

# 第十八屆旺宏科學獎

## 成果報告書

參賽編號：SA18-065

作品名稱：2048 遊戲的勝率與時間差學

習法程式特徵之分析

姓名：樂峻佑

關鍵字：2048、TD Learning、Pattern

# 2048遊戲的勝率與時間差學習法程式特徵之分析

## -目錄-

摘要.....	02
Abstract.....	03
壹、 研究動機.....	04
貳、 研究目的.....	04
參、 研究設備及器材.....	05
肆、 研究架構.....	06
伍、 研究對象介紹.....	06
陸、 研究過程與探討.....	08
實驗一：特徵分析與初步測試.....	08
實驗二：單一特徵的執行與數據分析.....	11
實驗三：多組特徵的執行與學習效能分析.....	14
實驗四：機動性質的創新、整合與推廣.....	18
柒、 研究結果.....	21
捌、 討論.....	23
玖、 未來展望.....	24
拾、 研究進度.....	26
拾壹、 參考資料及其他.....	27

## 摘要

以下將探討並分析一篇以時間差學習法為基礎寫成的 2048 人工智慧程式，在解讀程式碼當中適合取來應用的部位後，加以分析探討，是本研究的研究核心與主旨。在此篇程式當中，以執行時遇到的每一個版面當中的特徵為基礎，以時間差學習法的方式去學習，執行時若遇到先前曾經遇到過的特徵位置、形狀，就會依循先前所學的經驗，做出對自己最有利的判斷。為了使程式在執行時的勝率有所提升，我將著手探討每一種電腦可執行的特徵與勝率的關係。

首先分析出各種方格數組成的相異特徵種類，統計方格數對應的特徵數量後，得出了一個意外的發現，本研究命名之為「二倍定理」，對於往後電腦在執行高階方格數時有極高的參考價值。隨後為了因應本研究對於四到六方格數特徵的探討並避免窮舉，自創了一種新方法：「已得特徵放置方格法」，對於特徵種類的嚴謹分析有一大進展。之後將電腦所能支援且排除無探討價值後的130種特徵皆作數十萬次的遊戲勝率分析，排出了各種特徵之間的優劣，接著以特徵之間的組合與原版特徵比較，驗證了此系列研究方法的成功。

「機動性質」是本研究現階段最有可看性也最具應用性的新創研究方法。在本研究後半即藉由計算特徵版面上的位置對次數分析，以此方法也為先前實驗現象做了一些合理解釋。未來即將展現的就是機動性質回推法，藉此我正在設計利用更龐大數據的歸納，試著找出一個最理想的「機動性模型」，這種從被動獲得到主動回推的新思維演進方法，是可以被應用而且對於現有的特徵工程做一大革新。

本研究共耗時了九個月，以資訊教室48台電腦為主要設備，共計執行了近2500萬次的遊戲，達成本研究的最終成果。除了以上，也期望整篇研究以及新創研究方法可以被往後的資訊科技、人工智慧、大數據、特徵工程.....等領域應用。

## Abstract

The following is a discussion and analysis of an artificial intelligence program based on Temporal Difference Learning ( TDL ). After interpreting each part of the code, analyzing and discussing is the core and main purpose of this research. In this program, based on the features in each layout encountered during the program running, the TDL method can record it, then recognize them next time and thus make a more favorable decision. In order to raised the win rate and efficiency, I explored the relationship between each computer's executable features and winning percentage.

Firstly, I analyzed the different types of patterns composed of various amount of tuple. In this research, I named it "Double Rule", which can be well-applied to the development of high tuples in the future. Besides, "Pattern Extension Method" was also created while I was analyzing all kinds of patterns so as not to miss any possible one. After that, the 130 features that can be supported by the computer and excluded from the non-discussion value are analyzed for hundreds of thousands of times, and the advantages and disadvantages of various features were figured out.

"Times distributed Nature" is the most unique and applicable research method thought by this research. In the second half of this study, I calculated the times of each place on the layout, which can explained some phenomenon I discovered before. In the future, I am going to develop a new kind of research way which was named "Back Step from the Times distributed Nature". With this newly-created way, I can be able to figure out the most ideal model by generalizing plenty of patterns and their win rate or their learning efficiency. I think that these will be able to be well-applied to many fields.

The study took a total of nine months, with 48 computers in the computer science laboratory as the main equipment, and a total of nearly 25 million games were performed to achieve the final results of this study. What's more, I also hope that this research result will be applied in many other fields, such as computer science, artificial intelligence, big data, feature engineering or other technology products.

## 壹、研究動機

人工智慧是近年來的科技發展趨勢，而大數據亦是不可或缺的領域之一。西元2014年由倫敦 Google Deep Mind 科技所開發出的 AlphaGo 人工智慧圍棋軟體，實屬一款優越的 AI 程式，透過蒙地卡羅樹狀搜尋法結合以兩個深度神經網路，被它打敗過的人類職業棋手都表示其行為有如人類玩家一般，是有能力自發學習並進行自主訓練的。

除了 AlphaGo 所使用的蒙地卡羅樹狀搜尋法，還有一些傳統的人工智慧方法，例如：窮舉搜索、啟發式搜尋、深度優先搜尋演算、Alpha-beta 剪枝法.....等。本人數年前曾經研究過一篇以 Alpha-beta 剪枝法為底的程式碼，探討如何在一個既定版面當中，創造出利益的最大值，即勝率最大化、程式優化。利用程式裡評估函數的每一個變數，修改它們之間的權重，期望可以在眾多變數當中取得一個對程式執行時最有利的平衡點。在研究當中，我花了長期時間觀測每下一步決策後各變數在不同版面的變化，歸納出各種變數的特性，並以比例關係結合。最終找出了最佳權重，以有系統的方法成功有效的優化該人工智慧程式。

以下此篇研究起源於2018年9月，以資料當中一篇以TD Learning為底寫成的人工智慧程式為基礎，我著手從事這篇研究，期望能為資訊科技帶來更創新、更有效的結果與觀點。

## 貳、研究目的



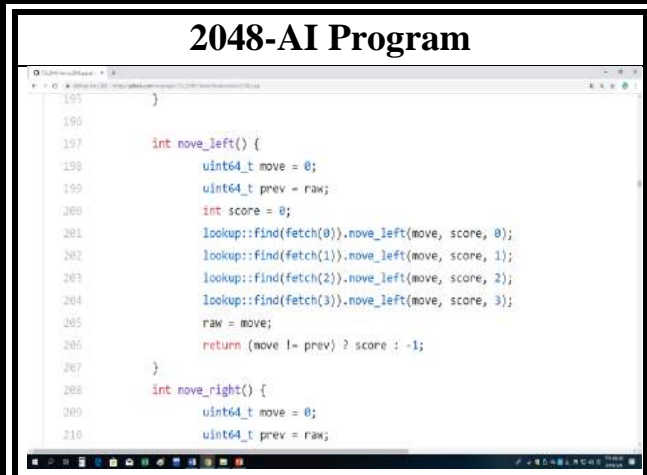


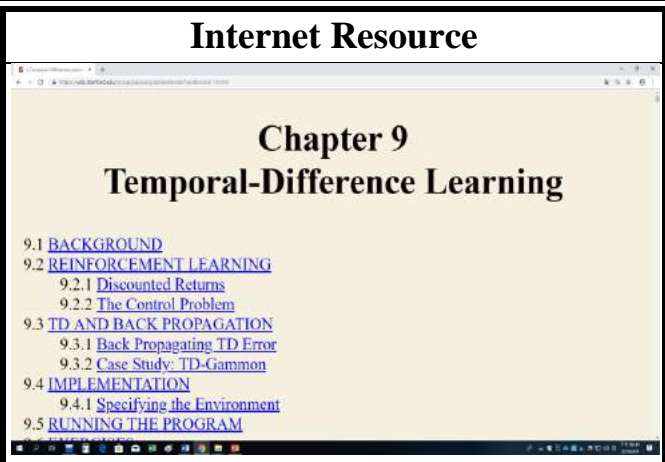
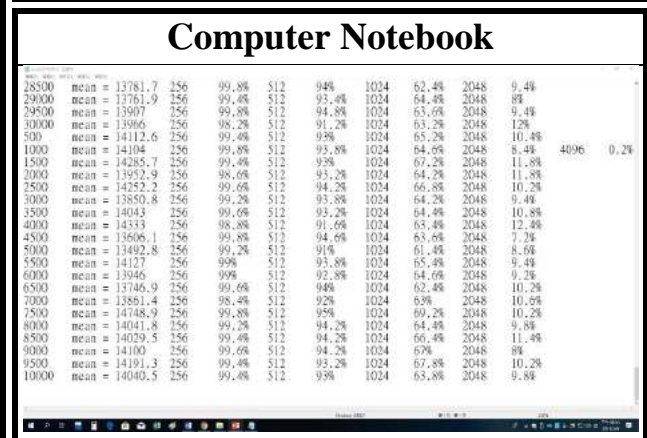
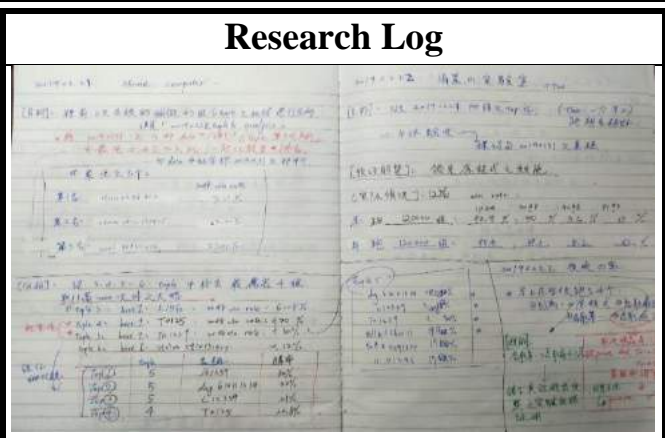
本研究期望可以達成以下三大項目標，而三大目標底下又有數個層次去完成：

- 一、透過電腦實際執行分析出所有可行特徵，並藉由大數據歸納出究竟何種特徵是最有利於遊戲進行的，最後期望可以透過妥善的擇優分組，明顯看出優化程式的成果。
- 二、找出一種方法是可以主動回推所有優良特徵的共通性，期望藉由數學方法加以應用，結合特徵工程(Feature Engineering)，試著回推出一組優良特徵所共通的特徵模型。
- 三、能夠對現今資訊科技及人工智慧領域做更相關的連結，諸如：將單人遊戲程式設計為雙人遊戲思考模式，並且在評估函數中加入更多面向的變數計算.....等。

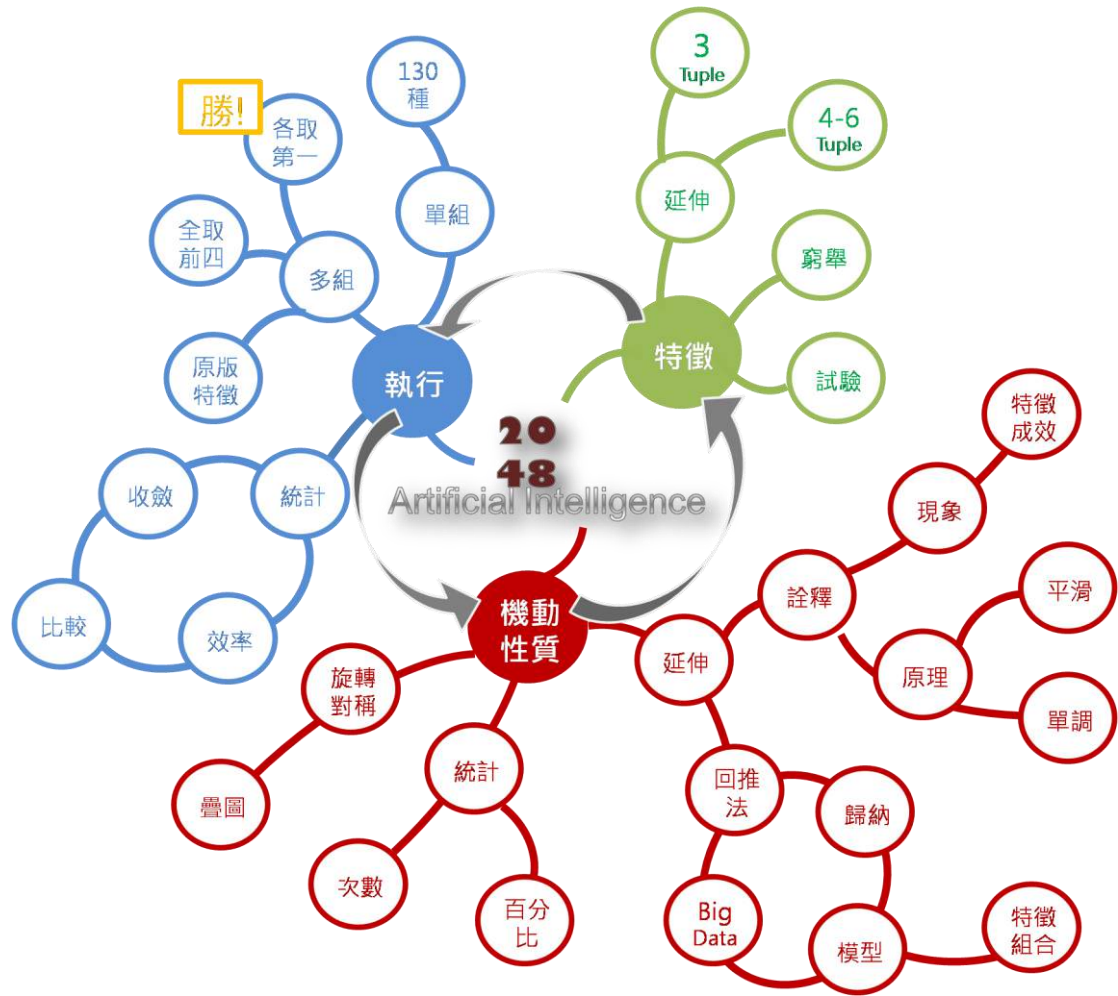
## 參、研究設備及器材

以下是在研究過程當中所使用的設備以及器材：（備註：本論文之附件有收錄研究日誌的重要段落，呈現許多計算過程及研究程序，亦紀錄了不少本人的觀點。）

(表3-1：研究設備器材列表)

	
<p>Programming Language : C++ Content : program &amp; content of 2048-AI</p>	<p>(1) To collect and arrange the big data (2) Tools for drawing win rate scatter charts.</p>
	
<p>Making adjustment and running the programs.</p>	<p>To comprehend the principles of TD Learning.</p>
	
<p>Tools for statistics.</p>	<p>Tools for recording my ideas, plans and progress.</p>

## 肆、研究架構

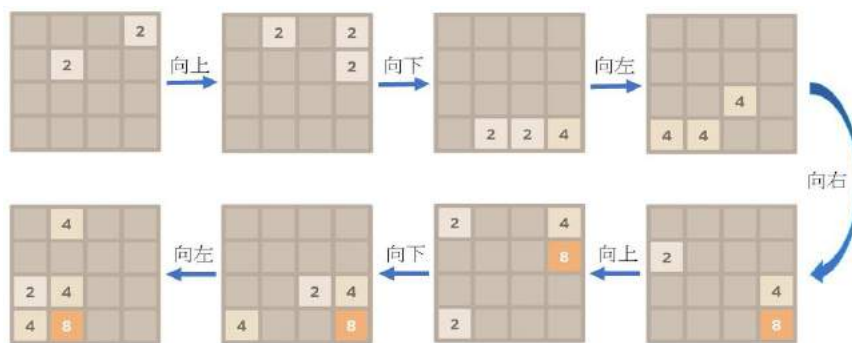


(圖 4-1：研究流程心智圖)

## 伍、研究對象介紹

### 一、「2048」遊戲簡介

2048 是一個單人益智遊戲，遊戲任務是使用方向鍵使方塊上下左右的移動，而兩個相同數字的方塊則會合併成一個高一次方的方塊，每次移動會多出現一個 2 或 4 的方塊。遊戲最終的目的是讓方塊合併成2048，即成功。

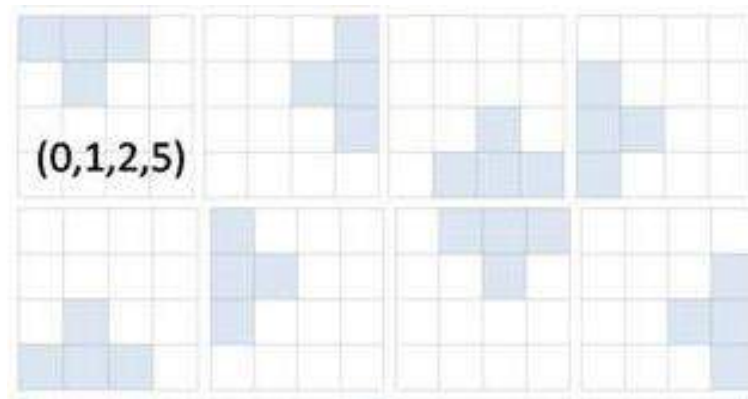


(圖5-1：2048 移動說明圖)

## 二、Pattern 介紹

特徵(Pattern)是由數個方格在版面上所形成的圖形，平移在不同位置的圖形成為不同特徵，但經過旋轉以及對稱所得到的皆列為同一種。例如：假設在程式中打入(0,1,2,5)，程式除了觀察圖中左上角的圖形範圍，也會同時觀察其他七種由旋轉以及對稱所得到的圖形。

(註解：以下的編碼皆是從4\*4版面左上方的格子，往右下數，從0編碼到15)



(圖 5-2：特徵旋轉對稱示意圖)

## 三、本專題所提及之專有名詞解釋及定義

在此專題中，有一些程式裡的專有名詞，以及為了這篇研究而取的專有名詞。為了使讀者可以方便並有效率的理理解此份研究，在這一頁將所有專有名詞做註解：

(表5-1：專有名詞定義)

專有名詞	注釋
Pattern (特徵)	非指圖形形狀，而是那些形狀在4*4的版面上的不同位置所得特徵。
Tuple(元組)	一個特徵所佔有的格數
連續圖形(自創名詞)	各方格之間是以至少一條邊作為接觸，而非僅一點的接觸
多組執行	由多種特徵共同學習，即為原程式的原本設定。
OO型(自創名詞)	(例：Z型、田型、L型...等)由該特徵所相近的形狀命名，此設計是為了增進實驗的效率並有效觀察出較佳的特徵是何種形狀所構成。
機動性質 (自創名詞)	一個特徵所能深入探討並且經過旋轉對稱後能夠包含的範圍



## 陸、研究過程與探討

### 實驗一：特徵分析與初步測試

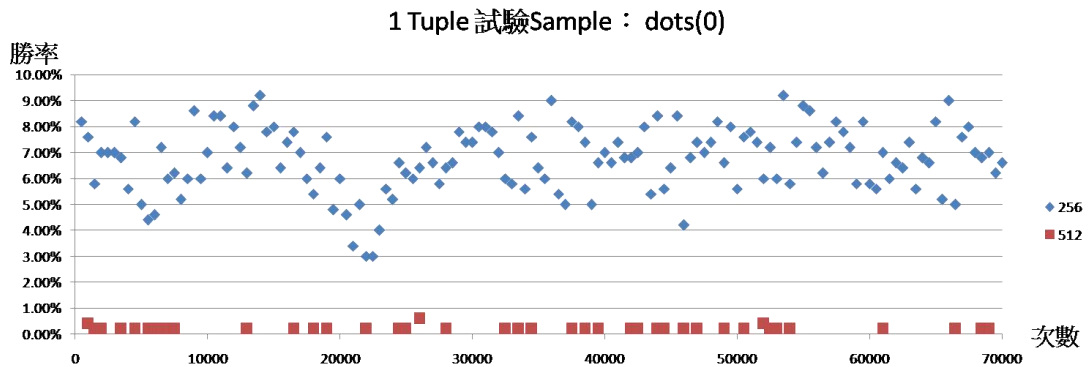
一、實驗目的：找出3-6Tuple所有可能在2048的4\*4版面出現的特徵

二、實驗說明：

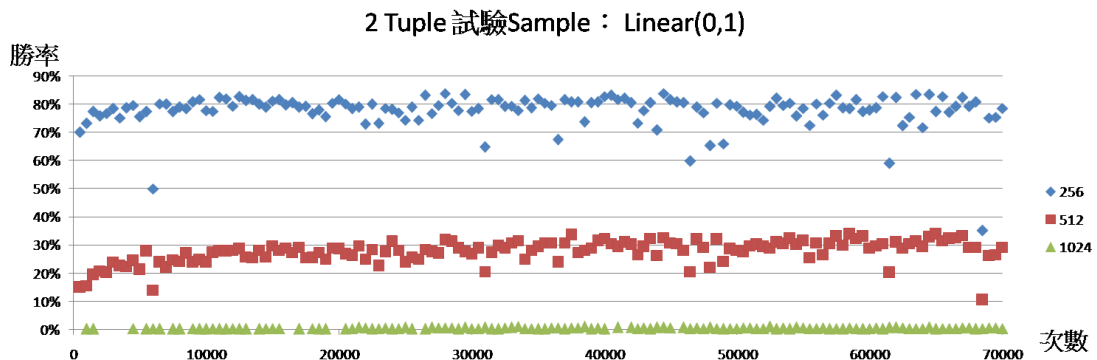
原程式設定有四組特徵，分別為(0,1,2,3,4,5)、(4,5,6,7,8,9)、(0,1,2,4,5,6)、(4,5,6,8,9,10)。為了找出更佳的特徵，決定測試其它Tuple數所構成的特徵。

在實際測試後，發現最大只能測到6個方格所構成的特徵圖案(即6Tuple的圖形)，再加上1Tuple完全沒有意義，而2Tuple也對獲勝率不太有幫助。因此，最小方格數決定從3Tuple開始。

為了證明以上對於1、2Tuple的否定，以下圖6-1及圖6-2為抽樣試驗7萬次的結果：其中，1Tuple取特徵dots(0)來試驗；而2Tuple取特徵Linear(0,1)來試驗。



(圖6-1：1Tuple 試驗結果)



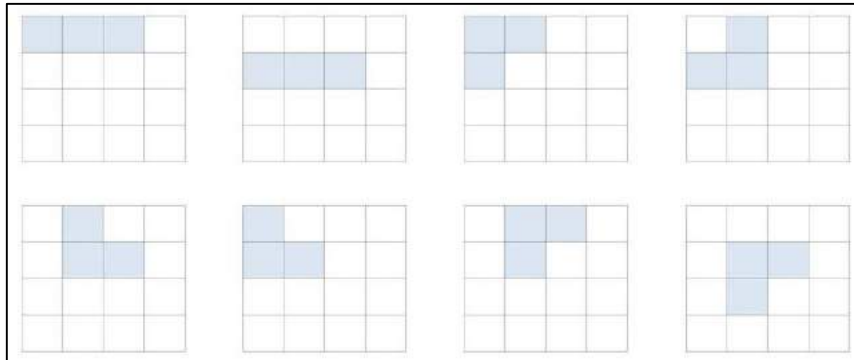
(圖6-2：2Tuple 試驗結果)

實際觀察以上兩圖表後，發現1、2Tuple的勝率經過7萬次的學習後幾乎沒有進步，而前10000次學習的勝率斜率也不高，因此證明1、2Tuple的確沒有參考價值。總結以上，本篇研究的單一特徵測試範圍由3Tuple到6Tuple，並逐一Tuple找出所有可能的特徵，以便分析。

三、實驗器材： 實驗日誌、紙、筆

四、實驗步驟：

1. 先找出不超過4\*4版面且為連續圖形的3方格圖形，可快速且正確找出以下8個圖形。



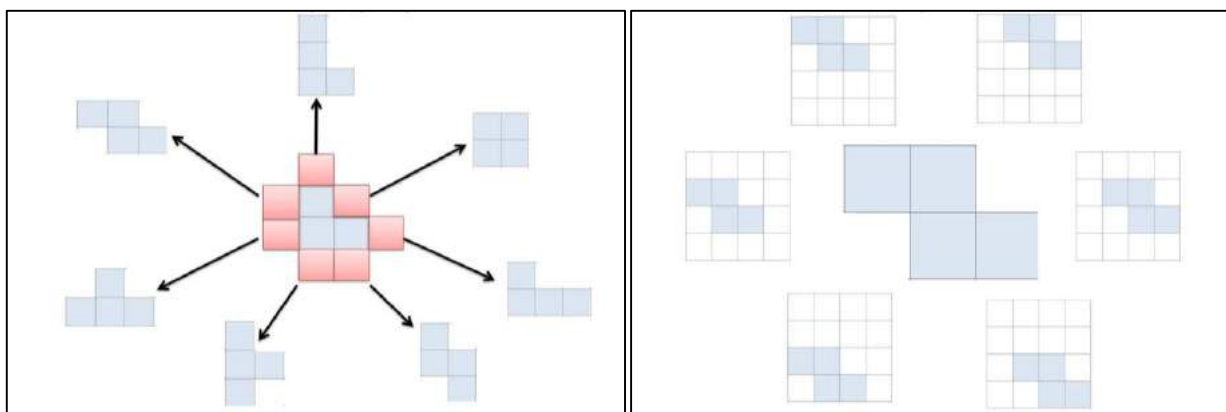
(圖 6-3：3Tuple 的所有特徵種類圖)

透過步驟一，可以找出3Tuple的2種排列方式：一種為三方塊並列，稱之為直線型，而在對稱與旋轉可能會重複的圖形位置刪除後，有(0,1,2)以及(4,5,6)這2種特徵；另為一種形狀為L型，一如前者之方法，刪除旋轉以及對稱的重複排列後，有(0,1,4)、(1,4,5)、(1,5,6)、(0,4,5)、(5,6,9)，共6種特徵。

2. 接下來，仍需找出4、5、6Tuple的所有正確特徵形狀。假如也利用如步驟一的那種窮舉方法。那就極有可能漏掉，畢竟Tuple數增加，特徵數也會隨之增加，出現更多不同的新圖形。

因此，有一個方法可以使分析多Tuple的特徵時不漏掉。想法如下：每個4Tuple的圖形一定可以取出3Tuple的連續圖案包含在其中；倒過來想，只要將3Tuple的圖形：L型、直線型的每一個接觸邊都加上一個方格，統計下來並將重複形狀、旋轉、對稱的都刪除，便會得到所有4Tuple的圖形種類了。(如圖6-4)

3. 接著再將以上的形狀種類，放入不同位置，且刪去旋轉對稱者，即得所有4Tuple的正確特徵。此種方法可以讓我們在往後研究高階Tuple時不易漏掉。(如圖6-5)



(左圖6-4：3Tuple延伸至4Tuple的方法示意圖；右圖6-5：4Tuple在各種位置形成不同特徵種類)

4. 依照以上原理，又可得到5Tuple以及6Tuple的所有正確特徵。因為可以組成眾多不同的形狀，故在此篇研究中，每種形狀皆會取名。諸如:L型、Z型、+型...等。一方面做實驗時較好歸納，一方面可以有效觀察何種形狀的效果最佳。

#### 五、實驗結果：

表6-1為不同Tuple數所含的特徵數量統計，可在其中發現一些規律：

(表6-1：各Tuple的特徵數量統計)

Tuple數 比較	3Tuple	4Tuple	5Tuple	6Tuple
Pattern 數量	8	17	33	72
倍率		$\frac{8}{17}=2.125$	$\frac{17}{33}=1.941$	$\frac{33}{72}=2.182$

#### 六、實驗討論：

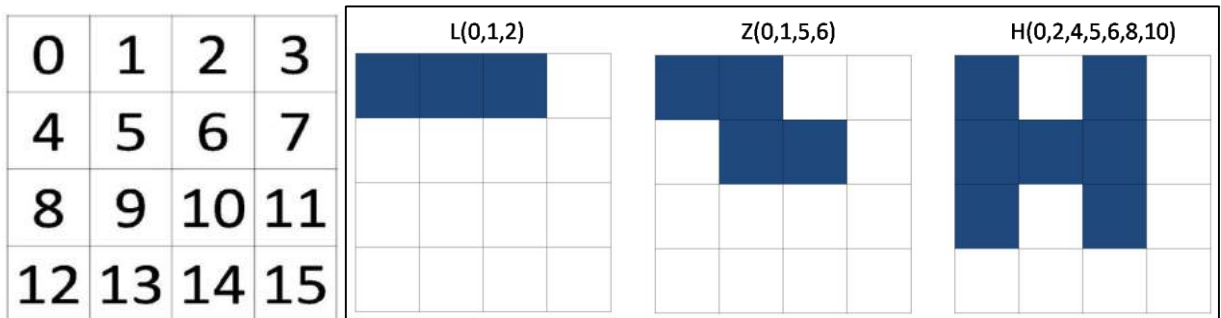
1. 「已得特徵放置方格法」：本實驗想出一個較為嚴謹的方法整理出所有可能特徵：透過已有的簡單特徵在每一條接觸邊上各加一個方塊組成新的一系列特徵，對往後有不小的助益。
2. 其他特徵分析方法：除了步驟二、三的自創方法外，還想到了可以利用IUPAC化學異構物命名方式，配合繪製異構物的方法，也能有效率地找出正確的特徵。
3. 「二倍定律」：在找尋所有可能的特徵中，發現了4Tuple的特徵數量是3Tuple的兩倍，而5Tuple的特徵數量大約是4Tuple的兩倍，6Tuple亦如此。此定律的發現有助於探討高階Tuple時，能了解每一個Tuple數大致會有多少種特徵，極具參考價值。

## 實驗二：單一特徵的執行與數據分析

一、實驗目的：透過實際執行找出各Tuple最佳的特徵。

二、實驗說明：

在該份程式檔中，是利用將圖形轉成數字的方式，將特徵放在圖6-6這個4\*4的版面上，查看對應到的數字，便可歸納。而圖6-7舉出一些特徵形狀及本研究命名的例子，如：(0,1,2)為直線型，(0,1,5,6)為Z型，而(0,2,4,5,6,8,10)為H型...等。



(圖6-6：0-15編號示意圖；圖6-7：直線型、Z型、H型特徵解說)

在特徵轉換成數字後，將透過實際執行探討各種特徵的效能，而單一執行便是為了減少變因，避免其它特徵的互相影響，純粹觀察出該特徵的效能。

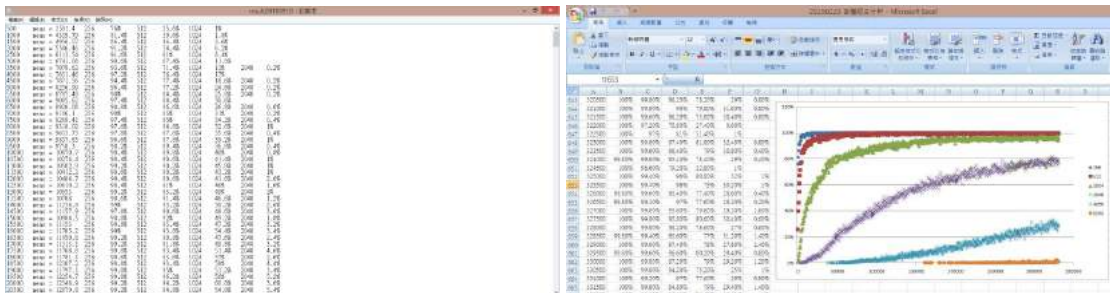
三、實驗器材：Dev C++ 5.11版本、記事本、Microsoft Office Excel

四、實驗步驟：

1. 將下列程式碼中大括弧區域內改為實驗一整理出的每一種特徵：

```
// initialize the features
tdl.add_feature(new pattern ( { 輸入欲測試之特徵 } ) );
```

2. 執行程式並透過記事本蒐集數據(如圖6-8)，將完整的數據儲存到Excel檔編列。最後繪製勝率散佈圖觀察其勝率趨勢(如圖6-9)。



(左圖6-8：程式執行後的數據蒐集；右圖6-9：Excel的數據分析以及勝率散佈圖繪製)

為了得知散佈曲線的趨勢，故挑了一個特徵，L0126，設定執行了170萬次，並製成趨勢圖，視為「空白實驗」。之後，將實驗一找到的所有特徵輸入至程式碼當中，比較其數據，找出較佳者。

## 五、實驗結果：

表6-2為3-6Tuple所有特徵的勝率分析，表上的數據皆是跑了十餘萬次後的結果，而為了避免誤差，本研究選取學習次數末5000次之間的平均。以下各種格數(Tuple)的分析排名，皆只取前五有效能的收錄進此份研究報告書中。另外由於1024勝率皆相近，故表6-2的排名主要以2048的獲勝率為主。

(表6-2：3-6Tuple 特徵勝率排名)

<b>3Tuple分析 (10萬次)</b>	<b>1024</b>	<b>2048</b>	<b>排名</b>	<b>4Tuple分析 (10萬次)</b>	<b>1024</b>	<b>2048</b>	<b>排名</b>
L156	44.00%	5.20%	1	T0125	80.80%	18.00%	1
直456	43.40%	3.26%	2	L0126	73.40%	16.40%	2
L125	26.20%	1.40%	3	Window1256	58.00%	10.60%	3
L145	15.60%	1.00%	4	Z45910	50.00%	9.00%	4
L014	30%	0.40%	5	L1567	49.60%	9.00%	5

<b>5Tuple分析 (12萬次)</b>	<b>1024</b>	<b>2048</b>	<b>排名</b>	<b>6Tuple分析 (17萬次)</b>	<b>1024</b>	<b>2048</b>	<b>排名</b>
T01259	79.00%	31.00%	1	Stairs0458910	80.80%	18.00%	1
dog610111314	79.00%	23.00%	2	stairs5910131415	73.40%	16.40%	2
L12359	74.00%	20.00%	3	Cool489101213	58.00%	10.60%	3
U01246	71.00%	20.00%	4	Cool045689	50.00%	9.00%	4
tallz1261011	74.00%	16.00%	5	J1236910	49.60%	9.00%	5

為了得知實驗一所整理出來3-6Tuple中的130種特徵之間在同樣學習次數時的強度關係，本研究將每一種特徵的數據結合起來，取相同的學習次數(11萬次)為參考次數，做出了前四名的優劣比較，將運用在下一個實驗當中。(如表6-3)

(表6-3：130全種Tuple之最佳四種)

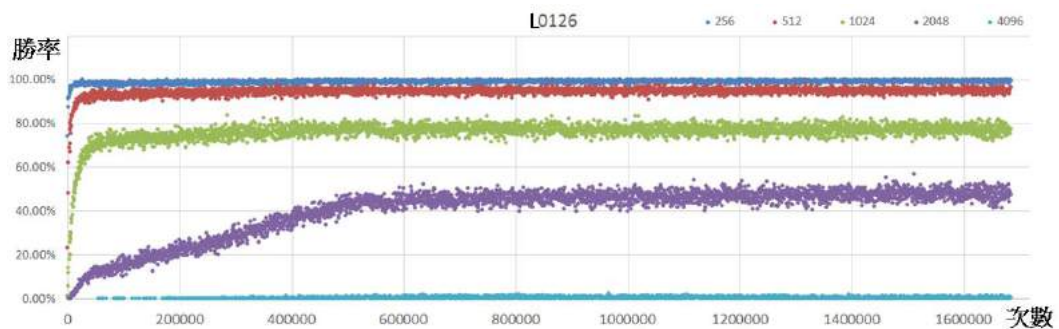
<b>4-6Tuple分析 (11萬次)</b>	<b>Tuple 數</b>	<b>2048</b>	<b>排名</b>
T01259	5	30.00%	1
Dog610111314	5	22.00%	2
L12659	5	21.00%	3
T0125	4	20.80%	4

透過表6-2，可以明顯發現各種Tuple的特徵優劣。利用實驗一所做的特徵命名可以看出最有利於程式執行的形狀，例如：在4Tuple與5Tuple中皆是以T型的特徵具有最高的勝率；而從3-5Tuple的特徵排序當中也可看出L型對於程式的學習亦有不小助益。而在表6-3中也可以發現4Tuple組成的特徵：T0125位居第四名，不過前三名的勝率都屬於5Tuple，可見6Tuple雖然方格數多，在執行次數相同的時間，並沒有比較高的勝率與學習效果。

## 六、實驗討論：

### 1. 執行實驗二前的「空白實驗」—清楚了解一條特徵勝率曲線的收斂情況：

為了避免在實驗二研究過程中遺失重要實驗數據，在操作前先隨機挑取了一個特徵：L0126，執行過量次數的遊戲，並一樣透過記事本以及excel蒐集數據，繪製成一個學習次數170萬次的勝率散佈圖：(本研究散佈圖以一個數據點表示500次的平均)



(圖6-10：空白實驗抽樣L0126—執行170萬次之勝率對次數散佈圖)

藉由圖6-10的空白實驗可以得知一個特徵勝率曲線的型態，都是從接近0%開始，隨著學習次數增加，勝率也以近乎斜直線的型態增長，而不同獲勝條件也有不同的獲勝情況，呈現在收斂時的勝率中。同時也可以得知一條勝率曲線收斂之後，就很難再有所進步，不必考量過量的學習次數。此一空白實驗為實驗二的操作打下基礎。

### 2. 從「單一特徵」拓展到「多組特徵」的分析：

在實驗一以及實驗二中，已歸納出較佳的數個特徵，接下來的實驗中將測試出最佳組合並與原程式碼做比較。

### 3. 發現與設想不同的「格數(Tuple) 成效結果」：

在此實驗中可以看出5Tuple的整體效果是比6Tuple還要佳的，而完整的排序大致上是5Tuple>4Tuple>6Tuple>3Tuple，與普通的直觀想像大不相同。原先認為6Tuple因為涵蓋的Tuple數較多，故其勝率應較大，卻發現事實不然。為了解釋合理現象，將由後續實驗持續探討。

### 實驗三：多組特徵的執行與學習效能分析

一、實驗目的：透過四個特徵的較佳組合，實際執行後證明成功優化了程式。

二、實驗說明：

在原程式中，設有4個特徵，分別為：(0,1,2,3,4,5)、(4,5,6,7,8,9)、(0,1,2,4,5,6)、(4,5,6,8,9,10)。在此將由實驗二所得的較佳特徵，組合出較優的勝率。

對於以上有兩種想法：第一種為總取前四，是將3、4、5、6Tuple的所有特徵綜合起來取最佳的前四名；第二種是各Tuple數各取出第一名來執行。而究竟何者具有較佳的學習速率、勝率將在結果呈現。

三、實驗器材：Dev C++ 5.11版本、記事本、Microsoft Office Excel

四、實驗步驟：

1. 將總取前四以及各取第一的特徵逐一輸入程式當中。

```
// initialize the features
tdl.add_feature(new pattern ({ 特徵1 }));
tdl.add_feature(new pattern ({ 特徵2 }));
tdl.add_feature(new pattern ({ 特徵3 }));
tdl.add_feature(new pattern ({ 特徵4 }));
```

以下列出實驗三欲分析的三組特徵組合列表：

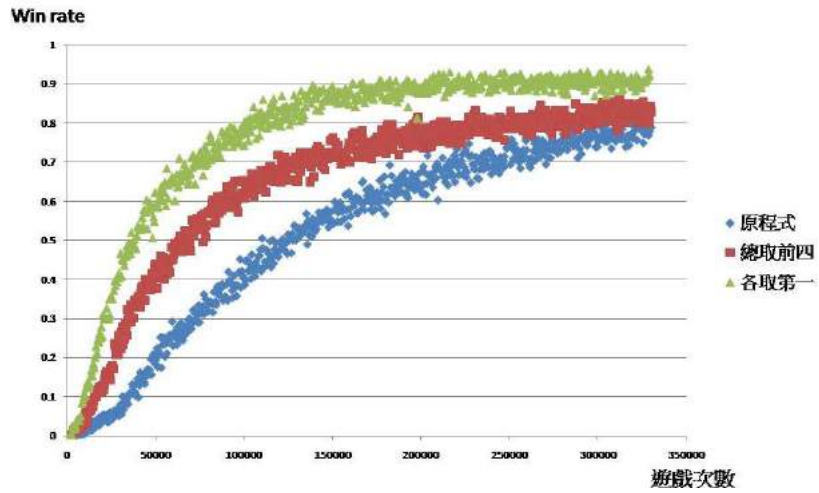
(表6-4：特徵組合之組成特徵)

成分 \ 組合	各取第一	總取前四	原版特徵
特徵1	L(1,5,6)	T(0,1,2,5,9)	Thumb(0,1,2,3,4,5)
特徵2	T(0,1,2,5)	Dog(6,10,11,13,14)	Window(0,1,2,4,5,6)
特徵3	T(0,1,2,5,9)	L(1,2,5,6,9)	Thumb(4,5,6,7,8,9)
特徵4	Stairs(0,4,5,8,9,10)	T(0,1,2,5)	Window(4,5,6,8,9,10)

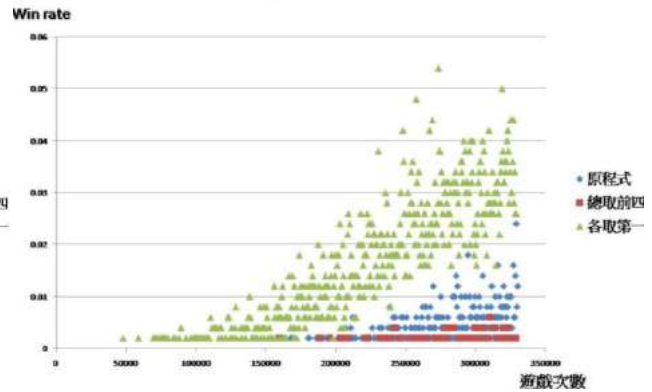
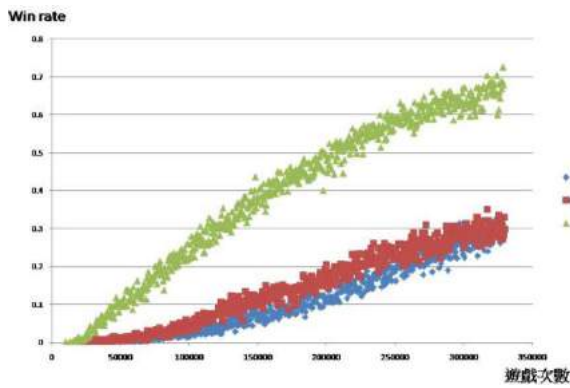
2. 如同實驗二，執行程式並透過記事本蒐集數據，將完整的數據儲存到EXCEL檔編列。最後繪製勝率散佈圖觀察其勝率趨勢。

五、實驗結果：

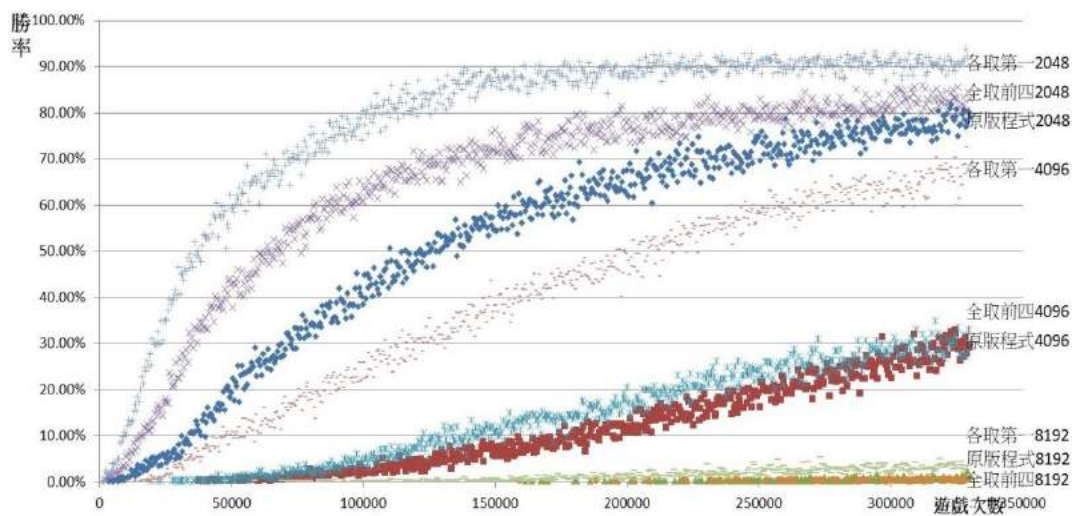
以下為執行33萬次的數據，而最終結果證明了新特徵的確優於原程式。其中尤以各取第一特徵組合具有最佳勝率。



(圖6-11：2048勝率比較圖)



(圖6-12：4096勝率比較圖；圖6-13：8192勝率比較圖)



(圖6-14：獲勝條件：2048/4096/8192 三種特徵組合勝率分析圖)



## 六、實驗討論：

1. 修改後的程式4096勝率在執行33萬次後已逐漸逼近原程式的2048勝率，成果相當可觀。此外，由2048勝率的散佈圖來看，可發現各取第一特徵組合的勝率高達93%，較原版特徵組合高出了將近10%，而4096勝率則高出了40%，屬於成果最佳的部分。
2. 國中時曾經研究過另外一種2048人工智慧程式，它蘊含著平滑度以及單調性的性質，也就是說，要打好一場漂亮的勝局，可能需要在2048版面上盡量配出一個對自己有利的數字方塊版面，可利用評估函數(Evaluation Function)解決此問題。

## 七、學習效能&收斂成果分析

1. 分析目的：

探討一個特徵組合的勝率隨著學習次數的變化，並經由軟體算出其學習效率以便於不同組合間的嚴謹比較，亦希冀可以推廣到實驗二的單一特徵分析，更希望此套算法可以運用在更多不同的領域上，諸如人工智慧、特徵工程.....等。

2. 分析說明：

本研究將學習效能 (E<sub>L</sub>) 定義如下：勝率散佈曲線學習勝率達到目標勝率時所需效率。即

$$E = \frac{\text{目標勝率}(\%)}{\text{學習次數}(\text{萬次})}$$

以下將藉由勝率曲線得知達標時，每一種特徵組合的效率。

3. 分析器材： Microsoft Office Excel、計算機

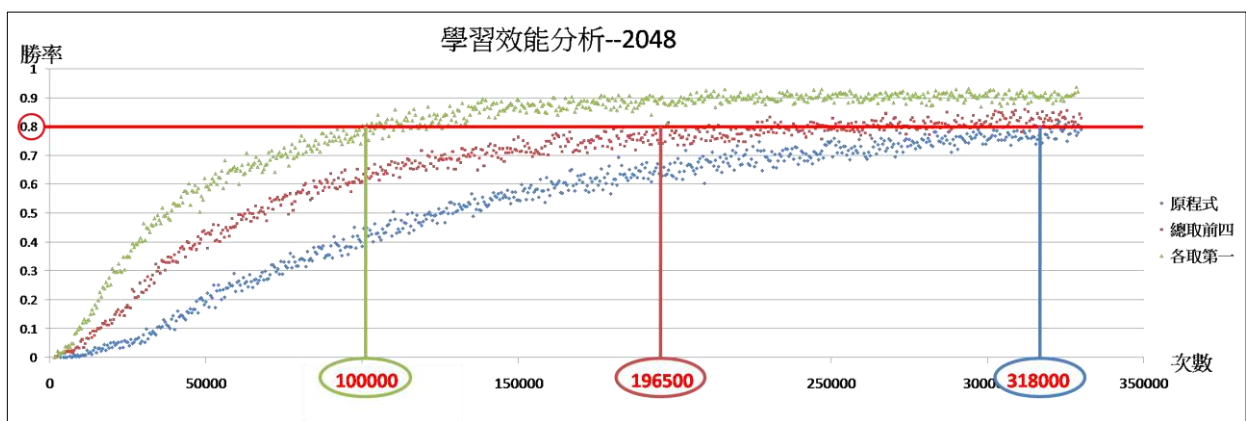
4. 分析步驟：

(1) 設定一個相同獲勝條件共有的勝率目標，尋找該勝率分布數據點第一次達標時的次數。此方法沒有機率性誤差的疑慮是因為，每一個點都是經過500次平均後的結果，數據量夠大，所以每個數據點都具參考價值。

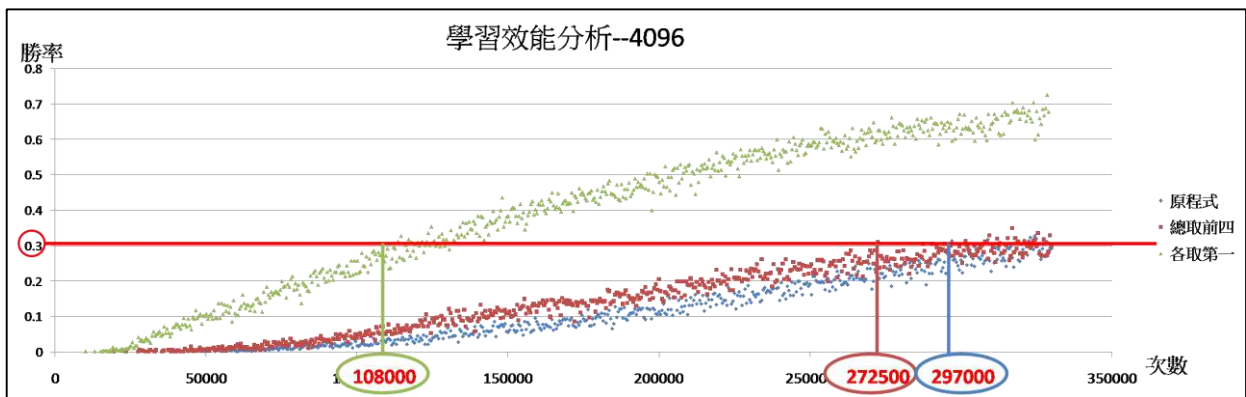
(2) 接著將三種特徵組合所對應到的學習次數記錄下來，並計算上文所定義的學習效能(E<sub>L</sub>)，比較其與原版效能的大小幅度(效能比較)。在本研究階段暫時不討論8192獲勝條件的學習效能，因為從實驗三的結果可看出經過33萬次學習後勝率仍然不大，因此不討論。

5. 分析結果：

透過實際分析，可以發現學習效能成果如下。為了比較，2048獲勝條件的散佈圖以80%的高勝率為目標勝率；而4096獲勝條件則以30%為目標勝率。透過圖6-15以及6-16，觀察到效能最佳的各取第一特徵組合皆是最快達標的。其在經過一萬次的學習之後即可達到80%的高勝率，而原版特徵則需要31萬8千次才能達到一樣的效果，可見從實驗一到實驗三的特徵分析有極大的成效，效能約為原本的3.17倍左右。而4096的學習效能亦如此，遠本需要29萬7千次才能達成的勝率，現在只需10萬8千次即可成功，效能約為2.75倍。以下將由圖表呈現其學習效能證明本研究之優化：



(圖6-15：學習效能分析/獲勝條件：2048)



(圖6-16：學習效能分析/獲勝條件：4096)

(表6-5：實驗三之效能比較)

獲勝條件：2048				獲勝條件：4096			
目標勝率：80%				目標勝率：30%			
特徵組合	各取第一	總取前四	原版特徵	特徵組合	各取第一	總取前四	原版特徵
學習次數(萬)	10.00	19.65	31.80	學習次數(萬)	10.80	27.25	29.70
學習效能	8.00	4.07	2.52	學習效能	2.78	1.10	1.01
效能比較	3.17	1.62	1.00	效能比較	2.75	1.09	1.00

## 實驗四：機動性質的創新、整合與推廣

一、實驗目的：將實驗三的三種組合拿來做數學相關的位置分析，期望能有所推廣。

二、實驗說明：

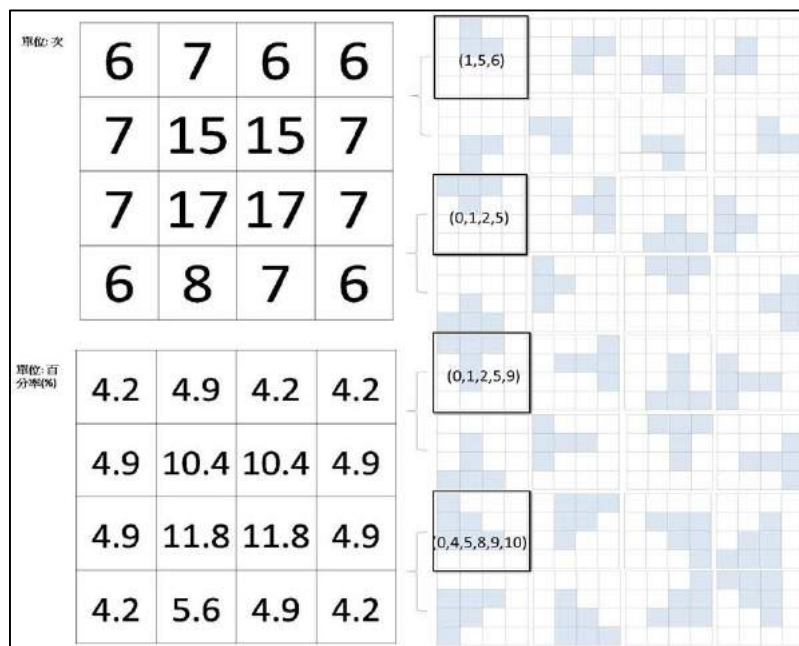
做完了如此龐大的實驗數據後，萌生了幾個想法：究竟特徵在旋轉對稱後，有哪些方格是被顧及到的？而那些方格被忽略？又那些方塊較受重視呢？

因此我設計了一種特性－機動性，意指方塊所能顧及之範圍、次數以及百分比。為了使讀者更加理解，打比方來說：日常生活中一輛腳踏車可以走遍大街小巷，屬於機動性高的工具；然而一輛遊覽車卻只能在大馬路上行駛，是屬於機動性小的。

與此同理，本研究將此想法推廣到2048版面與特徵的分析上。

三、實驗器材：實驗日誌、筆、計算機

四、實驗步驟&結果：



(圖6-17：機動性質的統計與次數及百分率分布圖)

1. 圖6-17的右圖的每八個是同一種特徵經過旋轉對稱後所得到的圖形，各種皆是同一種特徵所會考慮到的範圍。為了方便探討機動性，將之全數列出。
2. 圖6-17的左上圖即是將右圖的所有特徵疊圖後各個格子的次數總和。
3. 圖6-17的左下圖為百分比的散佈圖，是將次數除以右圖的總次數所得到的。此步驟是為了推廣，在往後能將各種不同Tuple數總和的特徵組合都可以分析，故利用同理也分析了原程式以及全取前四的機動性散佈圖。

【算法】

$$\text{機動性各方格百分率} = \frac{\text{該格的次數}}{\text{總次數}} = \frac{\text{該格的次數}}{\text{每種特徵的Tuple數總和} * 8}$$

$$\text{舉圖6-17之左上圖為例，機動性值即為} = \frac{6}{(3+4+5+6)*8} = \frac{6}{144} = 4.2\%$$

4. 與上述同理，應用在實驗三的另外2組特徵組合，可得以下為三個組合的機動性圖。

原程式機動性圖				全取前四機動性圖				各取第一機動性圖			
單位:百分率(%)				單位:百分率(%)				單位:百分率(%)			
3.6	5.4	6.0	3.6	3.3	5.3	5.3	3.3	4.2	4.9	4.2	4.2
6.0	8.9	9.5	6.0	6.6	10.5	9.2	5.3	4.9	10.4	10.4	4.9
6.0	9.5	10.2	6.0	6.6	9.2	8.9	5.3	4.9	11.8	11.8	4.9
3.6	6.0	6.5	3.6	4.6	6.6	6.6	4.6	4.2	5.6	4.9	4.2

(圖6-18：3Tuple 的所有特徵種類圖)

五、實驗討論：

各取第一機動性圖				單位:百分率(%)
4.2	4.9	4.2	4.2	
4.9	10.4	10.4	4.9	
4.9	11.8	11.8	4.9	
4.2	5.6	4.9	4.2	

(圖6-19：各取第一機動性圖)

1. 在步驟四的三個機動性質分布圖中，皆以中央四格分布最密集，其次為邊上中央兩格，最低為四角落，而這三種地方內的機動值皆很接近，推測此即各區塊的重要性差別。
2. 圖6-19之綠區以及藍區的總合相同，而且此為慣例，每一個例子算出來都是這樣。可見此一特性是數學上旋轉、對稱後加成所得到之必然結果。

3. 從討論第一點可發現數值是從角落最小，往中間漸漸變大，此種漸進的現象與人類玩家或其他種人工智慧程式常見的策略相似，因此未來將以此種現象與前文所提到的單調性以及平滑度的性質做連結，與評估函數整合。

各取第一機動性漸層圖 <small>單位:百分率(%)</small>				平滑度				單調性			
4.2	4.9	4.2	4.2			512	512	64	128	256	512
4.9	10.4	10.4	4.9			512	512		64	128	256
4.9	11.8	11.8	4.9							64	128
4.2	5.6	4.9	4.2								64

(圖6-20：從各取第一機動性圖延伸至平滑度以及單調性值的思考)

平滑度(圖6-20-中)，是一種2048遊戲的策略之一，強調將同數字聚集起來，因為假若同一種數字分別散據各地，則要合併就困難許多，勝率也就不高。

單調性(圖6-20-右)，屬於另一種常見的玩法策略，主要將數字方塊漸層排列，使未來相同數字合併時，能夠較有機會發生「連鎖合併效應」：256與256合併變成1024後，即可馬上與周邊的1024合併成2048，再移到下一階即可與2048合併，以此類推。

4. 上述之討論較接近數學方面的探討。不過仍與2048版面以及特徵組合有很深的關聯。或許之後可以由此知道程式學習，到底學到了什麼?以及暗藏其中的遊戲策略?只要知道了那些導致高勝率的潛在策略，便可以跳脫出TD Learning的框架，將之寫為新的程式檔，假設許多變數分別表示各種功能，計算總值與評估函數，便可以在每一次移動前，考慮每一種特質。希冀未來可以實際操作並能在排除機器學習之外，達到與之相差不遠的成果，最終將兩種演算法加以整合，使人工智慧程式具有最大功效。

## 柒、研究結果

### 一、「已得特徵放置方格法」&「二倍定律」的發現

透過實驗一，先用最原始的方法找出3個Tuple組成的特徵，在各邊上加上一個方格形成新形狀，再放入不同位置形成不同特徵，即找到4Tuple的17個特徵。同理，即可找到5Tuple組成的33種特徵以及6Tuple的72種特徵。而也在其中發現特殊的結果，每個Tuple所形成的特徵的數量，皆是前一個Tuple所形成的特徵數量的兩倍。

(表7-1：各Tuple特徵數量統計)

	3Tuple	4Tuple	5Tuple	6Tuple
特徵 數量	8	17	33	72

### 二、成功找出所有優良特徵&各Tuple數的最佳特徵

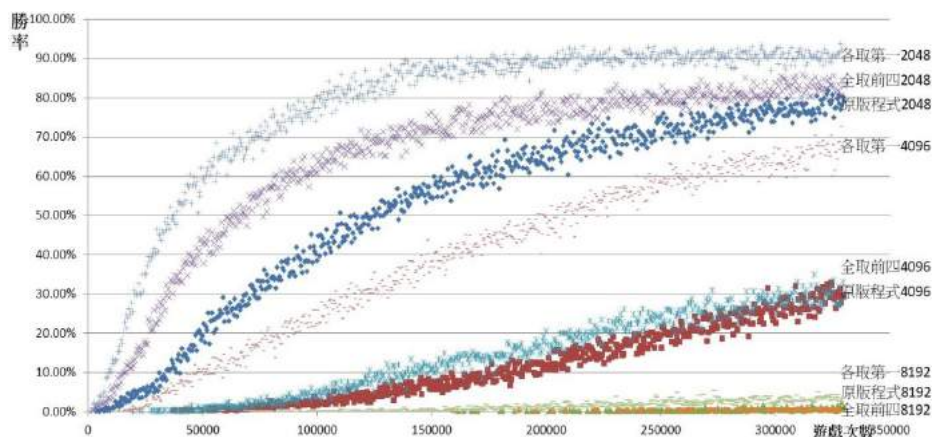
在實驗二中，透過3-6Tuple的130種特徵的實際測試，成功找出相對於同Tuple的特徵中最好的幾個。而因為各種Tuple的收斂程度不一，因此下表也特別列出參考次數。

(表7-2：各Tuple最佳效能表)

各Tuple最佳效能列表				
	特徵種類	1024win rate	2048win rate	參考次數
3Tuple	L156	44.00%	5.20%	10萬次
4Tuple	T0125	80.80%	18.00%	10萬次
5Tuple	T01259	79.00%	31.00%	12萬次
6Tuple	stairs0458910	77.00%	23.33%	17萬次

### 三、2048勝率提升10%，4096勝率提升40%

實驗三即是將實驗二的結果組合起來與原程式比較，另外加入了一個特徵組合：以11萬次為參考次數，從130種特徵中挑出最強的4個，自成一組。此實驗主要是在比較此三個特徵組合的效能優劣。



(圖7-1：2048/4096/8192 三種特徵組合勝率分析圖)

圖7-1中可以明顯發現，8192因為門檻太高，三組能達標的機率都不大而無明顯差異。不過各取第一那一組無論在2048或4096仍是領先。可觀察到各取第一的學習速度幾乎超過了原特徵組合的2倍以上，更是快超越原版特徵的2048勝率。而2048勝率僅有各取第一特徵組合可以達到93%的成果，領先原版特徵將近10%。

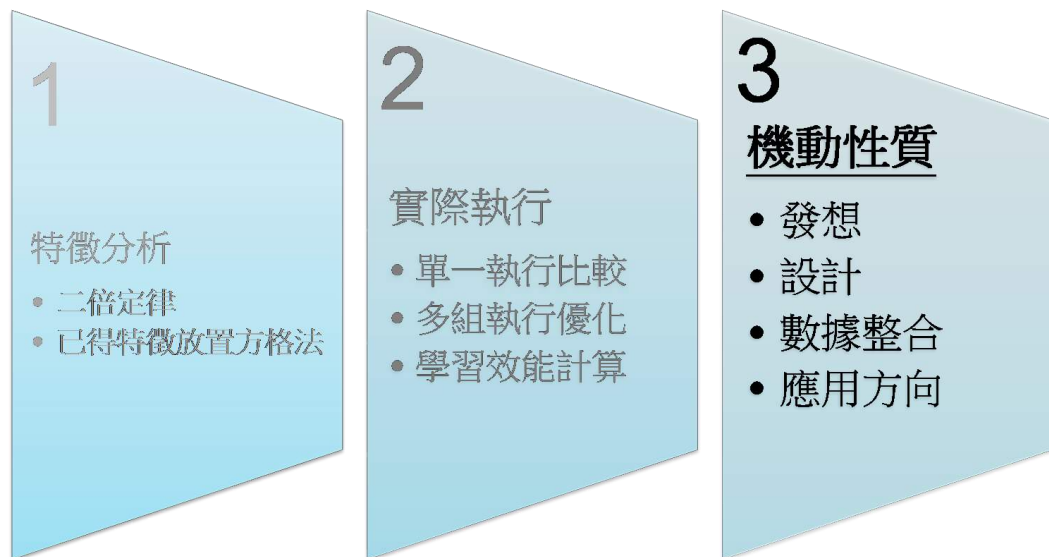
#### 四、自創新性質－機動性質

經由實驗一到實驗三的數據分析後，突發奇想並設計出了實驗四：機動性質。在這其中發現了一些規則，如中間四塊所考慮到的機率總是特別大，而角落卻最小，並在機動性的漸層分析中，隱約可看出一些可被寫入評估函數的遊戲策略，為未來發展方向。

原程式機動性圖	原配:百 分率(%)	全取前四機動性圖	原配:百 分率(%)	各取第一機動性圖	原配:百 分率(%)	各取第一機動性漸層圖	原配:百 分率(%)
3.6 5.4 6.0 3.6		3.3 5.3 5.3 3.3		4.2 4.9 4.2 4.2		4.2 4.9 4.2 4.2	
6.0 8.9 9.5 6.0		6.6 10.5 9.2 5.3		4.9 10.4 10.4 4.9		4.9 10.4 10.4 4.9	
6.0 9.5 10.2 6.0		6.6 9.2 8.9 5.3		4.9 11.8 11.8 4.9		4.9 11.8 11.8 4.9	
3.6 6.0 6.5 3.6		4.6 6.6 6.6 4.6		4.2 5.6 4.9 4.2		4.2 5.6 4.9 4.2	

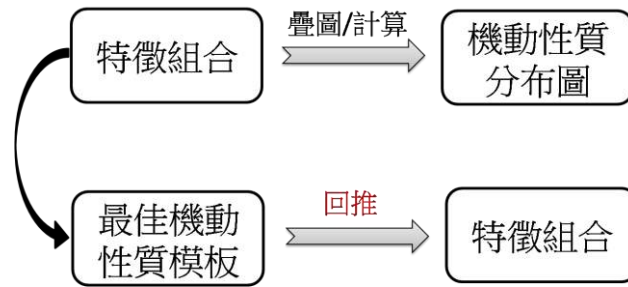
(圖7-2：機動性圖以及漸層性質)

總結以上，本研究從既有的程式執行更深入的大數據分析，為了使整體研究更加嚴謹，發展出了自創機動性質，在其中我看見了很實質的可發展性－「可回推性」。接著只要找出一種能配合本研究的數學方法，就可以將特徵性質發展的淋漓盡致。總而言之，這將為我未來的研究產生不一樣的影響。

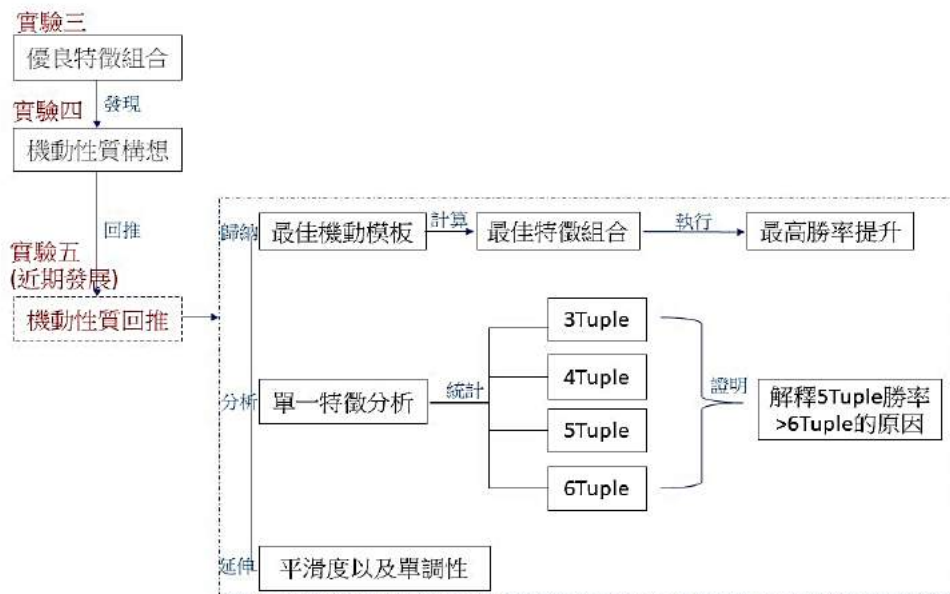


(圖7-3：研究階段成果圖)

## 捌、討論



(圖8-1：「從被動求值到主動求解」示意圖)



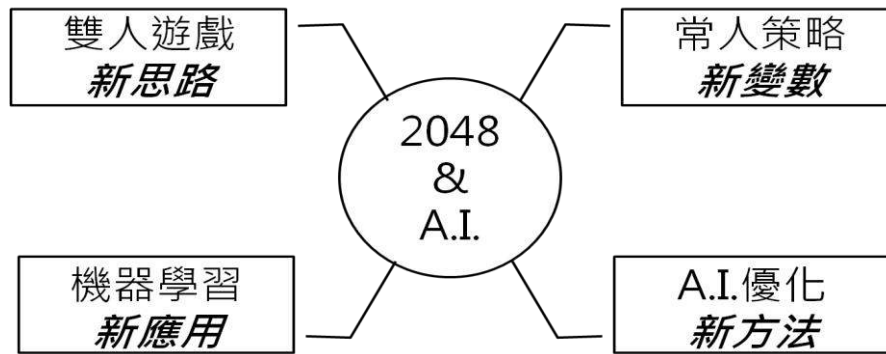
(圖8-2：自創機動性質之應用以及發展)

此為顛覆想法之考量，針對實驗四的自創機動性質可進行更深入的討論：

機動性質是一種主動討論的概念，先跑完大量的程式數據，再來挑出優越者進行特徵以及版面的性質分析，即可得到如實驗四所呈現的機動性質分布圖。不過，也出現一個想法：將機動性質往回討論，先歸納出一些相對優越的特徵組合，討論其共同性質，再將那些性質用在挑選特徵上，畢竟現在也不能嚴格證明實驗四所看到的各取第一是最佳的特徵組合，因為也許某些特徵單獨跑程式的時候是優越的，但是兩個以上一起跑甚至是四個一起的時候，可能會因為互相干擾、加成性...等因素導致結果不如預期。故應該深入理解機動性質百分比的配置後，實際從機動性質散佈圖，透過人工的方式，被動挑選特徵去符合機動性質百分率，查看實驗結果，即可驗證。



## 玖、未來展望



(圖9-1：未來展望)

### 一、 結合「特徵工程」(Features Engineering)－自創最有效能的特徵回推方法

透過大量數據歸納出優越的特徵模型，了解它們勝出的原因並結合以機動性質。期望找出多種數學方法有效率並嚴謹地回推產生最佳機動性的特徵組合，從中找到規律，此方法實屬一種跳脫的思路，而且非常容易實踐，未來規劃朝線性代數的矩陣去做聯想及應用。

### 二、 效法「雙人遊戲」的思維模式－從單人遊戲到雙人遊戲的思考

思考是否能使用其它種演算法，譬如蒙地卡羅樹狀搜尋法，並結合以神經網路來做出一個類似的程式，使得2048人工智慧也有不錯的效果。並將單人遊戲的性質變向：假設現在是人機的一對一對戰，參考雙人下棋的對戰思路，探討能否結合一對一對戰想法，跳脫「單人遊戲」的框架。未來只要將2048從單人遊戲跳脫出來，想成是雙人遊戲，並在每一步都預設下一步電腦會故意選一個最不利玩家的地方放置新的數字方塊，以此去做結果的分析，我相信這也可以是一種嶄新的遊戲策略。

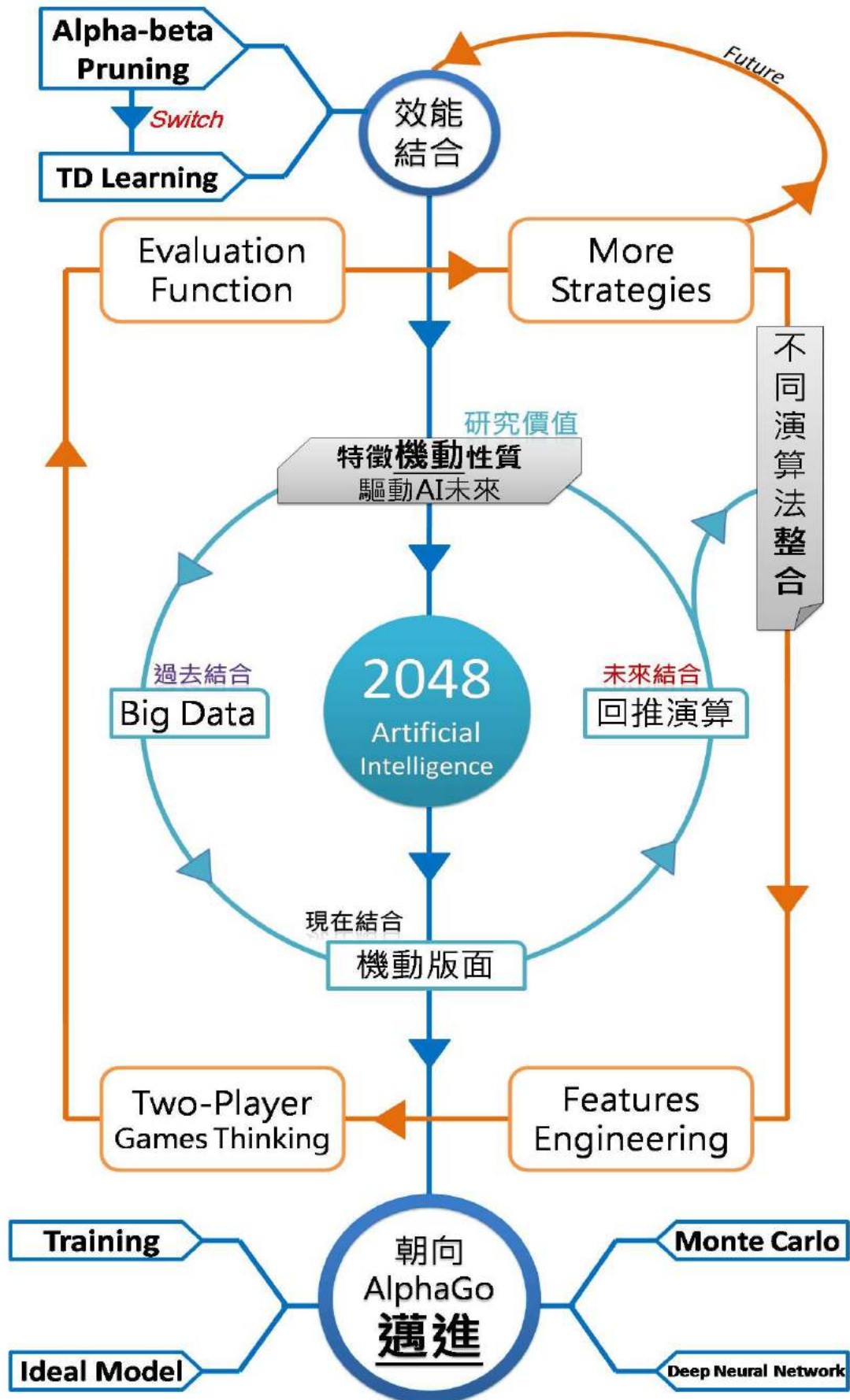
### 三、 寫入「評估函數」(Evaluation Function)－結合人類的遊戲策略

訪問許多周遭玩2048遊戲的玩家，參考大家玩2048遊戲時的頭腦思路，將其策略寫入評估函數當中，歸納出各種多面向的思考方式後一一寫成變數，並透過適當的權重將之整合成一套加權分數的系統。不但不必再訓練數十萬次，還能在每一次決定移動方向前，仔細考慮每一種遊戲策略的變數，成為參考總值的諸多變因。

### 四、 日常生活全方位的應用－將機器學習效率提升大幅推廣

此篇研究的原程式與2048僅僅是機器學習的一個應用，而在這篇研究中也想出了一些能提升的辦法，未來希望能透過類似的手法，利用特徵分析、數據資料，統整出更有效率的學習方法，推廣到日常生活中的其它程式及應用當中。

五、 本研究之未來藍圖：



## 拾、研究進度

2018/09/18	確立研究方向
2018/09/25	開始啟用資訊研究室的 48 台電腦
2018/09/25	開始分析各種特徵的形式(4Tuple)
2018/09/30	試跑 6 組特徵，各測試 50000 次
2018/10/01	空白實驗 – L0126 之 1700000 次試驗
2018/10/02	取 7 組特徵各做 100000 次試驗
2018/10/02	分析、整理出 4Tuple 的 17 種特徵
2018/10/09	發現特徵勝率曲線之收斂現象
2018/10/23	部分 4Tuple 特徵都跑各 100000 次
2018/10/30	部分 4Tuple 特徵都跑各 100000 次
2018/11/20	探究&實際執行 3Tuple 之所有特徵(8 種)
2018/11/20	分析 5Tuple 之所有特徵
2018/12/11	執行 5Tuple 特徵(33 種)
2018/12/25	刪去 5Tuple 中效能明顯不佳者
2018/12/25	分析出 5Tuple 之前六佳的特徵
2019/01/08	持續執行 6 個 5Tuple 特徵 1 萬次
2019/01/08	運用「已得特徵放置方格法」分析
2019/01/08	分析 6Tuple 特徵
2019/01/24	實際執行 6Tuple 特徵
2019/01/28	刪去 6Tuple 中效能明顯不佳者
2019/01/28	分析出最強 5Tuple
2019/01/28	6Tuple 初步性能資料比較(附錄)
2019/01/29	6Tuple 第二次性能資料比較(附錄)
2019/01/30	6Tuple 最終性能資料比較(附錄)
2019/01/30	嘗試與數學性質連結
2019/01/31	持續檢驗最佳的四個 6Tuple 特徵
2019/02/19	發現 6Tuple 漏掉一些特徵未討論
2019/02/20	特徵數量「二倍定律」的發現
2019/02/21	確認 6Tuple 特徵之排序
2019/02/21	著手分析特徵組合之效能
2019/02/22	各取第一/總取前四/原版特徵比較
2019/02/23	持續執行共計 100 萬次
2019/02/24	「機動性質」的設計
2019/02/26	開始設計機動性回推實驗
2019/08/09	測試 1、2Tuple 並證明其無效能
2019/08/10	定義學習效能算法並比較其優劣
2019~Future	To be continued.....

## 拾壹、參考資料及其他

### 一、中文部分

#### 【書籍】

陳德來(2018)。輕鬆學會：運算思維與 C 程式設計。深石出版社

湯曉鷗、陳玉琨(2018)。人工智能基礎(高中版)。華東師範大學出版社

洪維恩(2009)。C 語言教學手冊(第四版)。旗標出版社

#### 【期刊/科展文章】

許馨云、孫瑋彤、翁子恒(2014)。停不下來的 2048。金門第 54 屆中小學科學展覽會

### 二、英文部分

#### 【期刊文章】

Marcin Szubert, Wojciech Jaskowski (2014). Temporal Difference Learning of N-Tuple Networks for the Game 2048.

### 三、網路資源

#### (一)中文部分

#### 【百科全書】

Wikipedia (2019)。2048。Wikipedia Foundation。取自 <https://zh.wikipedia.org/wiki/2048>

#### 【軟體開發網頁】

NCTU。TDL2048-DEMO(2018)。2018/5/12。取自 <https://github.com/moporgic/TDL2048-Demo>

## (二)英文部分

### 【資訊講義】

Stanford University ◦ Chapter 9 : Temporal Difference Learning ◦ 取自

<https://web.stanford.edu/group/pdplab/pdphandbook/handbookch10.html>

### 【期刊/論文】

1. Oka, Kazuto, and Kiminori Matsuzaki. "Systematic selection of N-tuple networks for 2048." International Conference on Computers and Games. Springer, Cham, 2016.
2. Jaśkowski, Wojciech. "Mastering 2048 With Delayed Temporal Coherence Learning, Multistage Weight Promotion, Redundant Encoding, and Carousel Shaping." IEEE Transactions on Games 10.1 (2017): 3-14.
3. Yeh, Kun-Hao, et al. "Multistage temporal difference learning for 2048-like games." IEEE Transactions on Computational Intelligence and AI in Games 9.4 (2016): 369-380.

# 第十八屆旺宏科學獎

## 研究附錄

參賽編號：SA18-065

作品名稱：2048 遊戲的勝率與時間差學

習法程式特徵之分析

姓名：樂峻佑

關鍵字：2048、TD Learning、Pattern

## 附錄一：特徵分析的圖形列表

(表1：3Tuple 特徵列表)

3 Tuple			
L014	L045	L125	L145
L156	L569	直012	直456

(表2：4Tuple 特徵列表)

4 Tuple			
直0123	直4567	田0145	田1256
田56910	L0124	L0126	L4568
L5679	L0456	L1567	Z0156
Z1267	Z45910	Z561011	T0125
T4569			

(表3：5Tuple 特徵列表)

5 Tuple			
Dog610111314	Dog15689	Dog59101213	Dog267910
L1234	L01248	L456812	L567913
L123590	L45678	+14569	Short z 4591011
Short z 01567	Stairs015610	Stairs126711	Stairs56101115
Stairs4591014	T01235	T01259	T456913
T45679	Tall z 015910	Tall z 56101415	Tall z 1261011
Tall z 4591314	Thumb01245	Thumb567910	Thumb12356
Thumb45689	U01246	U02456	U456810
U468910			

(表4：6Tuple 特徴列表)

6 Tuple			
L48121314	L159131415	Thumb159101314	Thumb2610111415
Thumb04891213	F15691314	F267101415	F04581213
U47891011	U81112131415	U4810121314	U15791011
U5911131415	U034567	U0468910	T4567913
T4567910	T0125913	T012359	T012356
Dog269101415	Dog256101415	Dog589131415	Dog5910111213
Dog15891314	Dog14591314	Dog14591011	Dog156789
Z126101415	Z459131415	Z15691014	Z267101115
Z1591314	Z01591011	Z0485913	Stairs15691011
Stairs489121314	Stairs489131415	Stairs459101115	Stairs156101115
Stairs5910131415	Stairs0458910	Stairs04591011	Stairs0156711
Stairs04591014	Q569101415	Q45891314	Q0145910
Q12561011	Medal1256710	Medal4589113	Medal569101114
Medal014569	Cool1567910	Cool489101213	Cool5910111314
Cool45689	Window4568910	Window012456	+1456913
J1236910	J45691213	J567101314	J012589
X689101113	X245679	Duck45791011	Duck013567
Bird459101114	Bird0156710	Y46891013	Y024569



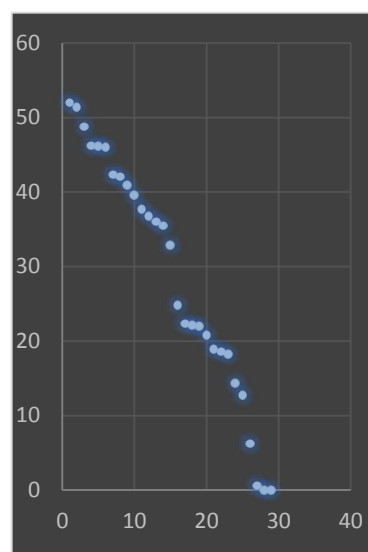
## 附錄二：6Tuple 特徵比較(第一階段)

tuple6 種類	1024 win rate	2048 win rate
J	52	2
stairs0458910	51.41	2
cool	48.77	1.86
Thumb	46.24	4.21
stairs0156711	46.18	1
T0125913	46.05	2.52
dog15891314	42.33	3.22
window	42.06	1.19
U04	40.96	1.29
F	39.56	1
dog14591314	37.73	2.37
z04	36.77	1
T012359	36	2
T012356	35.52	1
Q	32.9	1
duck	24.79	0.28
dog156789	22.3	0.74
stairs04591014	22.1	0.27
medal	22.01	0.52
bird	20.78	0.33
y	18.92	0.26
stairs04591011	18.59	0.36
dog14591011	18.26	0.46
x	14.35	0
+	12.71	0.28
z01591011	6.26	6.26
L	0.57	0
U03	0	0
Z01591314	0	0

藍色區域進入第二階段實驗

錄取門檻:1024 達 46.05%

且 2048 達 1%



附錄 2 到附錄 4 顯示的是一波一波的淘汰工作，因為數據量過於龐大，且預測 Sample 過多，因此每執行數萬次就會淘汰一些效能明顯過低者。

### 附錄三：6Tuple 特徵比較(第二階段)

#### 1024 排名

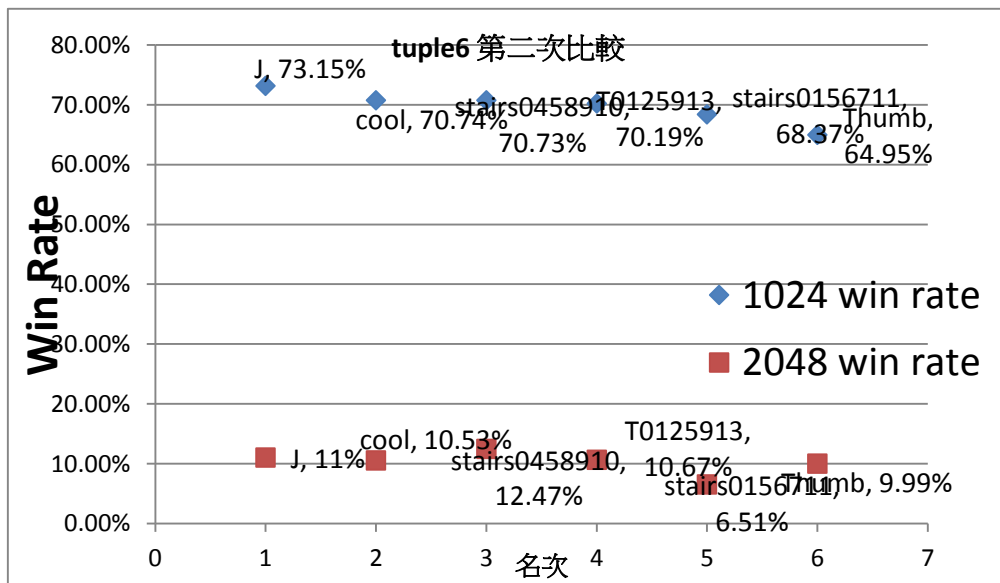
tuple6 種類	1024 win rate	2048 win rate
J	73.15%	11%
cool	70.74%	10.53%
stairs0458910	70.73%	12.47%
T0125913	70.19%	10.67%
stairs0156711	68.37%	6.51%
Thumb	64.95%	9.99%

#### 2048 排名

tuple6 種類	1024 win rate	2048 win rate
stairs0458910	70.73%	12.47%
J	73.15%	11%
T0125913	70.19%	10.67%
cool	70.74%	10.53%
Thumb	64.95%	9.99%
stairs0156711	68.37%	6.51%

結論: stairs01 與 thumb 無論在 1024 與 2048 皆略遜一籌

通過門檻: 1024 >70%      2048>10%



### 附錄四：6Tuple 特徵比較(第三階段)

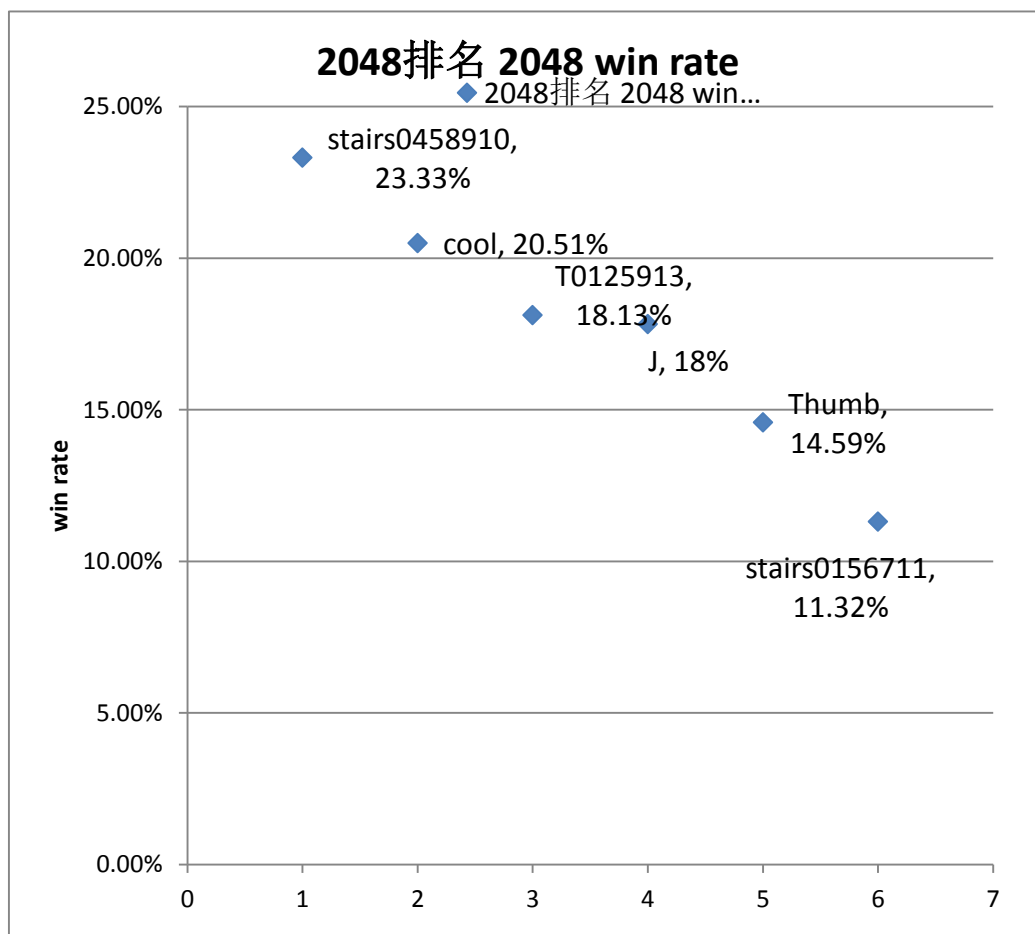
#### 2048 排名

tuple6 種類	2048 win rate
stairs0458910	23.33%
cool	20.51%
T0125913	18.13%
J	18%
Thumb	14.59%
stairs0156711	11.32%

通過門檻: 2048=23.33

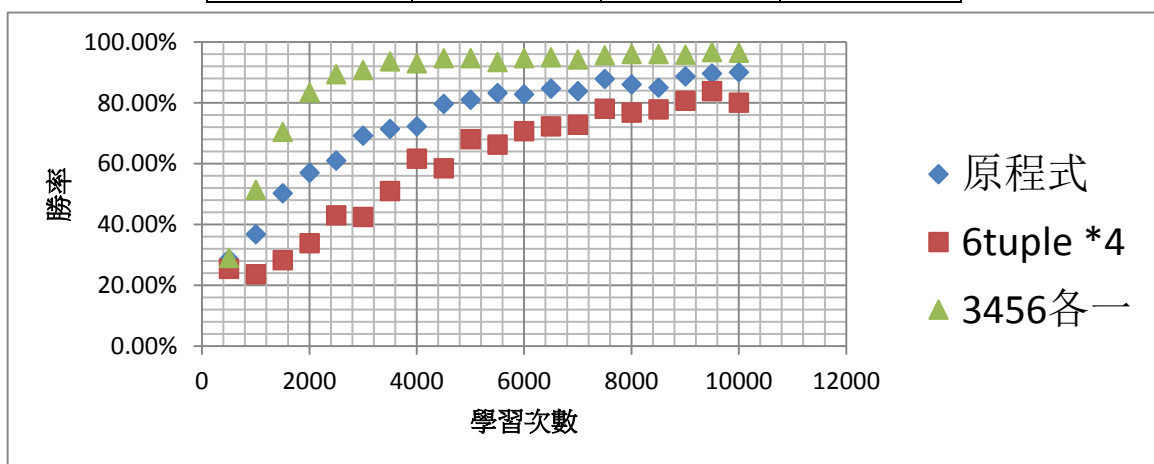
錄取名單: stairs0458910

6tuple 已定案， 20190130 上午 11:30



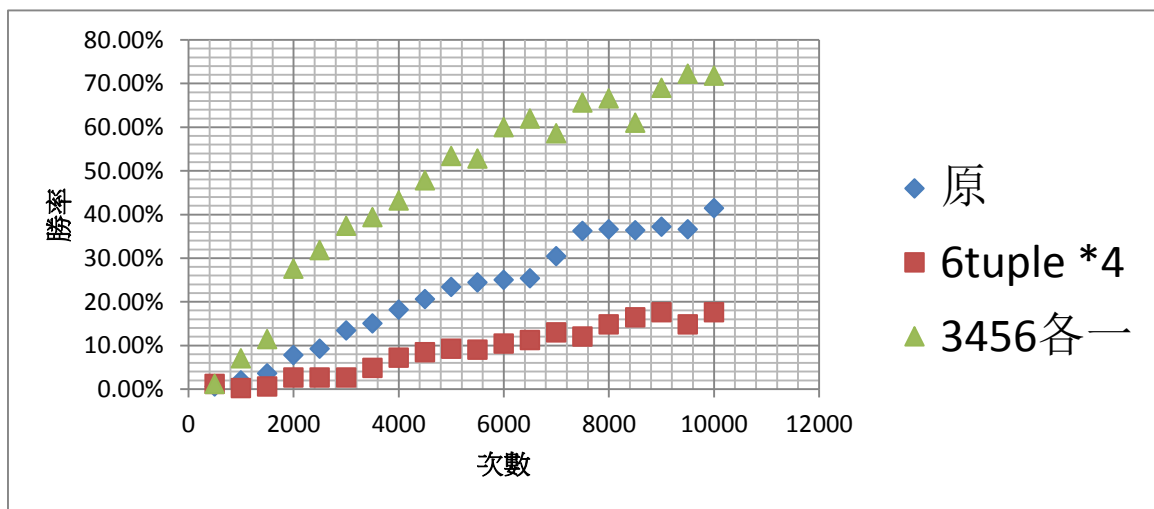
## 附錄五：實驗三前的小測試--初步測試 10000 次

<i>512 勝率</i>	原程式	6tuple *4	3456 各一
500	28.60%	25.40%	29%
1000	36.80%	23.60%	51.40%
1500	50.20%	28.20%	70.40%
2000	57%	33.80%	83.40%
2500	61%	43%	89.40%
3000	69.20%	42.40%	90.80%
3500	71.40%	51%	93.60%
4000	72.20%	61.60%	93%
4500	79.60%	58.40%	94.60%
5000	81%	68%	94.60%
5500	83.20%	66.20%	93.40%
6000	82.80%	70.60%	94.60%
6500	84.60%	72.20%	95%
7000	83.80%	72.80%	94.20%
7500	87.80%	78%	95.60%
8000	86%	76.80%	96%
8500	85%	77.80%	96%
9000	88.60%	80.60%	95.80%
9500	89.60%	83.80%	96.60%
10000	90%	80%	96.40%
<i>平均</i>	<i>73.42%</i>	<i>59.71%</i>	<i>87%</i>



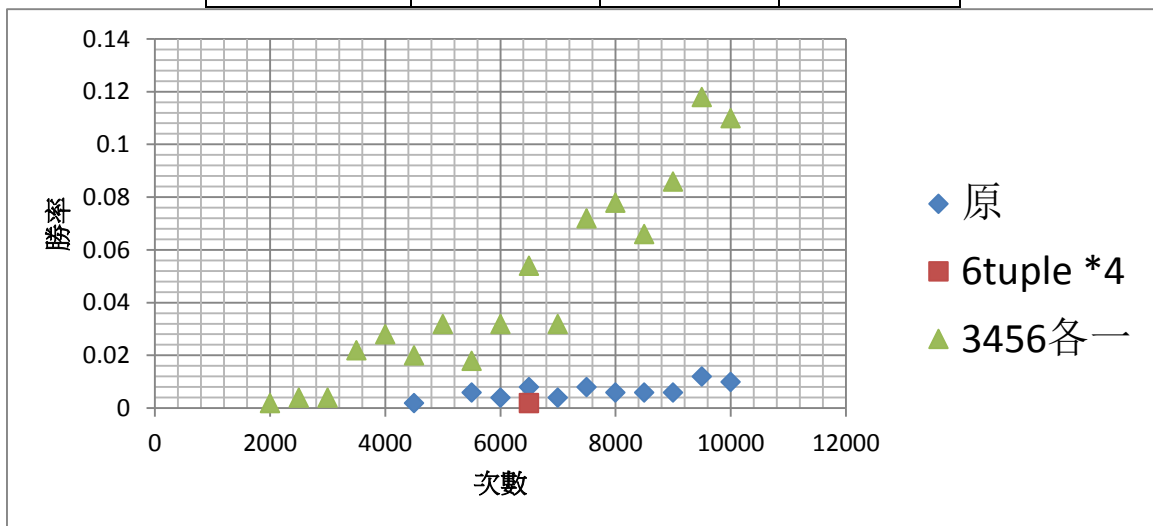
✚ 可發現前 10000 次的遊戲時，各取第一特徵組合就已經展現其極佳的學習效能與成果。

<i>1024 勝率</i>	原	6tuple *4	3456 各一
500	0.60%	1.20%	1.20%
1000	2%	0.20%	7%
1500	3.60%	0.60%	11.40%
2000	7.80%	2.60%	27.60%
2500	9.20%	2.60%	31.80%
3000	13.40%	2.60%	37.40%
3500	15%	4.80%	39.40%
4000	18.20%	7.20%	43.20%
4500	20.60%	8.40%	47.80%
5000	23.40%	9.20%	53.40%
5500	24.40%	9%	52.80%
6000	25%	10.40%	60%
6500	25.40%	11.20%	62%
7000	30.40%	13%	58.60%
7500	36.20%	12%	65.60%
8000	36.60%	14.80%	66.60%
8500	36.40%	16.40%	61%
9000	37.20%	17.60%	69%
9500	36.60%	14.80%	72.20%
10000	41.40%	17.60%	71.80%
<i>平均</i>	<i>22.17%</i>	<i>8.81%</i>	<i>46.99%</i>



✚ 1024 的勝率在經過 10000 次的學習後，各取第一的效能顯然已達原版特徵組合之兩倍以上。

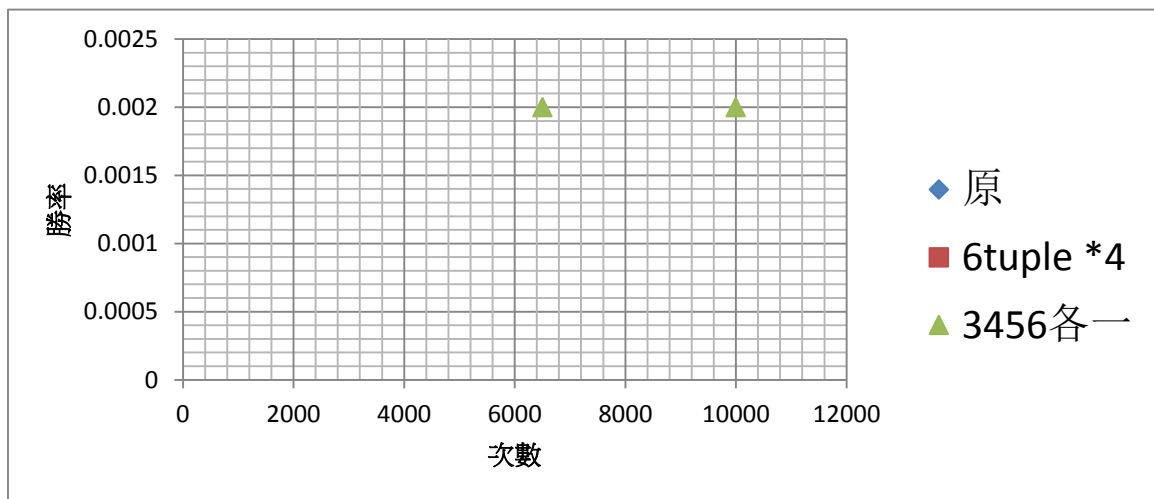
<i>2048 勝率</i>	原	6tuple *4	3456 各一
500			
1000			
1500			
2000			0.20%
2500			0.40%
3000			0.40%
3500			2.20%
4000			2.80%
4500	0.20%		2%
5000			3.20%
5500	0.60%		1.80%
6000	0.40%		3.20%
6500	0.80%	0.20%	5.40%
7000	0.40%		3.20%
7500	0.80%		7.20%
8000	0.60%		7.80%
8500	0.60%		6.60%
9000	0.60%		8.60%
9500	1.20%		11.80%
10000	1%		11%
<i>平均</i>	<i>0.39%</i>	<i>0.01%</i>	<i>3.89%</i>



✚ 看似勝率低，但因為上表執行 10000 次而已，所以造成此現象。

✚ 透過本文的實驗三就可以看到其成果：經過 33 萬次的學習後勝率高達 93%！

<i>4096 勝率</i>	原	6tuple *4	3456 各一
500			
1000			
1500			
2000			
2500			
3000			
3500			
4000			
4500			
5000			
5500			
6000			
6500			0.20%
7000			
7500			
8000			
8500			
9000			
9500			
10000			0.20%
<i>平均</i>	<i>0.00%</i>	<i>0.00%</i>	<i>0.02%</i>



✚ 10000 次學習後的成果只有勝率 0.2%，因為獲勝條件太高，而且學習次數過少，同樣可從實驗四的成果得知經過 32 萬 8500 次即可達到 72.8% 的高勝率！

附錄六：**全取前四**特徵組合勝率-次數列表

由於資料組高達數十萬，故取末 30000 組放在附錄，且每 500 組做平均

執行次數	2048	4096	8192	執行次數	2048	4096	8192
299000	81.40%	28.20%	0.40%	315500	80.20%	32.20%	0.20%
299500	81.80%	29.80%		316000	84.60%	26%	0.20%
300000	83.80%	27.40%	0.40%	316500	83.20%	27.20%	
300500	80.20%	25.20%		317000	82.80%	35%	
301000	80.80%	28.80%		317500	85.20%	28.20%	
301500	83%	31.40%	0.20%	318000	81.60%	32.20%	
302000	84.20%	30.20%		318500	82.80%	29.20%	
302500	85.40%	29%		319000	84%	31.20%	
303000	81.60%	27.80%		319500	82.20%	27.80%	0.40%
303500	85%	27.20%		320000	83%	28%	0.20%
304000	82.40%	27.40%		320500	82%	26.80%	0.40%
304500	84.80%	30%	0.20%	321000	84.80%	31.20%	0.20%
305000	84%	27.40%		321500	80.20%	28%	0.40%
305500	83%	28.40%	0.40%	322000	82.40%	29.80%	
306000	82.80%	29.20%		322500	82.40%	30.80%	
306500	80.40%	29.60%		323000	84.20%	29.40%	0.40%
307000	81.40%	30.20%		323500	85%	30.20%	
307500	83.20%	28.20%	0.20%	324000	83.20%	31.40%	0.20%
308000	85%	31.20%		324500	83.40%	31.80%	
308500	84.20%	31%	0.20%	325000	82.80%	29.80%	
309000	85.60%	26.40%		325500	85.60%	33.60%	
309500	82.20%	33%		326000	81.40%	27.80%	0.20%
310000	82%	28.60%	0.60%	326500	80%	27.40%	
310500	83.20%	29.60%		327000	81.80%	27.20%	0.20%
311000	78.20%	28%		327500	83.40%	31.80%	0.20%
311500	80%	25.40%		328000	81.60%	27.20%	0.20%
312000	86%	32%		328500	80%	27.40%	0.20%
312500	81.20%	30%	0.40%	329000	80.60%	28.40%	
313000	85.20%	30.80%					
313500	82%	28%	0.40%				
314000	83%	28%					
314500	82.60%	28.60%					
315000	80.40%	30.20%	0.20%				

執行次數：32.9 萬次  
 2048：80-85 %  
 4096：27-33 %  
 8192：0-0.4 %



## 附錄七：各取第一特徵組合勝率-次數列表

由於資料組高達數十萬，故取末 30000 組放在附錄，且每 500 組做平均

執行次數	2048	4096	8192	執行次數	2048	4096	8192
299000	90.80%	65.80%	1.80%	315500	90.60%	67%	3%
299500	91.80%	67.20%	2.40%	316000	90.40%	64.80%	3.20%
300000	90.80%	61.60%	2%	316500	90.20%	65.40%	2.60%
300500	92.40%	63%	1.60%	317000	90.80%	70.40%	2.20%
301000	92.80%	67%	2.20%	317500	90.60%	67.80%	2.80%
301500	88.60%	63.40%	2.60%	318000	90.40%	67.80%	3.40%
302000	92.80%	63%	3.20%	318500	91.80%	68.40%	3.60%
302500	90.40%	63.40%	3.40%	319000	89.60%	67.60%	5%
303000	90.80%	62.80%	3.60%	319500	89.40%	65.80%	3.80%
303500	91.60%	65.20%	3.60%	320000	90.20%	67.60%	3.20%
304000	88.20%	62%	3%	320500	91.20%	69%	3.20%
304500	87.20%	59.60%	2.40%	321000	88.20%	67.60%	3.60%
305000	89.80%	66.20%	4%	321500	92.20%	65.20%	3.40%
305500	91.40%	64%	2.60%	322000	89.20%	66.80%	3.60%
306000	91.40%	63.40%	3.80%	322500	91.40%	68.20%	4%
306500	88.20%	64.80%	3.40%	323000	91.20%	66%	4.20%
307000	92.80%	62.40%	2.20%	323500	89.20%	65%	3.80%
307500	90%	63.60%	2.60%	324000	90.20%	70.40%	2.80%
308000	89.80%	61.20%	1.80%	324500	89%	60%	3.40%
308500	88.20%	59.80%	3%	325000	90.60%	68%	4.40%
309000	89.80%	67.20%	2.80%	325500	91%	61.40%	3%
309500	91.60%	63%	4.20%	326000	90.60%	66%	3%
310000	90.60%	64.40%	2.80%	326500	90.60%	64.40%	3.60%
310500	91.80%	64.80%	2.60%	327000	91%	68.60%	4.40%
311000	92.60%	67.20%	2.60%	327500	91.20%	66.80%	2.80%
311500	90.80%	66%	2.80%	328000	91.80%	68.80%	3.40%
312000	90%	65.80%	3.40%	328500	93.80%	72.60%	2.60%
312500	90.60%	61.80%	2.60%	329000	92.20%	67.80%	2.60%
313000	88.20%	61.20%	2.80%	執行次數：32.9 萬次 2048：90-94 % 4096：65-70 % 8192：3-5 %			
313500	92%	66.80%	3%				
314000	88.80%	61.60%	2.80%				
314500	92%	67.80%	1.60%				
315000	91.20%	67%	3.20%				

附錄八：**原版程式**特徵組合勝率-次數列表

由於資料組高達數十萬，故取末 30000 組放在附錄，且每 500 組做平均

執行次數	2048	4096	8192	執行次數	2048	4096	8192
299000	76.60%	24.40%	0.80%	315500	74%	29.40%	0.60%
299500	74%	23.20%	1%	316000	75.60%	29.20%	1.60%
300000	75.40%	26%	0.60%	316500	77%	29.80%	0.60%
300500	75.80%	22.40%	0.60%	317000	76%	26.60%	1%
301000	76.20%	27%	0.60%	317500	74.40%	24%	
301500	77%	27%	0.60%	318000	80.40%	28.40%	0.80%
302000	78.80%	24.60%	0.40%	318500	77.40%	31.20%	0.60%
302500	76.80%	28.20%	1%	319000	77.20%	26%	0.80%
303000	78.60%	28.80%	0.80%	319500	76.20%	25.60%	0.40%
303500	75.40%	24.60%	0.40%	320000	79.40%	30.40%	1.20%
304000	77.40%	26.40%	0.60%	320500	78.20%	29%	0.80%
304500	75.40%	24.20%	0.40%	321000	79.80%	31.60%	0.80%
305000	79.80%	29%	1.40%	321500	75.80%	30.40%	0.80%
305500	75.40%	28.80%	0.60%	322000	75.80%	27.40%	0.60%
306000	76.40%	26.60%	0.60%	322500	81%	31.40%	1%
306500	78%	26.20%	0.40%	323000	81.80%	32.40%	0.80%
307000	77.20%	27.20%	0.20%	323500	79%	30.80%	0.40%
307500	73.60%	25%	0.20%	324000	78.40%	29%	0.40%
308000	76.40%	25.80%	0.80%	324500	79.20%	32.80%	1%
308500	78.40%	26.20%	0.40%	325000	80.80%	32%	1%
309000	76.60%	31.80%	0.60%	325500	75%	30.20%	1%
309500	78.60%	25.80%		326000	77.40%	28.60%	0.40%
310000	75.40%	22.80%	0.40%	326500	77.60%	26.20%	0.20%
310500	76%	28.40%	0.60%	327000	79.80%	29.20%	1.60%
311000	76%	25.20%	0.80%	327500	80.60%	30.40%	0.60%
311500	78.60%	29.40%	1.20%	328000	78.60%	27%	0.60%
312000	77.80%	26.20%	0.60%	328500	77%	31.20%	1.40%
312500	76.60%	29%	0.20%	329000	78%	27.80%	2.40%
313000	76%	27.40%					
313500	78.60%	25.80%	0.60%				
314000	76%	26.60%	1%				
314500	76.60%	25.80%	0.60%				
315000	76.40%	24.20%	0.40%				

執行次數：32.9 萬次  
2048：75-80 %  
4096：25-30 %  
8192：0-2 %



寒假 20190124. LAB.

- 進度 = ① tuple 5 進階.  
② tuple 6 min

寒假 20190128. LAB.

目的: 將沒效能的刪去. (∵已做 60000 組, 不佳者不實驗)

tuple 5 = 剩 T01259. keep going.

目前:

1024	70 ~ 90%
2048	30 ~ 50%
4096	0 ~ 2%

tuple 6: 做 1 個 excel 檢視 29 個之效能

刪去不佳者, 以利專題效率

\*決定優劣: 從 excel 數據中找出末 10000 次取 1024 - 2048 win rate 平均

	(%) 1024	(%) 2048		1024	2048	1024	2048	1024	2048
+	12.71	0.28	duck	24.79	0.28	stairs 0156711	46.18	100	10440.96 1.29
bird	20.98	0.33	F	39.56	1.00	stairs 04589105	51.41	2	103 0 0
cool	48.77	1.86	J	57	2.00	stairs 04591011	18.59	0.36	windonk 2.06 1.19
dog 14591314	57.73	2.37	L	0.57	0.00	T0125913	46.05	2.52	14.35 0
dog 15891314	42.33	3.22	medal	22.01	0.52	T012356	35.52	1.00	18.92 0.26
dog 14591011	18.26	0.46	Q	32.90	1.00	T012359	36.00	2.00	04 36.77 1
dog 156789	22.30	0.74	stairs 04591014	22.10	0.27	Thumb.	46.24	4.21	01591011 6.26 0
									015913140 0

實際執行過程中, 將明顯無效能的特徵刪除自本篇研究中。

20190124

2019 01 29 LAB.

目标: ① 确定 tuple 6.

想法: ① 之後可观察优越的 pattern 存無共通  
② 各 tuple 之 pattern 是否 有数学性

excel 分析.  
发现数据皆相差不远,  
决定明天 0130 keep going.

---

2019 01 30. LAB.

目标: 找出 7 tuple 之所有 pattern.

想法: ① 确定 6 tuple 最佳者 = stairs 0458910

② 7 tuple:  
将 29 种 6 tuple 拿出一一讨论,  
step 1: 在各种地方加上第 7 块,  
step 2: 删去重复者.

引进 organic compounds  
命名原则 avoid repeating.

想法: 将几百种编上编号, 放入 excel  
依大小顺序排序 → 删 repeat.  
用 "0" 以 / 表示. avoid excel 自动省略

expectation = 2019年01月31日 开始实验

✚ 透過數萬組實驗數據的分析，確定 6 Tuple 的最佳特徵：Stairs0458910。

20190129-20190130

(今日日记)

20190224 =

Strange Idea =

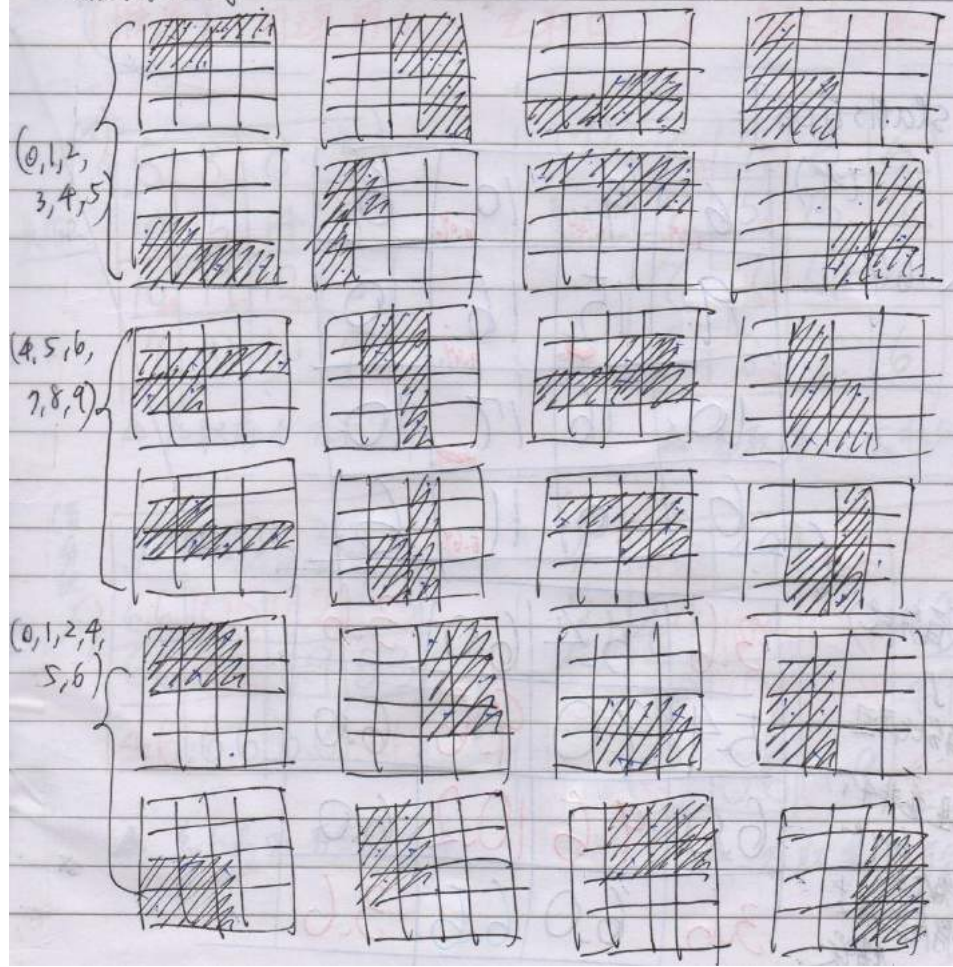
今天我萌生一个想法,在大马路上,大卡车习独及的范围比机车小,社会学上称之为「机动性」,於是将它推广至我的研究裡,称之为「机动性」。

20190223

多个 pattern 在跑时,为了查看 16 格中到底有哪些地方是有被学到的? 到底哪些格被注意的机率大?

方法: 考虑该 4 个 pattern 经过旋转 and 对称后的所有情况 总和即得。

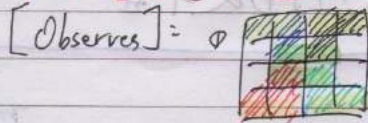
ex. 原程式



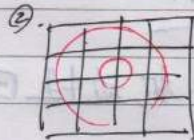
✚ 初步構想機動性質。

20190224

☆重大发现!!!



☆每组的着色块和=绿!  
(以下的推广 prove this is 技巧)



✓内层最多, 往外扩散.

①中间4格交叉相邻 always 同

[推度] 同理用在「全取四」 & 「各取 first place」.

(次) 共152

5.3	8.5	8	5
10.6	16.5	14	8.
10.8	14.2	12.7	8.
7	10.	10.	7.4.5

▲ 总取前4次取分布

(次) 共144

6.2	7.9	6.	6.
7.	15.4	15.	7.
7.	17.1	17.	7.
6.	8.5.6	7.	6.

▲ 各取第一次取分布

(百分比)

3.3	5.3	5.3	3.3
6.6	10.5	9.2	5.3
6.6	9.2	8.9	5.3
4.6	6.6	6.6	4.6

▲ 总取前四百分比分布

(百分比)

4.2	4.9	4.2	4.2
4.9	10.4	10.4	4.9
4.9	11.8	11.8	4.9
4.2	5.6	4.9	4.2

▲ 各取第一百分比分布

✚ 機動性質的設計成熟。

20190226

## 實驗五：機動性回推。

[目的]：透過「歸納」，找出較佳之機動性模板。

[說明]：在「實驗三」中，prove 找到的 pattern 組合勝過原在「實驗四」中，自創一種新特質 → 機動性實現欲自以前之較佳的「單一...」組合，算出機動百分比，→ 找出較佳機動模板，→ 拼出，組成優良模板的 Patten 組合。

② 勝率是否與機動性版面呈「正相關」？！

[預想步驟]：

一、將實驗二之 good pattern 引入實驗四。

二、歸納機動性版面百分比（較佳者）

三、找出嚴謹的「教學方法」→ 使得找出數個 Patten 組合以「拼」出 step 2 之 great 機動性版面

四、以實際執行 prove 此方法之可行性。

✚ 機動性質後續發展/實驗五的實驗設計。

20190417