.H06
38.G08 39.B04 40.G07 41.J04 42.H04 43.E10 44.F11
45.B07 46.H07 47.B05 48.A02 49.K04 50.A03 51.J06
52.E01 53.K03 54.A04 55.B03 56.B01 57.F10 58.H01
59.H08 60.L04 61.J05 62.G01 63.H09 64.J07 65.K05
66.J03 67.G09 68.K02 69.A01 70.J02 71.K06 72.J08
73.L05 74.A05 75.B06 76.A06 77.L03 78.K07 79.C08
80.L06 81.L02 82.L01 83.G10 84.K01
```

Fig.9 PGN格式檔案

(4) GUI介面開發

使用matlab design app製作的GUI, 初始介面包含開啟舊檔功能與開始遊戲的功能, 玩家可以選擇人數和電腦難易度, 分別是隨機走步電腦的easy難度、三層搜尋電腦的正常normal難度以及六層搜尋電腦的hard難度。

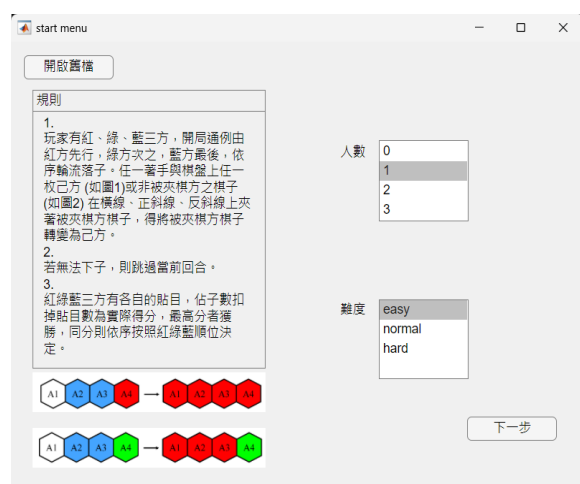


Fig.10 遊戲開始介面

若是至少有一位玩家, 我們會讓玩家自行輸入比賽資訊包含比賽名

稱、地點、名稱等，其中比賽名稱和地點可為空白。

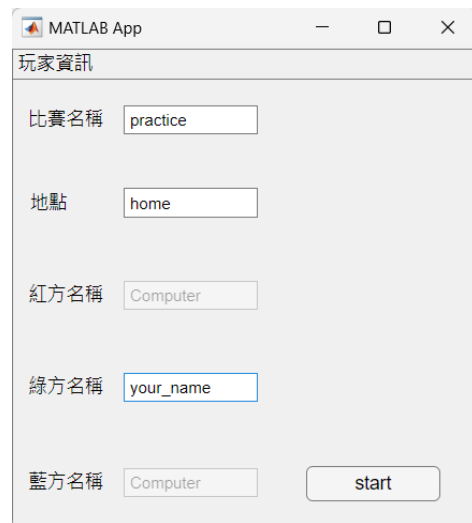


Fig.11 輸入資訊介面

之後按下start後即可開始遊戲，介面會顯示三方名稱、記分板、當前回合和行動方，若是玩家回合，則會列出合法走步供玩家選擇，並在右圖的盤面上標示為黃色外框，同時也可使用下拉式選單跳至選擇的回合盤面，或是使用上一步查看。



Fig.12 遊戲中之操作介面

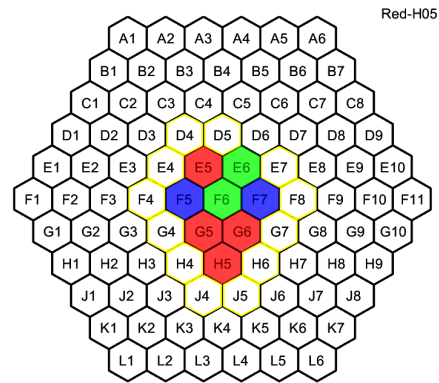


Fig.13 遊戲中之棋盤

比賽結束後也可以再次開啟app，進入到遊戲開始介面，點選左上角之開啟舊檔功能進行復盤，只要選擇我們在遊玩時生成的pgn_output.txt檔案，就可以開始。

4. 主要成果與評估

(1) 遊戲公平性

若無貼目的話三家的勝率比率大約會是 **27% : 31% : 42%**，可以看到藍方的後手優勢可以說是非常明顯，因此我們讓各強度電腦之間進行了上千場對戰，而我們也認為原本就佔優勢的藍方在經過貼目後仍然應保有一些優勢，因此綜合考量後最後我們選擇了**0:1:2**的貼目數量，此時的三方勝率約為 **32.5% : 32% : 35.5%**

(2) 電腦對局程式之成果

因為三國翻轉棋為一新棋類遊戲，因此我們的成果主要會放在對局程式的進步上。

三國翻轉棋為三人遊戲，若是三方實力相當時勝率應該各為33%上下，我們讓僅使用路徑表及翻轉次數表的三層電腦對戰兩個隨機走步電腦，藉此來觀察路徑表與翻轉次數表的效果，而效果很不錯，勝率可以達到接近95%，而我們再讓使用權重表的三層電腦與兩個沒有權重表的三層電腦對

戰來觀察權重表的效果，效果同樣明顯，勝率達到63%左右。

三層(無權重) vs. 兩隨機走步電腦	勝率	三層(有權重) vs. 兩個三層(無權重)	勝率
紅方三層	94.6%	紅方有權重	63%
綠方三層	94.8%	綠方有權重	62%
藍方三層	93.5%	藍方有權重	64%

Fig.14 無權重及有權重三層電腦對戰結果

最後我們再讓做完合法點分組的六層電腦分別與使用權重表及不使用權重表的三層電腦進行對戰，觀察合法點分組的效果，勝率分別來到了48%及78%左右。

六層(分組+權重) vs. 兩個三層(有權重)	勝率	六層(分組+權重) vs. 兩個三層(無權重)	勝率
紅方六層	48%	紅方六層	78%
綠方六層	47%	綠方六層	76%
藍方六層	48%	藍方六層	78%

Fig.15 六層電腦對戰結果

5. 結語與展望

雖然三國翻轉棋的整體設計大致上已經完成，不過在公平性上、尤其是電腦對局程式的表現都還可以進一步提升，因為是一種新設計的棋，因此對局程式在穩定性上可能不太足夠，如果未來有機會遇到使用不同對策的玩家或是其他對局程式的話很難保證我們目前的對局程式還能夠保持良好的表現。我們所使用的演算法及各種判斷也許還可以再進一步強化，而如果可以獲得更多值得參考的數據的話就可以如AlphaGo那樣，用深度學習去實現。

6. 銘謝

謝謝指導教授給予我們的建議及幫助，不管是在專題的進行方向、實作過程上還是最後簡報製作、成果發表的地方都給予我們很大的幫助。另外也要感謝實驗室的學長協助我們進行專題的製作。

7. 參考文獻

- [1]Luckhardt, C.A. (1986) An Algorithmic Solution of N-Person Game
[2]Sturtevant, N.R. (2000) On Pruning Techniques for Multi-Player Games

