

國立交通大學

資訊科學與工程研究所

碩士論文

在即時戰略遊戲中探勘與預測玩家的策略風格

Mining and Predicting the Strategy Style of Player  
in Real-Time Strategy Game

研究生：王正宏

指導教授：孫春在 教授

中華民國九十六年六月

# 在即時戰略遊戲中探勘與預測玩家的策略風格

學生：王正宏

指導教授：孫春在 教授

國立交通大學資訊科學與工程研究所

## 中文摘要

如何設計代理人的決策系統，並使其能夠自動學習或發展新的策略一直是人工智慧領域尚待解決的問題。本論文在即時戰略遊戲（Real-Time Strategy Game）的環境下，藉由分析玩家的策略風格以建立具學習能力的決策系統。除了建立具有特殊玩家風格的自動決策系統外，尚能做為假想敵以供對戰演練。即時戰略遊戲的策略考量包括了資源分配、資訊隱藏、善用環境以及策略佈局，這些在遊戲中的考量權衡與真實世界中的戰場、商場、以及球場上所考慮的主要因素是一致的。在即時戰略上設計有計畫與學習能力的決策系統常會遭遇到複雜性的問題，例如決策空間太過龐大，系統不太可能即時下定決策，部分研究者嘗試以人工設計幾種固定策略，再從中採用動態挑選、學習的機制，這種作法雖然成功的降低了策略空間的複雜度，卻又回到了最原始的問題—「電腦的策略（打法）在一定的程度上被預先限制住了」。

人類之所以能夠進步快速的原因就在於懂得善用前人的經驗，生病的時候我們知道要吃什麼藥，而不需要再嘗百草。透過過去發生的記錄，人類可以預知什麼行為可能會帶來什麼結果，我們學會如何去擇優避劣而不需要親自體驗。即時戰略所儲存的記錄檔是許多玩家競賽後留下的寶貴經驗，然而目前的即時戰略人工智慧卻缺乏吸收這些經驗的設計。在這次研究當中我們會收集大量的記錄檔，然後根據我們想要了解的玩家作篩選，篩選的方式是透過玩家的個人習慣，並將篩選出來的資料作為輸入，探勘出玩家在面臨各種狀況下的決策，以瞭解並形成玩家的個人風格。我們所採用的核心方法是案例推論(Case Based Reasoning)，已經有許多研究者很成功的將這套方法運用在各種領域，像是醫學、遊戲、以及機器人的移動上。實驗的最後我們比較決策系統所做的決策，以及記錄檔中玩家的實際決策，來驗證我們的模型。在驗證的過程中我們發現系統很容易達到五成以上的預測準確度，從一場場比賽的預測準確度變化來看，不同的玩家在遊戲中的策略方針也的確有各自的特色。

關鍵字：即時戰略遊戲、策略風格、決策探勘、代理人自動決策系統

# **Mining and Predicting the Strategy Style of Player in Real-Time Strategy Game**

Student: Zheng-Hong Wang

Advisors: Dr. Chuen-Tsai Sun

Institute of Computer Science and Engineering  
National Chiao Tung University

## **Abstract**

The author built a decision-making system with learning ability in real-time strategy(RTS) games. The focus falls on the strategy style of player which is used to build an intelligent opponent agent for practicing.

The factors of strategy considered in RTS games include resource management, information hiding, environment, and strategy priority-arrangements. Similar practical experience and application can be found in battlefield, business or ball game. Most of decision system in real-time strategy games with planning and learning ability cannot avoid uncertain and unpredictable complexity problem— "It cost too much time to make a decision in a complex, real-time environment." Some researchers try to dynamically choose and evolve sub-plans between fixed handmade plans. Although this kind of design was successful in reducing time complexity, it was limited in a sense.

In this thesis, case-based reasoning approach is applied to analyzing gaming experience from player's playing log for building a decision-making system. The end of this thesis verifies the proposed model by comparing the strategy made by our system and real decision made by human players. We found that each player has formed significant personal strategy style which can be found in different game plays.

Keywords: Real-Time, Strategy, RTS, case-based reasoning, decision-making system

# 誌謝

這篇論文的完成要感謝許多人，首先要感謝孫老師在很早以前就看出這個研究方向的價值，而吉隆、建勳以及實驗室的多位學長在研究的路上一直給予我鼓勵、激發我創意，讓我相信自己可以做得到。我還要感謝實驗室的好夥伴們，依文教會我如何使用排版軟體、岳暄也很積極的想要提供有用的意見給我，跟其他要一起畢業的同學互相激勵，才有辦法讓自己不斷的投入自己的心力而不會疲乏。還有很可愛的學弟妹們，跟你們的互動讓我能夠更輕鬆的面對嚴苛的考驗，也祝你們未來的一年也能夠順利通過考驗。我還要感謝我的好朋友，小妞、小鐵、小貓、跟阿多，如果少了你們，我的研究生涯會少掉很多樂趣。最後我要感謝我的家人，你們給予我無後顧之憂的生活環境，讓我能夠自由選擇未來的方向，這對我來說是一切的起源，謝謝你們。



# 目錄

|                                 |     |
|---------------------------------|-----|
| 中文摘要 .....                      | i   |
| Abstract.....                   | ii  |
| 誌謝 .....                        | iii |
| 目錄 .....                        | iv  |
| 表目錄 .....                       | vi  |
| 圖目錄 .....                       | vii |
| 一、緒論 .....                      | 1   |
| 1.1 策略挑選問題與即時戰略遊戲 .....         | 1   |
| 1.2 玩家策略模型 .....                | 2   |
| 1.3 從觀戰中學習 .....                | 2   |
| 1.4 研究目標 .....                  | 3   |
| 1.5 論文架構 .....                  | 3   |
| 二、研究背景 .....                    | 4   |
| 2.1 即時戰略 .....                  | 4   |
| 2.1.1 即時戰略簡介 .....              | 4   |
| 2.1.2 電子競技的發展與遊戲的高度競爭 .....     | 5   |
| 2.1.3 即時戰略的特性 .....             | 6   |
| 2.1.4 策略佈署：科技樹 .....            | 11  |
| 2.1.5 遊戲歷程：記錄檔 .....            | 13  |
| 2.2 即時戰略中的戰術與觀念 .....           | 14  |
| 2.2.1 極端戰術 .....                | 14  |
| 2.2.2 資源採集 .....                | 15  |
| 2.2.3 環境的考量 .....               | 16  |
| 2.2.4 補給 .....                  | 17  |
| 2.2.5 微控 .....                  | 17  |
| 2.2.6 兵力配置、科技走向 .....           | 17  |
| 2.3 人工智慧在即時戰略上的發展 .....         | 18  |
| 2.3.1 發展緩慢的人工智慧 .....           | 18  |
| 2.3.2 支持發展人工智慧的論調 .....         | 18  |
| 2.3.3 即時戰略遊戲中的新一代人工智慧發展目標 ..... | 19  |

|                                     |    |
|-------------------------------------|----|
| 2.4 即時戰略遊戲的 AI 研究平台 .....           | 20 |
| 2.5 案例推論(Case-Based Reasoning)..... | 21 |
| 三、研究設計與實驗步驟 .....                   | 23 |
| 3.1 原始資料的取得 .....                   | 23 |
| 3.2 資料的格式與限制 .....                  | 24 |
| 3.3 實驗模型與架構 .....                   | 27 |
| 3.3.1 玩家動作模型 .....                  | 27 |
| 3.3.2 玩家策略模型 .....                  | 29 |
| 四、實驗與結果 .....                       | 34 |
| 4.1 玩家辨識實驗 .....                    | 34 |
| 4.2 玩家策略的預測 .....                   | 34 |
| 4.2.1 建造策略的母空間 .....                | 34 |
| 4.2.2 建造策略的觀察值 .....                | 35 |
| 4.2.3 玩家辨識模型 VS.人工整理資料 .....        | 37 |
| 4.2.4 盤面判定法的效果檢測 .....              | 38 |
| 4.2.5 玩家的策略集中程度 .....               | 40 |
| 五、結論 .....                          | 44 |
| 參考文獻 .....                          | 46 |

# 表目錄

|                                    |    |
|------------------------------------|----|
| 表 1 輪流制與即時制的差異性 .....              | 6  |
| 表 2 三個種族的策略母空間 .....               | 35 |
| 表 3 玩家辨識實驗結果:.....                 | 38 |
| 表 4 一百場訓練以後的玩家決策空間大小佔母空間比例表。 ..... | 41 |
| 表 5 兩百場訓練以後系統的預測準度變化 .....         | 41 |
| 表 6 三百場訓練以後系統的預測準度變化 .....         | 42 |



# 圖目錄

|   |    |
|---|----|
| 圖 1 即時戰略遊戲「魔獸爭霸」中的戰霧系統。.....  | 5  |
| 圖 2 透過玩家指令估計遊戲世界的狀態。.....   | 9  |
| 圖 3 加入初始狀態的盤面估計法.....   | 10 |
| 圖 4 加入趨勢到盤面估計中。.....  | 11 |
| 圖 5 科技樹、建築狀態與策略圖.....   | 12 |
| 圖 6 星海爭霸(starcraft)中的人類科技樹。.....  | 13 |
| 圖 7 即時戰略「星海爭霸」中的資源收集-採集礦物.....  | 15 |
| 圖 8 「星海爭霸」中四人地圖中的礦物分佈-對稱.....   | 16 |
| 圖 9 案例推論法的步驟。.....  | 22 |
| 圖 10 在公開網站上下載記錄檔.....   | 23 |
| 圖 11 透過 BWchart 來管理遊戲記錄檔( <a href="http://www.bwchart.com">http://www.bwchart.com</a> )。..... | 24 |
| 圖 12 透過 BWchart 檢視單場比賽的玩家雙方手速變化。.....   | 24 |
| 圖 13 透過 BWchart 檢視單場比賽的玩家資源變化。.....   | 25 |
| 圖 14 透過 BWchart 檢視單場比賽玩家的建造與生產。.....  | 25 |
| 圖 15 透過 BWchart 檢視單場比賽玩家的配兵比例。.....   | 26 |
| 圖 16 解碼以後的記錄檔內容。.....   | 27 |
| 圖 17 策略的預測流程表.....  | 33 |
| 圖 18 「星海爭霸」遊戲設定下的所有可能的策略分支度分佈圖.....   | 35 |
| 圖 19 檢驗的變數圖示.....   | 36 |
| 圖 20 玩家辨識實驗流程。.....   | 37 |
| 圖 21 系統成長圖-人類玩家.....  | 38 |
| 圖 22 系統成長圖-蟲族玩家.....  | 39 |
| 圖 23 系統成長圖-神族玩家.....  | 39 |
| 圖 24 蟲族玩家的策略成長圖 (系統搜尋資料).....   | 40 |
| 圖 25 蟲族玩家的策略成長圖 (人工整理資料).....   | 40 |
| 圖 26 玩家決策分支度分佈圖。該統計圖為三百場訓練後的結果.....   | 42 |
| 圖 27 人類玩家 Boxer 對上蟲族對手一百場之後的建築策略圖.....  | 43 |

# 一、緒論

## 1.1 策略挑選問題與即時戰略遊戲

策略挑選問題一直是人工智慧領域中的一個重要主題，尤其是在遊戲領域上的應用，想要設計出能夠自動和玩家互動的角色，需要與領域相關的知識，配合上合適的策略挑選機制[1]。所謂的策略挑選問題，簡單來說就是：“什麼局勢狀況下該做什麼事？”，這是一個普通人每天都要不斷面對的問題。幾點睡覺？幾點起床？什麼時候要寫作業？去哪裡度假？一些日常生活中的決定似乎沒有什麼大不了，但是在軍事上、商業上、醫學上或是其他職業界中的許多決策就不是這樣子了，一個錯誤的決定可能導致公司倒閉、病人因此喪失生命、或是更多無辜的人被戰火波及。

策略挑選問題裡的“動作”，可以是低階的肢體行為，或是較複雜的策略抉擇，在我們挑選的即時戰略遊戲中，同時穿插著低階以及高階的行為，玩家需要不同的決策層次上考量並不斷地切換，遊戲中三個重要的勝負因素—經濟、科技、與部隊彼此搶奪著資源卻又相輔相成，這意味著如果我們要提出一個完整的挑選機制，必須具有階層[2]的概念以及多個衝突子目標的處理能力[3, 4]，在階層的部份此我們先將問題簡單化，僅考慮較高層次的策略佈局，並將子目標衝突的問題，透過玩家經驗的方式來解決。

即時戰略遊戲與實質的軍事競賽非常相近，而打從人類有歷史以來軍事競爭就從未停過，大比例的人力與物力投入到其中，因為一個國家的軍事能力直接的影響到該國的安全以及地位。像孫子兵法[5]、三十六計等各種軍事戰術與觀念從很久以前就被整理成冊，直到今天這些觀念在軍事上、商業上、體育界、或是遊戲競賽中都被廣泛的應用。

電腦遊戲是很合適的 AI 研究平台[6]，遊戲中的世界只會被系統定義好的指令所影響，這意味著我們可以完全掌握遊戲世界的運轉方式。而即時戰略遊戲更是眾多遊戲類型中最具 AI 挑戰性的一項，玩家們必須在即時的环境中互動並在資訊不完整的情形下和對手較勁。由於即時戰略的極高複雜性，目前在遊戲中缺乏學習能力的 AI 很容易被玩家所擊敗，只能靠著一些作弊的手法來獲得與玩家抗衡的優勢，這降低了電腦對手帶來的樂趣以及練習效益，為了讓電腦對手能夠變強、變得更具娛樂性、更有說服力，許多進階的 AI 功能是需要被開發出來的[7]。

## 1.2 玩家策略模型

複雜的策略挑選問題遭遇到的最大困難就是時間與空間的複雜度，我們沒有足夠的時間從所有的動作中挑選最好的動作，尤其當現實生活中的問題都具有時效性，過了這時間點世界又完全不一樣了，就算最好的動作被你計算出來，那也不一定符合現在的狀況。為了降低問題的複雜度，限制可能執行的動作、以及加入一些合理的偏好是最常見的兩種作法。即時戰略遊戲已經幫我們限制了所有可行的動作，也就是遊戲中定義好的指令，在這裡我們要以玩家為單位，利用他們的策略偏好來進一步加速策略的挑選，當問題的選擇不多的時候，系統自然就能夠在有限的時間內挑出一個較好的策略。

《孫子兵法》的《謀攻篇》說到：「知己知彼，百戰不殆；不知彼而知己，一勝一負；不知彼，不知己，每戰必殆。」想要戰勝對手，最好先了解對手的風格與偏好。在體育界，教練與球員常常需要在比賽前研究對手可能派出的陣容，以及每位選手的優缺點；在軍事競賽中，情報員則扮演著相當重要的角色，能夠掌握對手越詳細的資訊，就能夠在戰爭開始之前做好更多的準備。我們提出有鎖定對象功能的策略學習系統，如果鎖定的是我們即將面對的敵手，那系統就可以提供正式競賽之前的假想敵作為訓練，如果我們鎖定的是信任的專家，則系統可以提供新手建議、或輔助老手做決策。

## 1.3 從觀戰中學習

現階段即時戰略遊戲裡頭的人工智慧大多是在遊戲前就制定好的，無論採用的是有限自動機或是配合上比較高層次的動態觸發手稿，其本質還是受限於無法學習。這在遊戲發行的早期來說其實效果還很不錯，有很多玩家也是透過跟電腦的互動、學習電腦的某些策略來進步，但是對一款在大量玩家高度腦力激盪下的遊戲來說，就顯得不太足夠。許多常跟電腦對戰的玩家，害怕在網路上與真人進行對戰，一些網路上的玩家不太願意嘗試新的地圖，最主要的原因就是缺乏適當練習作為信心的後盾，當玩家無法與電腦練習網路上大家常用的打法，那只有透過失敗換取經驗，但是許多對勝負相當在意的玩家並無法接受這樣的結果，如果遊戲本身的重要功能就是娛樂，為什麼玩家必須接受重複不斷的挫折才能學會玩？

為了讓電腦擁有學習的能力，許多研究者著手在一些新的開發平台[8, 9]上研究如何讓電腦在與玩家的對戰中學習[10]，但是大多偏向於讓系統在預定的幾種策略中去做選擇[11]、或是在茫茫大海中探索。然而人類玩家真正厲害的地方，在於可以透過觀戰的方式去發掘對戰雙方的優點與缺點，我們希望我們的決策系統也能做到類似的功能。

目前線上公開的記錄檔資料非常豐富，玩家很願意將自己的遊戲經驗分享出來，會

放出來的記錄檔也大多是非常具有觀賞性與實戰性的部分。而且已經有一些遊戲社群發展出解碼軟體，最初的目的是方便玩家管理、瀏覽大量的記錄檔，但也間接增加記錄檔的可用性，所以我們打算採取不同的方式，妥善地利用這些記錄檔，我們的系統將從觀戰的角度去學習玩家們的決策經驗。

## 1.4 研究目標

本篇論文的研究目標是設計一個決策模型，以觀察者的角度去學習競賽中有用的經驗，我們所採用的平台是即時戰略遊戲，除了遊戲本身方便 AI 實驗的性質外，更因為戰爭中的戰術與觀念掌握大多數競爭型態的考量，我們希望這個建模的流程與想法能夠運用在其他類型的競爭上。

為了達成目標我們必須建立兩個模型，第一個是**玩家辨識模型**、第二個是**玩家的策略預測模型**。玩家辨識模型將幫助我們在茫茫的資料海中鎖定我們感興趣的對象，在網路的世界當中一個玩家擁有的帳號可能有很多個，我們希望能夠找出同一位玩家在不同帳號上玩過的所有遊戲記錄檔，辨識玩家所仰賴的原理是難以變更的個人按鍵使用習慣。而玩家策略的預測模型中，我們會以建築的建造順序作為主要的預測對象，比較我們決策系統所採用的建築順序以及玩家真實狀況下的決策，如果兩者相同表示系統有學習到玩家的策略偏好，這是我們希望達到的，在即時戰略的遊戲中建造屬於較高層次的策略指標，建造的科技樹概念我們可以沿用到其他的領域。

## 1.5 論文架構

本篇論文撰寫的順序如下，第一章為「緒論」，提出本研究的研究方向、研究動機、以及主要的研究目標。第二章為「研究背景」，此章節特別探討與本篇論文相關的背景知識、過去的研究與其主題，其中包含了即時戰略的特性、戰術的簡介、人工智慧在即時戰略遊戲上的發展、以及研究者有興趣的目標，我們也會介紹幾種研究人工智慧所仰賴的即時戰略平台來與我們所提出的方法做對比，並在第二章的最後介紹有關於經驗學習的一些文獻。第三章為「研究設計與實驗步驟」，此章我們將會對實驗的架構，以及步驟的細節加以說明。第四章為「實驗結果」，在此章我們將展示資料的訓練效果，並驗證我們的模型。最後一章是「結論」，在此章我們將以該模型的可能應用做個總結。

## 二、研究背景

### 2.1 即時戰略

#### 2.1.1 即時戰略簡介

在這篇論文中，我們將面對一個高複雜度的遊戲環境—「即時戰略」，玩家必須要在一個「即時」的環境中進行「收集資源」、「建造建築」、「研發科技」、「生產單位」、「控制單位」來戰勝他的對手。

多數的即時戰略遊戲具有「大量物件互動」、「資訊不完全」、「節奏快速」的特性。在傳統的棋賽中，大部分的棋子只有移動跟吃子的功能而已，對戰的雙方都可以看到棋盤上完整的資訊，玩家必須以輪流的方式下棋，勝負僅取決於雙方的腦力。而在即時戰略遊戲中，玩家可以收集資源並生產出許多種類的單位，單位的屬性包含有血量、攻擊力、防禦力、攻擊距離、移動速度、還有許多特殊的功能，更真實的呈現戰爭中的部隊狀況。廣為即時戰略採用的「戰霧系統」(圖 1)則限制玩家所觀察到的視野，玩家只能看到地圖上同一陣營的部隊、建築附近的視野，讓玩家的策略抉擇更加困難。但也因為資訊的不完全，讓玩家必須花很大的力氣進行資訊的蒐集，在大部分的情況下資訊的蒐集往往要犧牲單位，但是高手玩家有越來越捨得犧牲單位換取對手資訊的趨勢。



圖 1 即時戰略遊戲「魔獸爭霸」中的戰霧系統。

隨著玩家探索地圖，所能看到的區域會增加，玩家可以看見的視野被限制於己方單位附近。

## 2.1.2 電子競技的發展與遊戲的高度競爭

隨著時代的進步，遊戲與工作、生活的界線已經模糊化，越來越多的玩家在線上線下過著不同的生活，部分的玩家甚至是活在網路的虛擬世界中。其中有一群玩家我們稱之為重度玩家或是高手玩家[12]，他們沉浸的狀況更為顯著。當重度玩家以類似工作狂的態度進行遊戲的同時，讓人們不禁會想：「那真正以遊戲為職業的玩家，是怎麼在進行遊戲的？」

電子競技(eSports)是指使用電腦遊戲來比賽的體育項目，隨著遊戲對社會與經濟的影響不斷的增加，電子競技在某些國家已經相當成熟。目前推廣職業電子競技的力量主要來自於韓國、大陸、以及歐美，其中又以韓國的成功最為顯著[13]。

即時戰略遊戲一直是電子競技的寵兒之一，最主要的原因來自於他的複雜度逼近真實世界中的戰爭，而觀賞性也不輸給其他類型的競賽。想要成功的職業選手必須每日花上十幾個小時的時間，重複進行同一款遊戲的訓練來保持他的手感、以及對戰局的敏感度。成名的玩家則享有每年十數萬美元的收入，這其中包含工資、獎金、以及廣告。每天晚上的黃金時段，職業比賽的線上轉播吸引了數百萬人收看[14]。

職業化的高度競爭下讓即時戰略不再只是遊戲而已，玩家不只是對電腦的操控的能

力上有進步，也採用了很高層次的戰術。最明顯的就是玩家對於資訊的蒐集以及隱藏的重視程度一直在升高，部分的玩家開始善用假資訊來誘騙對手。

在一些職業聯賽當中，參賽者的分組方式由原本的抽籤改為選手指定，讓比賽在開始之前就已經有了外交的因素在裡頭，玩家不只要考慮如何打贏對手，也要慎選對手，提高自己的勝率或是知名度。《孫子兵法》的《謀攻篇》說到：「上兵伐謀，其次伐交，其次伐兵，其下攻城。」職業化以後，即時戰略從比賽制度到選手戰術的爾虞我詐就像是真正的戰爭。

### 2.1.3 即時戰略的特性

在建立我們的決策系統之前，必須先考慮到即時戰略遊戲與一般決策問題的差異性，讓我們的決策系統在稍做修改後就能轉換到不同的決策問題上。目前市面上的即時戰略遊戲具有的特性包括了「即時」、「重複」、「多人」、「記錄」、「重置」、「平衡」、「封閉」、「資訊不完整」。在這些特性之下，我們要能對玩家過去的策略進行「評估」的動作，因為我們希望系統能夠從過去經驗中成長。

#### (1) 即時

一個決策系統在面臨決策問題時通常可分為兩種方式，第一種是訂好完整的計畫直指全域目標而去，第二種是在面臨問題時馬上挑選一個最佳的區域解答。計畫所要花上的時間與空間複雜度會比較高，所以一些有時間限制上的決策問題會更難以用計畫的方式解決。即時的特性是相對於輪流制（Turn-based）來講的，兩者的比較如(表 1)所示。在回合制的狀況下，我們還有辦法算出幾回合內的所有狀況，但是當問題變成即時了以後，你沒辦法完整地預測到未來可能發生的所有狀況，就連短期的判斷都相對困難。而且即時下的決策的時機是一閃而過的，世界不會靜止在那邊等你做決定，所以大多數的即時問題都是透過刺激反應式的系統來達成，這類的系統最大的困難點在於，如何從區域最佳解邁向全域最佳解，有些研究者就針對計畫與刺激反應在即時問題上做一個融合。

表 1 輪流制與即時制的差異性

|     | 可決策時機 | 決策反應 | 決策中  | 決策重點  |
|-----|-------|------|------|-------|
| 輪流制 | 輪到自己時 | 馬上反應 | 世界靜止 | 動作    |
| 即時制 | 隨時    | 逐漸反應 | 世界變動 | 動作、時機 |

## (2) 重複

人類在某個領域的經驗累積速度，往往與該領域的問題重複性有關，如果同樣的狀況一直重複的出現，經驗就容易累積，也常需要利用這些經驗去解決類似的問題。在即時戰略遊戲中，遊戲是以回合不斷重複的方式在進行著，就遊戲中的建築策略而言玩家容易遭遇到類似的狀況，透過玩家的反覆嘗試與相互討論，建築策略就會逐漸集中並被重複使用，而這些常被玩家所採用的建築策略就是我們希望系統學會的。

對於一些重複性質比較低的領域，就必須考慮到我們要切入的策略層級是否要降低。例如在買賣的商場上，品牌的經營是從公司誕生開始一直延續下去的，但是如果我們切入的層面是以商品為單位，那重複性就會高很多。雖然降低切入的單位可以增加我們的重複性，但也同時降低我們的策略層級，過度降低層級的結果可能會失去焦點，太過注重小細節而忽略大方向的決策系統不是我們要的結果。一個可以同時掌握高層次的目標，又能控制好細節的系統需要階層式的合作。

## (3) 評估

並不是所有的經驗都是成功的，有很多時候我們做了錯誤的決定，導致失敗的結果。比起沒有評估能力的系統，擁有目標的決策系統在挑選策略的時候比較容易，因為它知道哪些決策將帶領她達到想要的結果[15]。評估的精確度可以從二分法到連續函數，後者比前者更能分出策略的優劣。系統評估的對象是定義好的策略，但是一次成功的結果是很多策略合作的結果，這其中還可能包括對手的失誤。如何給定每個子策略一個合理的分數不是一個容易的問題，除非我們可以評估出階段性的成功度。就即時戰略遊戲來說，每一次建立新種類的建築都可以視為一種高層次的策略布局，而我們希望這個策略能夠增加系統的勝率，因為即時戰略遊戲的最終目標在於戰勝對手。

投票是民主制度下的有效手段，在每個人各有意見的狀況下，我們統計出一個最具代表性的結果。在本篇論文中，我們對策略的評估也是採用投票的方式，一場輸掉的比賽中所有的策略我們都給定負一分，勝利的給定一分。而在運用過去經驗的過程中，策略的效能總分由下列公式算出：

$$Performance(Strategy, Game) = \begin{cases} Win : 1 \\ Unknown \& Draw : 0 \\ Lose : -1 \end{cases}$$
$$Score(Strategy) = \sum Performance(Strategy, Past_i) / Distance(Past_i, Now)$$

一個過去決策的效能(Performance)由最後的遊戲勝負決定，勝者得到一分、負者失去一分，平手或是未知給定零分。一場過去比賽(Past<sub>i</sub>)中如果曾經出現跟目前(Now)一樣

決策狀態的話就擁有投票權，投票的分數(Score)是過去的效能(Performance)除以過去跟目前盤面之間的距離(Distance)。系統最後會算出目前可執行決策(Strategy)各自的總分(Score)並採納最高分的決策。

#### (4) 多人

對玩家來說，多人遊戲的意義在於，競爭變得有變化了，你的對手是一個懂得學習的對象，同樣的一種招式不能每次都奏效。但是對於我們的系統來說，多人遊戲的意義在於，我們可以讓系統從玩家的經驗中學到一些東西，可以從勝利的一方學到：「如何給對方致命的一擊」，也可以從失敗的一方學到：「如何從困境中選擇一條比較有機會的路」。而最重要的這是一個有實際意義的學習，因為這些資料就是來自於真實的狀況。

#### (5) 記錄

記錄是比較操作面的問題，越自動、越完整規畫的記錄機制可以替我們省下較多的處理時間，在即時戰略遊戲中我們得到的記錄是玩家執行過的指令，在有些狀況下我們還能得到當時的環境反應，例如病人的病歷表，裡頭儲存的不只是醫生開的處方、還有病人的身體狀況。如果能完整記錄過程中環境的反應，那我們就能夠更準確的評估策略執行的效果，但是事情往往不如人意，沒有選對觀察值、儲存空間的限制、以及一些沒有權限得到的資訊，讓我們的記錄變得局部而破碎。如果記錄的內容多於需要的，我們只需要花些心力去篩選有用的資訊，但是如果記錄內容不夠充分，那系統對於狀況的掌握度就會比較侷限。還有一種情況我們永遠難得到足夠的資訊，像是天氣預測的問題，蝴蝶效應告訴我們一個小小的擾動很有可能經過一定程度的迭代後，造成巨大的影響[16]，在有限的資源下，對於這些問題我們勢必只能做到短期的預測。

(圖 2)中的虛線是我們在只有指令記錄而無當時環境反應記錄的作法，例如我們可以得到玩家建造建築的時間點跟建築種類，有可能建築還沒建好就被對手給打壞了，但是我們假設他下定這個動作之後，建築就會被建立在遊戲世界中。又例如在一場球賽中我們記錄下每位球員傳接球的時間點，那我們就可以估計出團隊的控球率。有很多數值我們都可以由因果關係來猜出來，但是畢竟這是估計值，如果我們有實際的狀況記錄，採用直接觀察的方式比較準確。

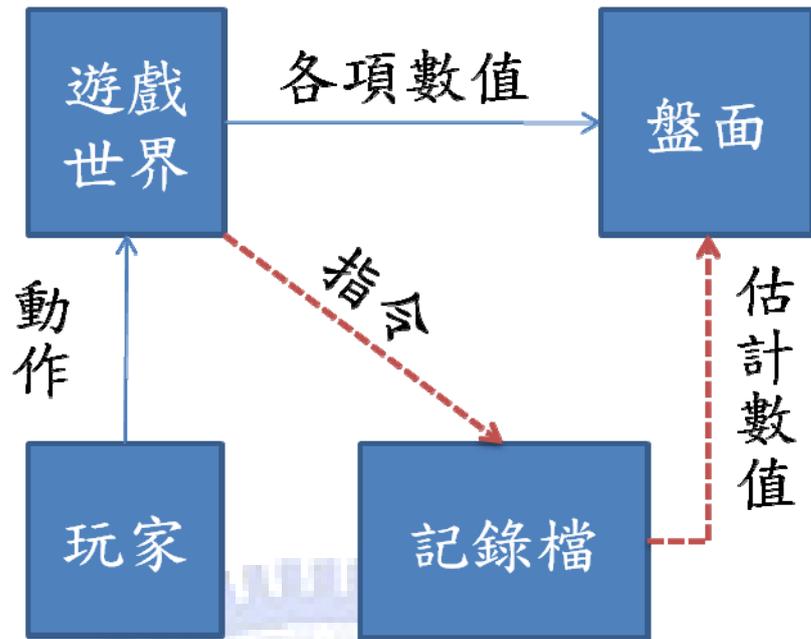


圖 2 透過玩家指令估計遊戲世界的狀態。

在缺少遊戲支援的狀況下，我們透過虛線的方式間接估計出遊戲世界中的各項數值。

## (6) 重置、平衡

在真實世界中，事件之間往往是會互相影響的，前面的成功讓你擁有更多的資源面對之後的挑戰。但對於遊戲來說，公平是一件很重要的問題，玩家希望是因為自己的能力而導致成功或是失敗，而不是因為遊戲偏袒其中的某些團體。因為即時戰略本身需要提供不同種族的差異性，所以平衡通常透過兩種方式來表達，第一種是立足點的平等，每位玩家會在遊戲地一開始被給予相當一致的初始資源；第二個是齊頭式的平等，如果遊戲中的每個種族在平均勝率接近五成的時候，就會被認定有相當的平衡性，平衡的調整通常是修改單位的強度或是花費，是一種嘗試錯誤法。

為了保證立足點的平等不會遭到破壞，所有玩家在一開始擁有一樣的初始資源，而且回合之間是獨立的，前一場比賽中的資源並不會留到下一輪繼續使用。在我們的即時戰略決策模型中，由於玩家的初始狀態都一樣，所以我們只須考慮決策對世界帶來的影響就可以了，但是在現實世界的初始資源往往是不平衡的，為了因應這樣的問題，我們的模型只需要再加上在世界中的初始狀態即可，(圖 3)表示在一個不平衡的初始狀態下我們世界現狀的估計方式。

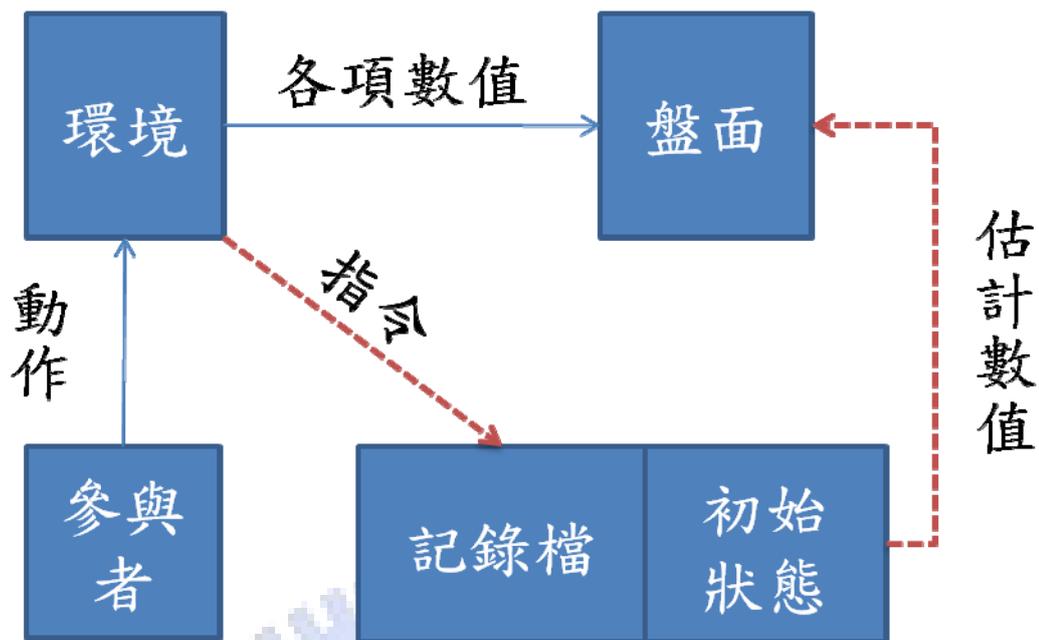


圖 3 加入初始狀態的盘面估計法

## (7) 封閉

封閉性是遊戲與電腦模擬普遍的性質，在這裡我們封閉的定義是世界只對我們可以觀察到的行為做反應，在遊戲中我們可以透過玩家執行過的指令重複模擬同一場比賽，遊戲的記錄檔就是封閉性最好的證明。封閉性對我們系統來說最大的意義在於，我們很容易可以掌握世界運轉的重要因素，在遊戲中我們可以很乾脆的將所有的合法指令、各類統計數字都考慮在內，並保證我們一定會掌握到遊戲世界變化的主要因素。但是許多現實世界中困難的問題就在於，我們往往觀察到的只是很小的一部分，例如在股市這個複雜的領域，你所做的動作其實對世界沒什麼影響力，真正有影響力的是那些具有大量資金可以運作的人、以及多數散戶的預期心理，而大多數人的買賣行為你並沒有權得知。

封閉性所遭遇到的問題與前面討論的記錄完整性相去不遠，在封閉性低的狀況下我們必須盡可能的得到重要因素的記錄，也要考慮是否要將環境的變化趨勢加入到盤面的判斷。(圖 4)表示在不封閉的世界下，我們盘面估計的方式，在圖右下方的趨勢，我們可以很簡單的以過去幾次盤面的變化得到差異值，也可以很複雜的用該領域的專業知識來預測，但是如此一來趨勢本身的問題可能會比我們這整個模型地其他部分加起來還要大，這個部分的討論我們留在未來工作。

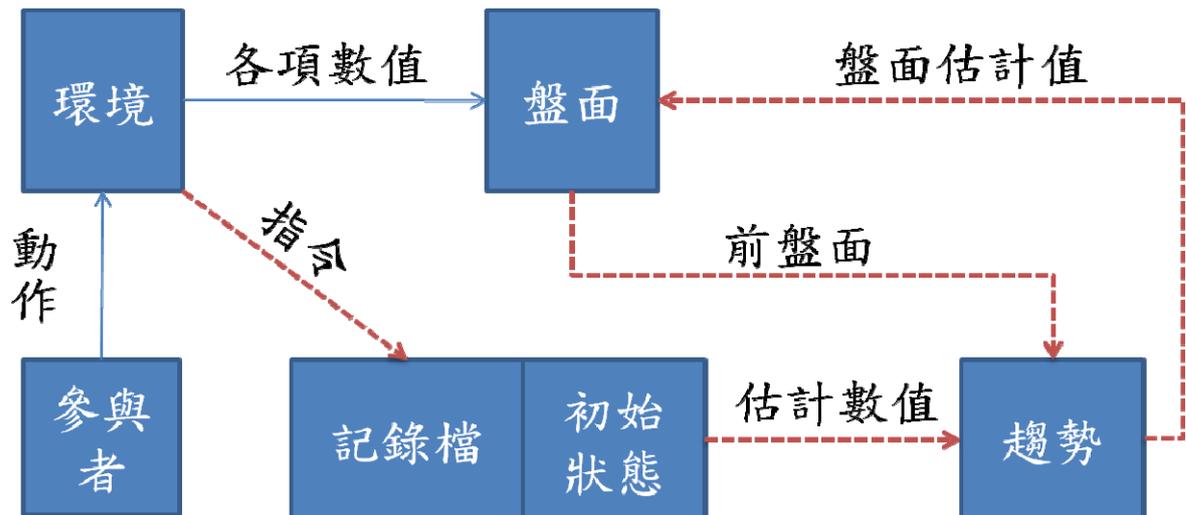


圖 4 加入趨勢到盤面估計中。

如果沒有辦法直接取得環資料，我們透過虛線的路徑去估計出環境中的各項數值。

## (8) 資訊不完整

資訊不完整是多數複雜問題中常要面對的狀況，以人工智慧中常被提到的 Wumpus World 為例，探險家的視野僅止於鄰近的格子，大多數的環境狀況是隱藏起來的直到探險家經過之後才會得到，對於探險家來說這個世界的資訊是不完整的。在多數的即時戰略遊戲中有所謂的戰霧系統，玩家在遊戲的當下並無法得知戰霧中的視野，但是遊戲過後玩家可以得到當時遊戲世界的所有資訊；但是在現實世界中，當下得不到的資訊通常在事後也不會得到。面對資訊不完整的解決方案通常有兩種，第一種是透過作弊，例如遊戲中的電腦是可以知道玩家的所有資訊的，第二種方式是猜測，決策系統必須要能夠猜測出一些可能狀況並觀察事件的發展趨勢來修正自己的猜測方向。

### 2.1.4 策略佈署：科技樹

在現實世界中的軍事佈署牽涉到的範圍非常的廣泛，除了表面上的軍事預算、軍事科技以及人員的佈署以外，還有國與國之間的外交政策、貿易政策。而在即時戰略遊戲中的佈署也是如此，玩家必須採集資源並決定經濟與軍事發展的預算比例，在一些遊戲模式中玩家之間甚至可以反覆結盟或是斷交。

對於即時戰略遊戲來說，建造屬於高層次的策略佈署，收集資源、研發科技、生產單位都仰賴不同的建築。厲害的玩家可以直接由建築的種類、數目來猜測出敵方的戰術全貌，所以玩家對戰時也常常需要隱藏自己的重要建築，避免被對手的偵查單位發現，有些遊戲更直接地將失去所有建築的一方判定為輸家。為了讓遊戲反應科技的相依性，功能類似的建築物有高低級之分，必須先擁有某些低級建築才能進一步建立高級

的建築(圖 5)，科技樹(圖 6)就在這樣的概念下形成。科技樹是典型的導向非循環圖形 DAG(Directed acyclic graph)，在科技樹限制下的一個拓撲排序就是一種決策的順序，在實際世界中的活動選擇大多具有如此的特性，例如我們必須要先買進原料才有辦法製造產品，而我們要有產品才有辦法開始販賣。

科技樹的限制決定了我們決策的複雜度上限，我們可以將合法的狀態數(節點數)，以及連接狀態的邊都計算出來，對我們的決策空間多一分了解。對於即時戰略以外的問題，我們可以著手繪製科技樹，也可以透過現有的決策順序來反推出科技樹，但策略的定義最好由專家來制定，確保我們每項決策能乾淨地被分割而不會遺漏什麼。

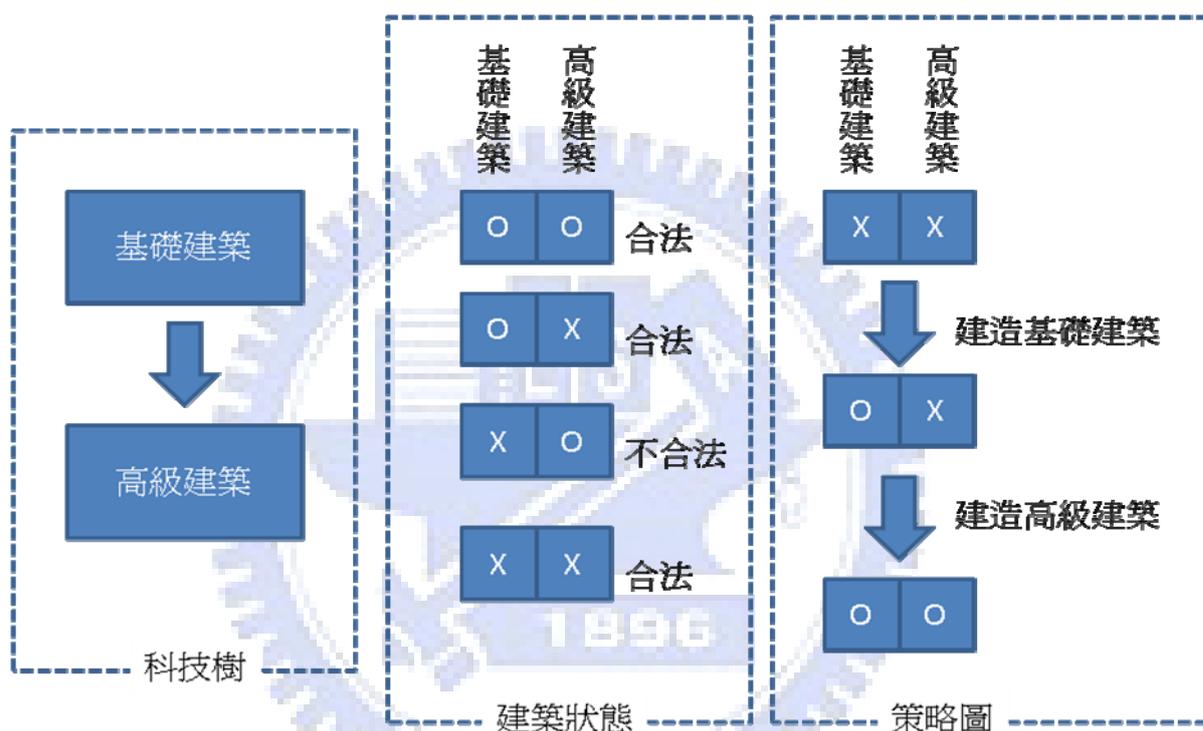


圖 5 科技樹、建築狀態與策略圖

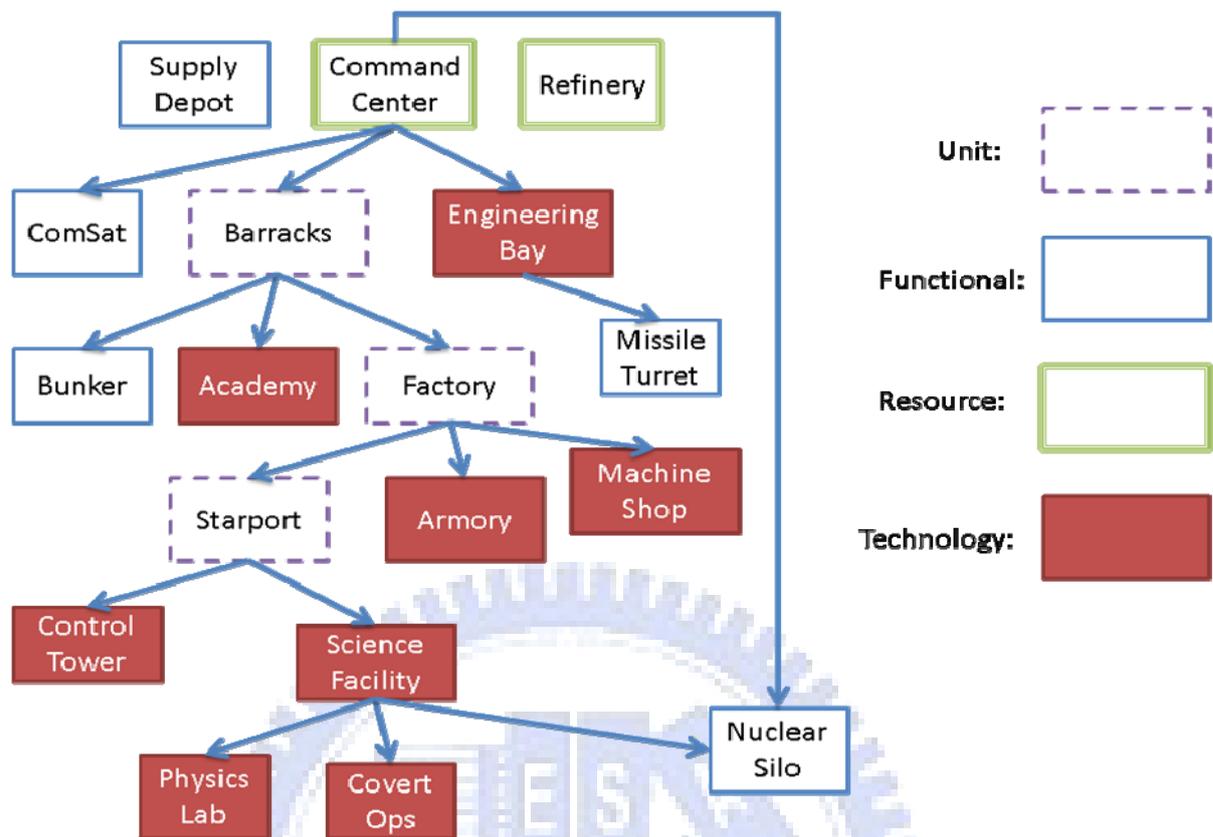


圖 6 星海爭霸(starcraft)中的人類科技樹。

玩家必須先建立箭頭源頭的建築，才能建立箭頭所指的新建築。

## 2.1.5 遊戲歷程：記錄檔

即時戰略提供玩家記錄遊戲歷程的功能，記錄檔內容包括玩家下過的所有定義好的遊戲指令及其執行的時間點，是標準的時間序列資料，資料擁有參與者的意圖在裡面，但是低階的指令要透過該領域的相關知識才有辦法化作有意義的資訊。在即時戰略中，玩家使用鍵盤滑鼠進行的各種動作會被遊戲轉換成模擬器所接受的指令格式，只要將玩家們的指令時間與內容序列化以後傳遞給所有參與遊戲的客端，客端電腦上的模擬器便可以將遊戲世界當時的狀態呈現給玩家。也就是說，只要將所有玩家的指令過程記錄下來，就可以不斷重複模擬整個遊戲進行的狀況。透過記錄檔我們可以得到玩家下過的各種指令，包括建造、生產、研發、控制...等。但是在缺少模擬器的形勢下我們很難得知某個時間點的實際遊戲世界狀況，有些玩家所採取的動作實際上並未被執行，例如我們對已經排滿工作的生產線下達新的生產命令，這樣子的命令會自動被該建築所忽略，可是記錄檔依舊有這一筆生產指令存在。一些篩選的機制將幫助我們盡量的得到玩家成功執行的指令。

## 2.2 即時戰略中的戰術與觀念

### 2.2.1 極端戰術

根據即時戰略中的部隊、經濟、科技大致上可以分為三種極端的戰術，這三方面可說是玩家在爭取勝利時的三個子目標，然而這三個目標彼此又因為爭奪資源而有衝突的狀況，如何在三者中取得平衡點是 AI 在即時戰略上的一個重要議題：「資源管理」。

#### (1) 搶先生產部隊

這類的戰術又稱做快攻，玩家在很早期就生出部隊，於對手還沒有足夠兵力防守的時候給予迎頭痛擊。如同我們在球類運動上常看到的，快攻是一把雙面刃，快攻建立在犧牲重要的實力基礎上，在即時戰略中玩家犧牲掉的是初期的經濟能力，而在球類運動中則是提早犧牲掉運動員的體力，如果快攻沒能奏效，往往都會很快地被對手擊敗。

#### (2) 搶先爭取經濟

和快攻相較之下這類戰術採取截然不同的做法，遊戲初期的時候玩家將資源全部投入經濟的發展上，配合著防守建築還有小部隊的偵查，企圖以後期的優勢經濟能力建造出大部隊擊敗對手。國家之間的戰爭要花掉的金錢是非常可觀的，如果沒有足夠的資源就貿然出擊，很有可能打到一半就因為後勤已經無法支援而必須被迫投降。戰爭失敗所要背負的後果太過於嚴重，所以自古以來國家間大多是先締結同盟，等到經濟發展到足夠規模，才開始有武力擴展國土的想法。

#### (3) 搶先研發科技

科技在遊戲中扮演的是非常微妙的角色，研發科技帶來的效果並非是線性的，往往是在通過一定的門檻下會有爆發性的差異，而搶先研發科技的做法目的就是研發出足以拉開敵我優勢的決定性科技。在這類的打法中，玩家必須在初期建造適當的防禦並隱藏自己的科技走向，然後盡可能地快速發展出致命的單位，打個對手措手不及。搶先科技的打法在初期和搶先經濟的戰術有點相近，都需要建造防禦性的建築，不同的是科技在未超過一定的門檻前，並不會貢獻太多在部隊以及經濟上，如果對手趁機而入、或是發現你科技研發的方向，很有可能功虧一簣。但是一旦對手沒掌握到情報，就有很大的機會給對手致命的一擊，讓他倒地不起。

在二次世界大戰的末期，核彈的研發就是最好的一個例子，核彈的威力相對於傳統的炸彈比起來有著爆炸性的增長，兩顆來自於美國的原子彈成為了壓垮駱駝的最後一根

稻草，逼得日本不得不投降。面對科技戰，資訊的收集變得非常重要，掌握對手所研發的科技重點，才能提早做對策。

## 2.2.2 資源採集

資源可以簡單分為有限的礦產(圖 7)、以及透過栽種的食物、或是國家對人民的稅收。資源的獲得在即時戰略中是非常重要的環，資源收入較少的一方較容易就會被對手所擊敗。考慮到遊戲設計的公平性，有限的礦產資源通常以對稱的方式分布在地圖上(圖 8)，玩家佔領一個資源點並開始採集的過程稱為擴張(Expanding)，擴張的時候玩家勢必要投入資源於經濟發展，這時候武力相對薄弱，佈陣防守是常見的做法。

增加資源收入保證玩家有更大的生產力來與他的對手抗衡，但也同時表示玩家需要更多的部隊來維護它所有資源點的安全，尤其資源越豐富的點，越容易成為對手攻擊的第一目標。在資源不多的地圖上，能夠比對手占領更多的資源點將會增加很多的獲勝機率，而在一些初始區域的資源含量就非常豐富的地圖上，資源對勝負的影響就比較小。



圖 7 即時戰略「星海爭霸」中的資源收集-採集礦物



圖 8 「星海爭霸」中四人地圖中的礦物分佈-對稱

### 2.2.3 環境的考量

一個成功的決策者必須要考量到環境可能帶來的影響，連《孫子兵法》都特地花一章的篇幅介紹各種地形的應對方式。在實際的戰爭中，不變的地形與萬變的氣候可能是一場戰爭勝負的轉捩點，如何利用已知的資訊為我方取得較高的優勢，如何在發生突發狀況的時候能有合適的備用方案，都考驗著決策者的能力。

在即時戰略中也設定了地形與氣候的影響，雖然比起現實世界中的影響來的小，但是能夠妥善運用地形與氣候的玩家能在交戰中獲得不小的優勢。例如在遊戲中的高低地形交戰，在高地的單位往往擁有比較大的視野以及炮火命中率。除此之外，地圖上的狹小通道出口更是兵家必爭之地，如何繞過對方的防守點、直接攻擊對手的弱點，仰賴玩家的場面偵查、判斷能力。在一些地圖中，玩家彼此被分割開來，必須進行障礙物的拆除或是透過空中部隊才有辦法接觸到對方基地，在這種狀況下，玩家需要考量的戰術就截然不同。在氣候方面，比較細膩的即時戰略會依照氣候的不同影響單位的能力參數，例如在大霧下玩家的部隊移動的速度較為緩慢。有的遊戲還帶入了時間的概念，不同的種族在白天與黑夜的不同時段能力有所差異。

## 2.2.4 補給

補給是抗爭持續下去的最基本要素，切斷對手的補給等於切斷了對手持續下去的力量，沒有資金可以運轉的公司無法長久、缺少替補球員的球隊應變力差、沒彈藥沒糧食的士兵無法繼續作戰。

在即時戰略中也有補給的概念，只是為了讓遊戲的過程順暢，補給的方式不像是現實生活中補充彈藥與食物那樣繁複，而是當前線在戰鬥的同時，玩家必須一邊在後方生產新的部隊。如果一次交火後玩家沒有來得及補上兵力，很有可能給對手趁機攻打的機會。有些玩家為了保證他有持續與對方抗衡的兵力，甚至會放任小部隊與敵軍自行交火，將視野回到後方的生產線進行補給兵力的動作。

## 2.2.5 微控

無論有多麼完美的決策，實際的執行能力帶來的影響都是不可忽視的，小小的優勢一點一滴累積起來可能就是一場伯仲之戰的決勝因素。在現實生活中，一個有領導能力的決策者必須要懂得將工作分配給適當的屬下，但有時候為了求勝，決策者必須視情況親自出馬。而在即時戰略中，玩家可以自由扮演各種層級的指揮者。玩家可以只控制整體的策略布局，放任部隊自行交火，也可以親自在前線指揮部隊作戰，所謂的微控就是指後者這種狀況。

在一場即時戰略比賽中，能夠讓觀賞者驚呼的不是出乎意料之外的策略、就是神乎其技的微控。在即時戰略中，系統自動的戰鬥方式往往比不上玩家親自操控，所以想要從中獲得優勢的玩家會親自督戰。例如，當我方遠距部隊遇上對手近距部隊的時候，讓部隊散亂的攻擊是很沒有效率的，善於微控的玩家會將自己的部隊聚集起來，命令他們攻擊同一位敵人。

## 2.2.6 兵力配置、科技走向

一支只有前鋒的足球隊，面對陣容齊全的對手，是很難在比賽中取得勝利的。缺少有經驗的後衛就很難阻止對手拿分，缺少了能夠控球與傳球的中場，前鋒根本沒有機會在前場接到來自隊友的傳球，但是如果對手是支只有後衛的球隊，那又何嘗不可。

在即時戰略中的兵種都有其優缺點，如何配置出最強的組合不是一個容易回答的問題，或者說，在平衡的遊戲世界中這不應該是一個絕對的問題。單純的兵種配置在玩家的控制要求上會比較少，由於兵種都有其各自的特性，有些並不適合放置在同一個隊

伍中，如果玩家帶領一群特性各異的綜合部隊，那就需要更高操作能力才能夠發揮部隊本身應有的水準。

兵力的配置常與科技的走向相關，科技走向相近的兵種比較容易被拿來搭配，因為科技的成長是爆發性的，集中科技的研發往往可以帶來較大的效益。但是集中並不是一個最好的解答，過度集中的科技如果被對手發現，很有可能被針對性打法破解。集中式的配置追求明顯的風格、分散的配置則顯示彈性，如何在這中間拿捏，還需要玩家依照自己的操作能力為基礎做搭配。

## 2.3 人工智慧在即時戰略上的發展

### 2.3.1 發展緩慢的人工智慧

與即時戰略長久的歷史相較之下，人工智慧在即時戰略上的發展其實異常緩慢，主要的幾個原因如下：

- (1) 即時戰略牽涉到大量物件互動、資訊不完全、節奏快速的特性，比傳統的棋賽還要複雜上許多。
- (2) 礙於遊戲開發的時間與資源，相較之下遊戲畫面的投資比較容易收到效果，導致遊戲公司對於這一塊並沒有足夠的動力。
- (3) 網路盛行的狀況下，玩家可以輕易地在網路上找到對手，花費大量功夫去創造世界級的人工智慧似乎顯得多餘。
- (4) 在市面上販售的即時戰略本身並沒有開放原始碼，也沒有提供 AI 研究的界面。在缺乏平台的狀況下，研究自然就比較難進行。

### 2.3.2 支持發展人工智慧的論調

有鑑於即時戰略發展緩慢，加上電腦硬體以及遊戲產業發展快速，一些支持人工智慧在即時戰略上發展的論調也隨之而出：

- (1) 即時戰略雖然複雜度高，但是我們可以將它切成各種層次獨立處理，切入點可以是低階的路徑尋找、或是高階的資源管理。
- (2) 遊戲畫面的加強已經到了飽和的地步，更強的人工智慧將會是一款遊戲與其它遊戲分高下的關鍵。
- (3) 雖然玩家可以上網跟其他人對戰，但是電腦是隨時隨地都可以找到的練習對象，增強電腦的能力必定增加玩家獨自練習的樂趣。

(4) 硬體發展快速，電腦擁有更多的閒置資源可以利用在以往被犧牲的人工智慧上。

### 2.3.3 即時戰略遊戲中的新一代人工智慧發展目標

在傳統的即時戰略遊戲當中所謂的人工智慧可說是由設計者事先訂立好的有限自動機，最大的缺點是玩家可以很容易的找出電腦對手的弱點加以擊敗，如果要增加電腦玩家的強度卻又不想依靠作弊的話，需要發展出像人類玩家這樣能夠學習的系統，才能稱得上是新一代的人工智慧，下面列出重要的子目標：

#### (1) 建立接近人類思考方式的決策系統

在即時戰略如此錯縱複雜的環境下，要讓電腦以低階的指令來看待遊戲中的世界是不合理的，許多單一的動作都對遊戲世界產生局部的影響，現有的人工智慧並無法知道對手攻擊後立刻逃跑的目的在於誘敵而非進攻。在進行各種人工智慧的設計前，我們需要更抽象化、更接近人類思考的方式來判斷各種局面，也需要將抽象化目標與遊戲中定義好的各種指令接軌。如果缺少這樣子的模型我們很難和有學習能力的玩家相抗衡，也很難設計出一個真正的指揮者，設計出一個真正的指揮者才能讓玩家有與真實玩家對戰的感覺。

#### (2) 加入系統的學習能力與敵手的建模

和玩家相比之下，現階段即時戰略中的人工智慧都很缺乏快速學習的能力，玩家往往能夠在少數的幾場比賽中觀察到電腦的弱點，並從中取得優勢。想要電腦擁有類似的功能，我們必須針對對手建立模型，從玩家過去的表現中發掘他的各種偏好，判斷玩家究竟是喜歡快攻還是習慣先搶資源、有沒有特別喜歡使用的部隊。如果能夠抓到玩家的遊戲風格，電腦就可以開始嘗試著針對玩家的弱點作學習、提高玩家與電腦對戰的挑戰性。

#### (3) 時間與空間的認知

在傳統的棋盤賽局中我們很容易地可以掌握到時間與空間的換算關係，棋子在一次的移動內可以行走的所有位置都是可以精確的計算出來，但是即時戰略的環境混合著靜態的地形、與動態的單位，最重要的是動作的執行是需要花費時間的，而且難以估計。一個面對即時環境的決策系統，必須將時間的考量也計算進去，才不會老是在錯誤的時機下定決策。

#### (4) 不完整的資訊

目前的即時戰略大部分都搭配有戰霧系統，玩家可以看見的視野只有自己擁有的單

位附近，在以往的人工智慧設計上為了方便，系統是以視野全開的方式在跟玩家進行對戰，電腦可以得知玩家所有的單位所在的位置，也可以得知玩家所下的每一個指令。這樣的設計可以讓電腦的強度增加，卻也減少了某些樂趣。 玩家之間對戰的時候，彼此之間的偷襲、偷採集資源、或是欺騙戰術都擁有一種害怕被發現的樂趣，但是電腦總是能夠知道你在哪裡你在做什麼，讓玩家只能進行千變一律的正面對戰。 如何在不完整資訊的狀況下讓電腦決定一些假設[17]，例如到哪裡去找對手，攻擊哪些位置，而又不讓電腦變的過度笨拙(i.e.永遠找不到對手)，可以增加遊戲的樂趣。

## (5) 資源管理

在即時戰略的遊戲進行中，玩家必須先採集資源、建造建築、生產單位、升級科技然後才有辦法和對手進行交戰。 玩家時時刻刻都面臨對資源分配的抉擇，何時要將資源投入經濟的發展、是否應該升級哪項科技、或是將資源投入到部隊的生產上。 一位善於管理資源的玩家會被稱作有大局觀的玩家，這在玩家實力的指標中常常被排在第一位，如果系統懂得在正確的時機將資源分配到正確的項目上，那系統的強度將會有很大的增長。

## 2.4 即時戰略遊戲的 AI 研究平台

即時戰略遊戲一直都是極具挑戰性的領域，近幾年來許多獎金比賽吸引了實力深厚的玩家參與。 為了讓系統從這些專家的身上學得一些能力，學者們必須著手讓系統能夠學習，但是大多數的遊戲公司並沒有提供遊戲進行的通訊協定，也沒有提供 AI 設計所需的界面。 所以，有一些專案開始著手創造開放式遊戲平台，目的是提供研究者在上面進行人工智慧的各種實驗。 ORTS 與 TIELT 是其中的兩個比較大的專案[8, 9]，他們定義好各種 AI 實驗必備的平台功能，並增加平台的結構化與彈性，讓跨遊戲甚至是跨類型的狀況下都可以快速地使用。 從他們的功能定義中，我們可以更仔細的去了解，在即時戰略的環境中進行 AI 的研究時，究竟有哪些細節是必須考量到的。

### (1) Open Real-Time Strategy(ORTS):

ORTS(Open Real-Time Strategy) 的概念在 2001 年開始被確定下來，最初的目的是要提供一個可程式界面，供有興趣的人在上面進行 AI 的各項實驗，而遠程的目標則是希望能夠研發出足以擊敗玩家的智慧系統。ORTS 定義了幾種基本的即時戰略元素，包括地形、即時性、動作、群體指令以及隊伍，確保系統能夠掌握到即時戰略的精隨，在上面研發的 AI 機制在運用到其他即時戰略遊戲的時候才有實用性。

除了基本的 RTS 特性外，ORTS 希望能夠增加傳統 RTS 所缺乏的設計，其中最大的改變在於連線的機制與自由軟體的概念。 ORTS 簽署在 GPL(GNU Public License)底下，希

望引入更多有興趣的社群參與設計。而為了杜絕傳統架構下的作弊缺點，ORTS採用了伺服器模擬的功能，也就是說每位客端的玩家只能接收到它應該接收的訊息，無法透過關閉客端模擬器戰霧的方式來作弊。此外，ORTS也加上了一些便於AI設計的功能，其實ORTS本身並不是一款特定的即時戰略遊戲，而是提供整合後的連線機制與圖形化界面，研究者可以依照自己的需要設定環境中的變數，諸如單位的大小、速度或是視野等屬性，而且ORTS定義了很多非常低階的指令，目的就在於提供研究者完整的操控能力，例如研究者可以透過裡面的直線行進指令來實作路徑尋找的功能。

## (2) Testbed for Integrating and Evaluating Learning Techniques(TIELT):

與ORTS相比TIELT(Testbed for Intergrating and Evaluating Learning Techniques)選擇了另外一種做法，它整合現有的一些遊戲，並開放一些設計，讓研究者方便在上面進行機器學習的開發工作。TIELT就跟ORTS一樣提供了系統整合的功能，而且整合的對象橫跨即時戰略、角色扮演、運動遊戲、還有第一人稱射擊遊戲等類型。TIELT將遊戲分成三大塊模型，分別是任務、玩家、與遊戲世界，其目的在於將設計好的決策系統能更便利的抽換在不同的遊戲間。在TIELT中更強制的支援了各種機器學習的方式，包括監督式、非監督式、強化式學習，以及線上、線下兩種學習機制，為了讓線下學習有更優質的資料，記錄檔可以儲存每個周期下的遊戲世界完整資訊。

## 2.5 案例推論(Case-Based Reasoning)

在很多時候我們會重複地遇到類似的事件，一些重要的因素會讓我們想起當初的狀況，如果當初的問題解決的很好，我們就會想透過同樣的方式解決，如果當初產生了很不好的後果，我們就希望這樣子事情不會再發生。案例推論(Case-Based Reasoning)簡單地說就是希望透過過去的經歷，對未來的決策產生幫助[18, 19]。

案例是案例推論中定義的一個專有名詞，一個案例中儲存著一次事件，事件中包含著前因後果，也就是當時的狀況、決策、以及效果。案例推論的流程是一種週而復始的方式，每次透過案例推論我們解決一個事件，接著我們從這個事件中學會一些經驗，將它存入事件資料庫中以便於下次處理新事件的時候能有所幫助。案例推論最大的優點在於它可以持續的學習，每當處理完一次事件以後，該次經驗馬上就可以納入到系統中，這比起每次都要重新計算的類神經網路要來得方便。下列是進行案例推論方法的四個步驟，(圖 9)詳細的描述了前面兩個步驟:

- (1) 從過去的事件中挑選狀況最相近的幾個例子
- (2) 從該群例子中獲取當時的解決方法，挑選好的策略運用到這次問題中。
- (3) 如果事件成功被解決了，將它的良好表現回報給資料庫，如果失敗了，就必須用到該問題的領域知識來做修改

(4) 無論是失敗或是成功解決的例子都會被測量它的效果，然後以適當的方式放入資料庫中

案例推論很適合拿來解決以往透過專家系統解決的問題，與專家系統最大的差異在於，案例推論改由從大量的資料中自動學習而非專家直接提供詳細的解決方法，這省去人工手動設定參數的困擾並增加系統自動學習的能力。目前已經很成功的運用在原本專家系統善於發揮的領域如醫學診斷[20]，也涉足傳統專家系統發揮上有困難的領域，像是機器人在真實空間中移動的問題[21]、以及與本篇論文有關的即時戰略策略挑選問題[22, 23]。

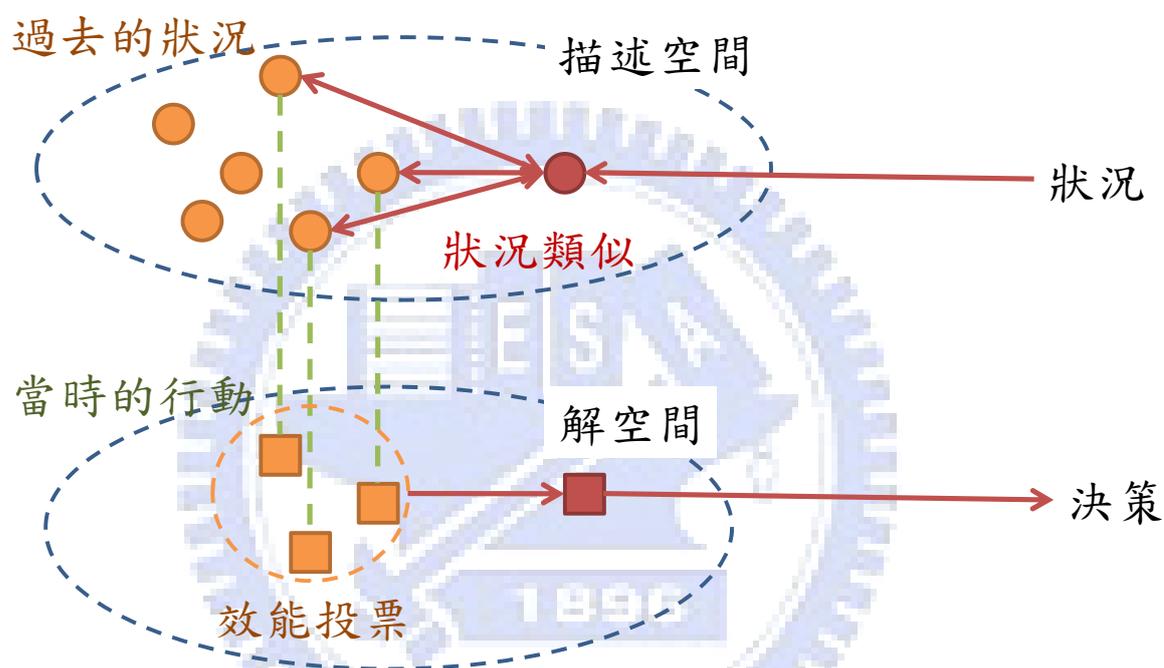


圖 9 案例推論法的步驟。

圓形實心的圖案是過去的一個狀況描述點，而每個狀況點都會對應到一個當初的解，我們以方形實心的圖案表達一個可能的解，兩個空間的上下虛線表示「狀況—解」的一個連結關係。新的狀況出現時，系統便會沿著實線，先在描述空間找出狀況相近的點，再由那些點對應的解票選出一個決策。

# 三、研究設計與實驗步驟

## 3.1 原始資料的取得

記錄檔在許多公開網站上都可以自由下載，(圖 10)是我們搜尋玩家"July"後得到的資料，透過簡單的點擊連結就可以將記錄檔抓下來，許多著名的遊戲網站利用人力的方式將記錄檔分類，有興趣的瀏覽者可以在網站上自由下載、留下評語。有些網站具有搜尋的功能，使用者可以透過玩家帳號、日期、或是比賽名稱進行搜尋，部分網站會對玩家做特別的處理，只要搜尋其中一個帳號，該位玩家使用其它帳號進行的比賽也會一併被找出來。我們將單一網站的七千多筆資料先全部下載，不預先分類，因為玩家的總數過多，我們沒有辦法仔細確認每位玩家擁有哪些帳號，隨後我們會提出一套比較準確而方便的方法來辨識玩家。

**搜尋條件**

GosuGamers \ Replays \ Filter replays

Player one: July, Race: Zerg

Player two: , Race: Any race

Map: , Event: , Filter

**搜尋結果**

| Com  | Player one    | Map            | Event             | Rate          | 下載連結 |      |            |
|------|---------------|----------------|-------------------|---------------|------|------|------------|
| (10) | Legendary     | JulyZerg       | Ride of Valkyries | Public game   | 7.2  | 2609 | 2006-12-25 |
| (12) | POS_July      | Casy[Alive]    | Forté             | Public game   | 9.4  | 2397 | 2006-11-19 |
| (06) | iloveoov      | Julyzerg       | Gaia              | WCG 2006      | 7.6  | 4855 | 2006-11-06 |
| (04) | iloveoov      | Julyzerg       | Azalea            | WCG 2006      | 6.6  | 4024 | 2006-11-06 |
| (06) | iloveoov      | Julyzerg       | Paranoid Android  | WCG 2006      | 8.6  | 4951 | 2006-11-06 |
| (04) | Lx            | July           | Paranoid Android  | WCG Monza     | 8.6  | 1469 | 2006-11-03 |
| (21) | Julyzerg      | jjando_V_      | Rush Hour         | Public game   | 8.1  | 1868 | 2006-10-25 |
| (18) | [Oops]Reach   | MBC_July       | Gaia              | WCG Korea     | 9.7  | 3811 | 2006-08-11 |
| (24) | JulyZerg      | In_Dove        | Gaia              | WCG Korea     | 9.9  | 3898 | 2006-08-07 |
| (10) | JulyZerg      | In_Dove        | Paranoid Android  | WCG Korea     | 9.7  | 2588 | 2006-08-07 |
| (09) | Illoveoov     | JulyZerg       | Paranoid Android  | WCG Korea     | 7.8  | 2891 | 2006-08-07 |
| (07) | Illoveoov     | JulyZerg       | Azalea            | WCG Korea     | 9.0  | 2519 | 2006-08-07 |
| (08) | Illoveoov     | JulyZerg       | Gaia              | WCG Korea     | 9.9  | 3650 | 2006-08-07 |
| (20) | JulyZerg      | Suzuki         | Luna              | Friendly game | 5.8  | 1669 | 2006-04-11 |
| (43) | July          | Siz)o_Ov       | Luna              | Jo pd show .. | 8.6  | 3112 | 2006-04-02 |
| (39) | Julyzerg      | GooDFrienD     | Forté             | Ever Final    | 10.0 | 4518 | 2006-03-24 |
| (97) | Julyzerg      | SONJJANG       | Bifrost3          | Friendly      | 7.6  | 2783 | 2006-02-21 |
| (35) | JulyZerg      | Sasin)Ryoko    | The Lost Temple   | Public game   | 8.5  | 3557 | 2006-01-29 |
| (23) | JulyZerg      | HwaSin         | ROV               | Public game   | 8.3  | 2474 | 2006-01-26 |
| (14) | KyO           | POS_July       | The Lost Temple   | Public game   | 4.0  | 632  | 2005-07-28 |
| (25) | CaSiNo[100su] | JulyZerg*      | The Lost Temple   | Public game   | 9.0  | 2536 | 2005-07-20 |
| (09) | kingdom       | Julyzerg       | Dahlia of Jungle  |               | 8.3  | 1707 | 2005-05-12 |
| (09) | JulyZerg      | Zelotito       | Ash Rose          | MMI           | 8.3  | 1787 | 2005-02-28 |
| (07) | JulyZerg      | ToT)Mondragon( | The Lost Temple   | Public game   | 9.2  | 5128 | 2004-07-19 |
| (02) | JulyZerg      | Mondragon      | The Lost Temple   | Public game   | 4.5  | 2306 | 2004-07-18 |

圖 10 在公開網站上下載記錄檔

## 3.2 資料的格式與限制

記錄檔的資料格式為編碼後的二進位檔，近來有些遊戲已經有玩家社群發展解碼的工具，目的是方便玩家管理、檢視自己手上擁有的記錄檔(圖 11)。透過這些工具玩家可以看到記錄檔中的各種統計數據，包括玩家的手速(圖 12)、資源的變化(圖 13)、建造的順序(圖 14)、兵種的配置比例(圖 15)等。

| Replay name                 | Player 1               | APM 1 | Player 2              | APM 2 | Map                       | Duration | Type | RWA Author | Game Date   | Engine  | Comment |
|-----------------------------|------------------------|-------|-----------------------|-------|---------------------------|----------|------|------------|-------------|---------|---------|
| 16.rep                      | oniespac (Z)           | 61    | Auruxa Tribe (P)      | 0     | Blow Night in Seoul...    | 00:15:47 | lv1  |            | 31 Mar 2007 | BW 1.13 |         |
| 16c.rep                     | plyceus (P)            | 91    | Yenatu Tribe (P)      | 0     | Blow Night in Seoul...    | 00:42:26 | lv1  |            | 07 Apr 2007 | BW 1.13 |         |
| 1977.COM...H11.rep          | DA.A94 (P)             | 84    | Baelrog Brood (Z)     | 0     | Blow Night in Seoul...    | 00:52:19 | lv1  |            | 31 Mar 2007 | BW 1.13 |         |
| 1977.rep                    | plyceus (P)            | 100   | Renus Brood (Z)       | 0     | Blow Night in Seoul...    | 01:31:58 | lv1  |            | 07 Apr 2007 | BW 1.13 |         |
| matas vs sZkat.rep          | midse[gm] (T)          | 312   | Advokate[S2] (T)      | 251   | Korhal                    | 00:08:43 | lv1  |            | 08 Oct 2004 | BW 1.11 |         |
| matas vs Azorpolimask.rep   | acropolimask (Z)       | 152   | Leads[gm] (T)         | 135   | Quillstone                | 00:08:36 | lv1  |            | 29 May 2003 | BW 1.10 |         |
| matas vs ALFA.rep           | midse[gm] (T)          | 281   | ALFA (P)              | 239   | Dahlia                    | 00:13:42 | lv1  |            | 08 Oct 2004 | BW 1.11 |         |
| matas vs Andromé 2.rep      | midse[gm] (T)          | 303   | PFC Andromé[S2] (...) | 264   | Dahlia                    | 00:07:42 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs Andromé.rep        | midse[gm] (T)          | 329   | PFC Andromé[S2] (...) | 215   | Marban_Cross              | 00:10:54 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs Anomak.rep         | midse[gm] (T)          | 263   | Anomak (Z)            | 209   | Rite of Valkyries 1.0     | 00:20:01 | lv1  |            | 14 Apr 2005 | BW 1.12 |         |
| matas vs best 1.rep         | DDanS[gm] (T)          | 342   | SKTelecomT1_Mida...   | 305   | Paranoid Android 1.0      | 00:05:46 | lv1  |            | 05 Aug 2006 | BW 1.14 |         |
| matas vs best 2.rep         | midse[gm] (T)          | 293   | SKTelecomT1_Mida...   | 289   | Azales 1.0                | 00:20:10 | lv1  |            | 05 Aug 2006 | BW 1.14 |         |
| matas vs EltonS2.rep        | sun-ahoc (T)           | 272   | EltonS2 (P)           | 275   | Lost Temple               | 00:38:47 | lv1  |            | 20 Oct 2003 | BW 1.10 |         |
| matas vs easy 1.rep         | Spikyke Really[Aliv... | 395   | SKTelecomT1_Mida...   | 285   | Paranoid Android 1.0      | 00:18:00 | lv1  |            | 05 Aug 2006 | BW 1.14 |         |
| matas vs Center[Name] 2.rep | cuteb[gm] (T)          | 326   | HyoJun[Name] (T)      | 252   | Gasma Gowon               | 00:10:38 | lv1  |            | 30 Jan 2004 | BW 1.10 |         |
| matas vs Center[Name].rep   | DDanS[gm] (T)          | 265   | SKTelecomT1_Mida...   | 171   | Asales 1.0                | 00:06:18 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs Choc-v-a.rep       | DDanS[gm] (T)          | 267   | Choc-v-a[G] (T)       | 246   | Lost Temple               | 00:43:27 | lv1  |            | 04 Jun 2003 | BW 1.10 |         |
| matas vs chojin 1.rep       | CHoJa (Z)              | 357   | SKTelecomT1_Mida...   | 346   | Asales 1.0                | 00:06:18 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs chojin 2.rep       | CHoJa (Z)              | 334   | SKTelecomT1_Mida...   | 336   | Paranoid Android 1.0      | 00:05:46 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs CuteAngel.rep      | PFCross[Name] (T)      | 291   | midse[gm] (T)         | 268   | Gasma Gowon               | 00:33:18 | lv1  |            | 30 Nov 2003 | BW 1.10 |         |
| matas vs dsk[Aliv].rep      | Dsk[Aliv] (P)          | 159   | midse[gm] (T)         | 172   | PST - New Power 2.1 [...] | 00:52:34 | lv1  |            | 03 Nov 2005 | BW 1.13 |         |
| matas vs FishYe 2.rep       | midse[gm] (T)          | 265   | FishYe (P)            | 157   | Korhal                    | 00:17:13 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs FishYe.rep         | midse[gm] (T)          | 259   | FishYe (P)            | 157   | Dahlia                    | 00:12:49 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs fOru 1.rep         | midse[gm] (T)          | 298   | fOru (P)              | 286   | Korhal                    | 00:15:56 | lv1  |            | 15 Aug 2004 | BW 1.11 |         |
| matas vs fOru 2.rep         | midse[gm] (T)          | 334   | fOru (P)              | 275   | Marban_Cross              | 00:10:41 | lv1  |            | 15 Aug 2004 | BW 1.11 |         |
| matas vs fOru 3.rep         | midse[gm] (T)          | 305   | bobae81_kr (P)        | 259   | Dahlia                    | 00:22:48 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs fOru 4.rep         | midse[gm] (T)          | 296   | bobae81_kr (P)        | 280   | Korhal                    | 00:15:12 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs GoJD_Cat.rep       | DDanS[gm] (T)          | 269   | OJO_Ori (T)           | 197   | Nostalgia                 | 00:15:59 | lv1  |            | 22 May 2003 | BW 1.10 |         |
| matas vs GooD[Name].rep     | 2jeon song weak (T)    | 309   | 11 Lee Byoung Min (T) | 290   | Gasma Gowon               | 00:19:16 | lv1  |            | 12 Oct 2003 | BW 1.10 |         |
| matas vs Hermito.rep        | DDanS[gm] (T)          | 299   | HERMITO[AnC] (P)      | 246   | Lost Temple               | 00:17:39 | lv1  |            | 22 Apr 2003 | BW 1.10 |         |
| matas vs Juvav.rep          | midse[gm] (T)          | 295   | IpSe_Foverer (Z)      | 268   | Dahlia                    | 00:22:17 | lv1  |            | 13 Aug 2004 | BW 1.11 |         |
| matas vs KMedusa.rep        | DDanS[gm] (T)          | 258   | GedJ_Medusa (Z)       | 171   | Lost Temple               | 00:28:44 | lv1  |            | 24 May 2003 | BW 1.10 |         |
| matas vs Mex 2.rep          | midse[gm] (T)          | 259   | Med31384 (P)          | 186   | Korhal                    | 00:10:00 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs Mex.rep            | midse[gm] (T)          | 274   | Med31384 (P)          | 198   | Dahlia                    | 00:16:15 | lv1  |            | 10 Oct 2004 | BW 1.11 |         |
| matas vs Nal_beke 2.rep     | ekn[gm] (T)            | 298   | Nal_beke (Z)          | 214   | The Lost Temple_Nal       | 00:35:28 | lv1  |            | 02 Nov 2003 | BW 1.10 |         |
| matas vs Nal_beke.rep       | midse[gm] (T)          | 294   | Nal_beke (Z)          | 209   | The Lost Temple_Nal       | 00:21:02 | lv1  |            | 02 Nov 2003 | BW 1.10 |         |
| matas vs Nal_rA.rep         | ball[gm] (T)           | 308   | O_O (T)               | 187   | Lost Temple (Game)...     | 00:12:37 | lv1  |            | 24 Mar 2004 | BW 1.10 |         |
| matas vs ODi.rep            | TheMasS (T)            | 147   | ODi (Z)               | 129   | PST - Lines The Final...  | 00:20:40 | lv1  |            | 20 Oct 2005 | BW 1.13 |         |
| matas vs oov 1.rep          | SKTelecomT1_Mida...    | 287   | Iloveoov (T)          | 284   | Azales 1.0                | 00:07:50 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs oov 2.rep          | SKTelecomT1_Mida...    | 283   | Iloveoov (T)          | 238   | Paranoid Android 1.0      | 00:19:57 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs oov 3.rep          | SKTelecomT1_Mida...    | 275   | Iloveoov (T)          | 231   | Gama 1.1                  | 00:17:49 | lv1  |            | 06 Aug 2006 | BW 1.14 |         |
| matas vs Pi 2.rep           | midse[gm] (T)          | 286   | fiw_pigpen (P)        | 154   | Lost Temple               | 00:20:41 | lv1  |            | 20 Oct 2003 | BW 1.10 |         |
| matas vs Pi.rep             | DDanS[gm] (T)          | 300   | LiaV[gm] (T)          | 167   | Lost Temple               | 00:19:36 | lv1  |            | 27 May 2003 | BW 1.10 |         |
| matas vs PuSan 2.rep        | DDanS[gm] (T)          | 300   | PuSanS[G] (P)         | 269   | EG Lost Temple OE         | 00:16:08 | lv1  |            | 26 May 2003 | BW 1.10 |         |
| matas vs PuSan 3.rep        | DDanS[gm] (T)          | 297   | PuSanS[G] (P)         | 267   | EG Lost Temple OB         | 00:20:11 | lv1  |            | 26 May 2003 | BW 1.10 |         |
| matas vs PuSan 4.rep        | ddan[gm] (T)           | 308   | PuSanS[G] (P)         | 251   | Lost Temple               | 00:18:48 | lv1  |            | 05 Apr 2003 | BW 1.10 |         |
| matas vs PuSan 5.rep        | DDanS[gm] (T)          | 291   | PuSanS[G] (P)         | 236   | EG Lost Temple OB         | 00:24:54 | lv1  |            | 26 May 2003 | BW 1.10 |         |
| matas vs PuSan 6.rep        | DDanS[gm] (T)          | 273   | PuSanS[G] (P)         | 251   | Lost Temple               | 00:27:20 | lv1  |            | 02 Jun 2003 | BW 1.10 |         |

圖 11 透過 BWchart 來管理遊戲記錄檔(<http://www.bwchart.com>)。



圖 12 透過 BWchart 檢視單場比賽的玩家雙方手速變化。  
縱軸表示手速，橫軸表示時間，上下分屬於兩位不同的玩家。



圖 13 透過 BWchart 檢視單場比賽的玩家資源變化。

縱軸表示累積資源量，橫軸表示時間，上下是兩位不同玩家，各有兩條曲線表示的是遊戲中的兩種資源。



圖 14 透過 BWchart 檢視單場比賽玩家的建造與生產。

橫軸表示時間，每條線的頂端有名稱表示生產、建立、研發的項目名稱。

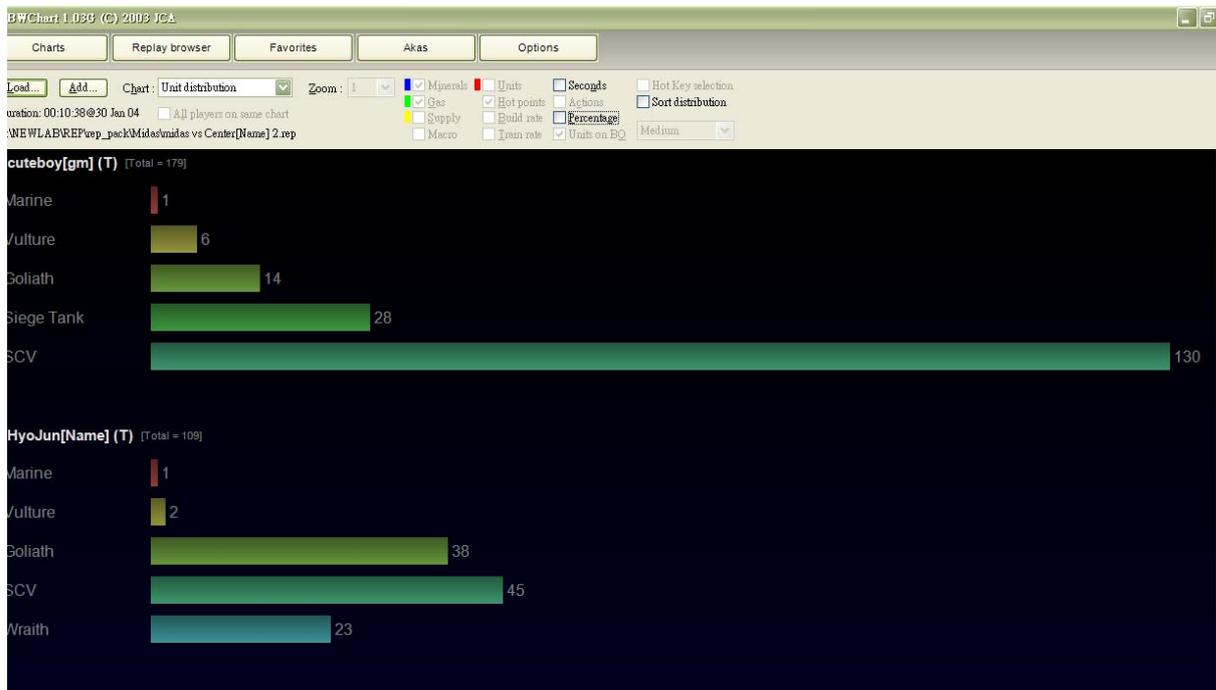


圖 15 透過 BWchart 檢視單場比賽玩家的配兵比例。

上方玩家的配置主要以工兵(SCV)為主的資源戰，下方玩家則是搶先發展空軍(Wraith)打的是科技戰。

這些工具也提供將二進位的編碼轉換成文字的功能，讓有興趣的玩家進一步進行研究。(圖 16)是利用一套叫做 Lmr(LordMartin Replay Browser)的免費軟體轉換出的文字檔，該檔案由標頭資訊以及玩家所下的遊戲指令構成，標頭資訊包含紀錄檔的日期、所使用的地圖、參與遊戲的帳號、使用的種族等資訊。在標頭資訊之後的是一長串的遊戲指令，每行指令都包含著執行該指令的玩家、執行該指令的時間點、以及跟該指令相關的參數，指令會依照時間的順序來排列。這些指令都是遊戲中定義好的，玩家透過鍵盤滑鼠進行遊戲的同時，會有介面負責將玩家的行為轉換成遊戲中合法的指令，為了節省電腦資源，一些對遊戲環境不會產生影響的動作並不會記錄下來，例如玩家切換視野的動作。

遊戲記錄檔是一個非常方便的資料訓練來源，因為它容易大量取得，記錄檔的品質通常也都不錯，玩家樂於將自己的得意之作放在網路上給大家分享，一些比賽的內容也透過知名的公開網站大量散佈。對我們來說，較大的限制來至於遊戲本身並未開放原始碼，也就是說我們可以得知玩家在遊戲中採取了哪些動作，但是我們卻無法得知採取這些動作以後實際的後果—「我們不知道兩軍交戰後各自損傷了多少單位、哪些建築被拆毀」。

```
<NORMAL_ACTIONS>
[E:\NEWLAB\REP\replays.net.orig\replays.net\200510\051001\Fly[jOin]_KT.MGW)LastMaN.rep]
_1,0
_074
_KT.MGW)LastMaN
_PGT - Nostalgia [02]
_2005-04-29 06:08
_SaDoL[S.G],0,T,1,Human,2,200
_KT.MGW)LastMaN,1,T,1,Human,2,233
```

標頭資訊

```
15,SaDoL[S.G],0,Train,31,7,SCV
15,KT.MGW)LastMaN,1,Train,31,7,SCV
45,SaDoL[S.G],0,Train,31,7,SCV
415,SaDoL[S.G],0,Train,31,7,SCV
430,KT.MGW)LastMaN,1,Train,31,7,SCV
650,KT.MGW)LastMaN,1,Train,31,7,SCV
655,SaDoL[S.G],0,Train,31,7,SCV
815,KT.MGW)LastMaN,1,Train,31,7,SCV
865,SaDoL[S.G],0,Train,31,7,SCV
1145,KT.MGW)LastMaN,1,Train,31,7,SCV
1250,SaDoL[S.G],0,Build,12,1966189,Bui
```

玩家指令

圖 16 解碼以後的記錄檔內容。

標頭資訊記錄比賽中的玩家、地圖、日期等資訊。

玩家指令則是由遊戲時間、玩家帳號、下達指令組成並以逗點區隔。

## 3.3 實驗模型與架構

### 3.3.1 玩家動作模型

即時戰略的即時性不僅替遊戲吸引到喜歡快節奏的玩家，也讓玩家開始思考操作速度的可能性，操作的速度和策略觀念被一些玩家比喻為軟體和硬體的關係，他們認為光有策略頭腦是不夠的，玩家必須具備有短時間執行多個操作的能力，唯有軟硬體互相搭配，才能提升自己的實力。手速的概念就在這種狀況下被帶入即時戰略的世界裡，手速的單位為 APM(actions per minute，每分鐘執行的指令數)，普通的熟練玩家手速可以到達一百多(每秒約三個指令)，職業的玩家中則有高達四百多的玩家(每秒 7 個指令)。

自從手速的概念被帶入到即時戰略遊戲以後，許多玩家開始注重個人的鍵盤滑鼠操作速度，為了追求更快、更有效率的操作速度，玩家需要學會快速鍵的使用。遊戲中設置的快速鍵分為兩類，一類是遊戲中設定好的動作鍵，玩家可以透過動作鍵下達攻擊

或是建造的指令，另外一類則是由玩家自行設置的選取鍵，在星海爭霸(starcraft)這款遊戲中遊戲允許玩家將部隊編制到數字鍵 0-9 底下，玩家可以透過數字鍵快速的選取編制過的單位，之後我們所稱的快速鍵(又稱熱鍵)都是指選取鍵。快速鍵的使用方式會因為玩家個人手掌大小、使用種族、使用戰術、鍵盤偏好、遊戲階段而有所不同，我們將會透過這樣的方式來辨認玩家。

### 3.3.1.1 玩家的按鍵使用習慣

在即時戰略的遊戲中，玩家必須先圈選想要控制的單位才能夠下指令。圈選的方式分為滑鼠圈選與熱鍵圈選，所謂的熱鍵圈選是玩家可以透過鍵盤上的按鍵對已經編制過的單位作圈選的動作。熱鍵的概念相當於戰場上的傳令兵，玩家在遊戲中扮演的是一位將軍的角色，如果使用的是滑鼠的圈選，表示將軍現場親自下令指揮部隊，但是很多時候玩家的視野並不在部隊的身上，尋找部隊的動作實在是太花時間了，透過熱鍵玩家可以快速地對部隊下達命令，或者將視野快速移到以部隊為中心，這就如同將軍透過電話向傳令兵下達指令或者詢問戰況一樣。

許多即時戰略遊戲都提供快速鍵的功能，方便玩家快速的切換在不同的單位之間，而我們選定星海爭霸(starcraft) 就提供了數字鍵 0-9 給玩家自行編輯想要快速圈選的單位，編輯的方式是先在遊戲中用滑鼠圈選該單位然後按下 ctrl + 數字鍵，之後如果要選取該單位就只要按一下該數字鍵就可以達到，如果按了兩下數字鍵還可以將視野快速移到以該部隊為中心。

玩家可以在任何一個時刻更換數字鍵的編制單位，編制的動作跟快速鍵選取的動作都會被記錄檔記錄下來。快速鍵的使用與玩家的鍵盤使用習慣有關係，隨著時間的推移，不同快速鍵的使用頻率會有所變化。有些教學網站更提供指法教學，將特定的戰術或是單位的操作與快速鍵的編制、使用聯繫在一起，企圖讓玩家可以增加手速及效率。我們將會篩選遊戲的前五分鐘，記錄十個快速鍵分別的累積使用率，這段時間處於遊戲的初期，快速鍵的使用比較不會因為採用的戰術、單位不同而出現太多變化，而且大部分的比赛也都超過五分鐘。

### 3.3.1.2 累積曲線與比較演算法

為了讓沒有使用快速鍵的玩家也能夠彼此區別，我們將滑鼠的圈選動作也列入觀察的對象，加上原本的十組熱鍵(數字鍵 0-9)總共可以得到 11 條曲線，採取累積曲線方便我們掌握該快速鍵的使用率隨著時間的移動產生什麼樣子的變化。我們每隔三秒觀察一次累積值，所以在遊戲的前五分鐘可以得到  $5 \times 60 / 3 = 100$  個點，利用這 100 個點我們

計算兩條熱鍵的使用累積曲線  $Curve_i$  以及  $Curve'_i$  之間的歐幾里得距離  $Curve\_Dist$ ，其中  $Curve_{i,j}$  是熱鍵  $i$  第  $3*j$  秒觀察到的使用次數累積值。

$$Curve\_Dist(Curve_i, Curve'_i) = \sqrt{\sum_{j=1}^{100} (Curve_{i,j} - Curve'_{i,j})^2}$$

以這樣的概念出發我們透過 11 條曲線的歐幾里得距離來算出兩位玩家 Ply1、Ply2 的指法距離(Finger\_Dist)，距離越短則表示他們擁有的快速鍵使用習慣(指法)越類似，我們期望同一位玩家的比賽會是擁有最小距離的一群，因為玩家間可能有一樣的編隊方式，但是熱鍵的使用頻率、隨時間的變化卻不容易一致。

$$Finger\_Dist(Ply1, Ply2) = \sqrt{\sum_{i=1}^{11} \sum_{j=1}^{100} (Ply1.Curve_{i,j} - Ply2.Curve'_{i,j})^2}$$

### 3.3.1.3 搜尋演算法

我們的搜尋過程採用兩個階段的搜尋，先以玩家的帳號作搜尋，挑出可以確定的該位玩家帳號，再利用那些資料作為基礎，蒐尋該位玩家散佈在不同帳號下的資料，詳細的實驗步驟如下：

- (1) 搜尋可能的玩家帳號，並挑選確定的幾組帳號，取出那些帳號的比賽。
- (2) 計算比賽間的指法間離，利用最小距離合找出最具代表性的一筆記錄檔。
- (4) 算出代表性資料與其他已知比賽的平均值與標準差。
- (5) 以  $Threshold = 平均值 + 標準差$ ，回到記錄檔資料庫(DB)進行搜尋的動作，所有指法距離小於  $Threshold$  的記錄檔都會被視為該為玩家(Target)的資料。

$$Searched(Target, DB) = \{i \mid Finger\_Dist(i, Target) \leq Threshold, \forall i \in DB\}$$

## 3.3.2 玩家策略模型

### 3.3.2.1 策略的定義與抽象化

即時戰略遊戲的進行由「收集資源」、「建造建築」、「升級科技」、「生產單位」、以及「控制單位」所組成，其中又以建造建築為指標，舉凡收集資源、升級科技、生產單位都仰賴不同種類的建築來完成。我們將建築類型的集合設定為我們的策略狀態，不同種類的建築集合表示著可以做的事情不同，而同種類的建築在數量上的變化代表著生產力，數量越多表示你可以同時進行的相同工作越多，但是可以做的事情基本上是一樣

的。可以想像的到，一場比賽的建築總類集合會不斷的擴大，不同的擴大的方式便是玩家所採用的不同策略，集合之間的差異度越大，表示所使用的策略越不相同。策略所形成的圖形會是樹狀的，從根節點一路往下的路徑就是一連串的決策，每個節點就是一個狀態，每場比賽會在地圖上貢獻一筆路徑，重複走過的路徑我們以權重的方式來表示。

### 3.3.2.2 盤面的定義

策略的預測成功率仰賴兩個因子，第一個是在過去的比賽中玩家是否有下過同樣的策略，第二個是我們對盤面的估計與玩家內心所想的有多接近。第一個因子我們只能透過增加訓練的資料來提升，而第二個因子則與我們的觀察值與設計的演算法有相關。為了讓這個模型可以方面的移植到其他的即時戰略遊戲上，我們應用案例推論中盤面描述的概念，並作適當的修改，讓它更一般化、更適合用在記錄檔的資料上。在原本的做法中，研究者提出了八項觀察值分別是一「雙方殺敵數差」、「雙方摧毀建築物差」、「敵方建造建築數」、「敵方生產戰鬥部隊數」、「敵方生產工兵數」、「我方目前建築數」、「我方目前戰鬥部隊數」、「我方目前工兵數」。該篇論文針對這八項觀察值系統取得八個數字，計算他們的歐基里德距離，取其距離最近的幾個，再從中取得效能最好的一組策略來增加系統的強度。該論文是一個在 TIELT 平台上所進行的研究，實驗中可以取得遊戲世界完整的資訊，只有得到模擬器我們才有可能透過記錄檔得到一樣多的資訊。在記錄檔的限制下關於殺敵數、摧毀建築物數、以及目前各項數值我們無法得到確切的數字，但是我們可以估計出大致的建造、生產數量。此外，在即時戰略遊戲當中事實上有種族的差異性在的，不同的種族對戰不宜用同樣的標準去測定雙方的戰鬥部隊數目。一種可行的方式是我們將部隊轉換成遊戲中生產該部隊所需要的資源量，直接來比較雙方的資源，不過這又會牽扯到遊戲設定的平衡性問題。所以我們採用了一個比較單純的方式，我們將敵我的每一種單位、科技、建築都列為一種觀察值，並利用評級的機制來平衡三方面的權重。

### 3.3.2.3 盤面的距離計算

在我的盤面描述中列了六大觀察值，顯示了我方與敵方的經濟狀態，分別是「我方單位生產數」、「我方科技研發數」、「我方建築建造數」、「敵方單位生產數」、「敵方科技研發數」、以及「敵方建築建造數」。每大觀察值又可細分成遊戲中對應的單位群、科技群、以及建築群如下式所示。

$$\begin{aligned}
Units &= [U_1=工兵數, U_2=步兵數, U_3=坦克數, \dots, U_n=飛機數] \\
Techs &= [T_1=攻擊力等級, T_2=防禦力等級, T_3=射程, \dots, T_m=跑速] \\
Builds &= [B_1=主堡數, B_2=兵營數, B_3=科技中心數, \dots, B_k=飛機場數]
\end{aligned}$$

我們針對兩個盤面描述中的「我方單位生產數」，取歐基里德距離後可以得到一個距離值(Distance)，其中 n 表示我方種族的單位共有多少種類，見下式。

$$Distance(Units, Units') = \sqrt{\sum_{i=1}^n (Units_i - Units'_i)^2}$$

針對剩下的五大觀察值我們也用同樣的方式取得距離，我們將得到六個數字，這六個數字代表著兩場比賽中敵我雙方形勢差距有多大，如果六個數字都很小表示兩場比賽中的這兩個時間點有很類似的狀況出現。由於單位、科技、建築在數量上的分佈有差異，一場遊戲下來玩家動輒生產上百位步兵，但是單一種科技往往只有不到三級的研發上限，而我們希望能夠平衡單位、科技、建築這三者的權重，所以我們簡單地採用一個「排名次」的方式來標準化三者的距離。我們依據目前盤面(Now)的觀察值，計算資料庫(DB)中所有盤面(Past)的觀察值距離(Distance)並作排序的動作，距離越短取得的名次數字越小。

將距離(Distance)換算成名次(Rank)以後我們得到一個標準化的數字，讓我們能夠平均六個觀察值的權重。我們將六個觀察值換算後的名次加起來，定義為過去的盤面(Past)與現在的盤面(Now)在過去的盤面資料庫 DB 下的名次距離值(Rank\_Distance)，距離越短則盤面越相似，我們用 Rank\_Distance 取代原本的 Distance 來計算下一節中的效能。

$$Rank\_Distance(Past, Now, DB) = \sum_{i=1}^6 Rank(Past_i, Now_i, DB)$$

### 3.3.2.4 效能

我們之所以要計算效能，是為了替過去的經驗給定一個評分，如果玩家在過去的經驗中因為採用此種戰術而很快地獲得勝利，那麼該玩家在之後的同樣狀況下就更有可能採取同樣的戰術，反之，玩家可能會因為過去的失敗經驗，而考慮採取其他種作法。我們計算效能的公式很簡單，一場勝利的比賽中所有的策略都有一分的進帳，而一場輸掉的比賽則都是負一分，一場平手或是不知勝負的比賽則是零分(我們判定勝負的方式是看完家離線的指令，但是並非每場比賽紀錄檔都會有這樣子的指令，因此是有可能無法得知勝負的)。一場過去比賽(Past<sub>i</sub>)中如果曾經出現跟目前(Now)一樣決策狀態的話就擁有投票權，投票的分數(Score)是過去的效能(Performance)除以過去跟目前盤面之間的

距離(Rank\_Distance)。系統最後會算出目前可執行決策(Strategy)各自的總分並採納最高分的決策。

$$Performance(Strategy, Game) = \begin{cases} Win : 1 \\ Unknown \& Draw : 0 \\ Lose : -1 \end{cases}$$

$$Score(Strategy) = \sum Performance(Strategy, Past_i) / Rank\_Distance(Past_i, Now, DB)$$

### 3.3.2.5 策略的預測

預測實驗步驟如下：

- (1) 根據指法距離篩選出玩家的記錄檔，並根據遊戲日期排序。
- (2) 從訓練資料中挑出最早以前的一場記錄檔並將他從訓練資料中移除(我們相信玩家也會在自己的遊戲過程中學會哪些策略比較好用，因此我們用比較早的記錄檔作學習並預測玩家後來的決策會比較合理)。
- (3) 尋找記錄檔中的事件點(新總類建築物被建造的時間點)，取出盤面狀況、玩家決策、以及決策效能
- (4) 將盤面狀況拿來詢問決策系統，並比較玩家決策與系統決策
- (5) 將預測的結果記錄下來。
- (6) 預測過的事件加入到資料庫中，以供之後的查詢。
- (7) 如果還有事件尚未處理則跳至(3)，否則跳至(2)。

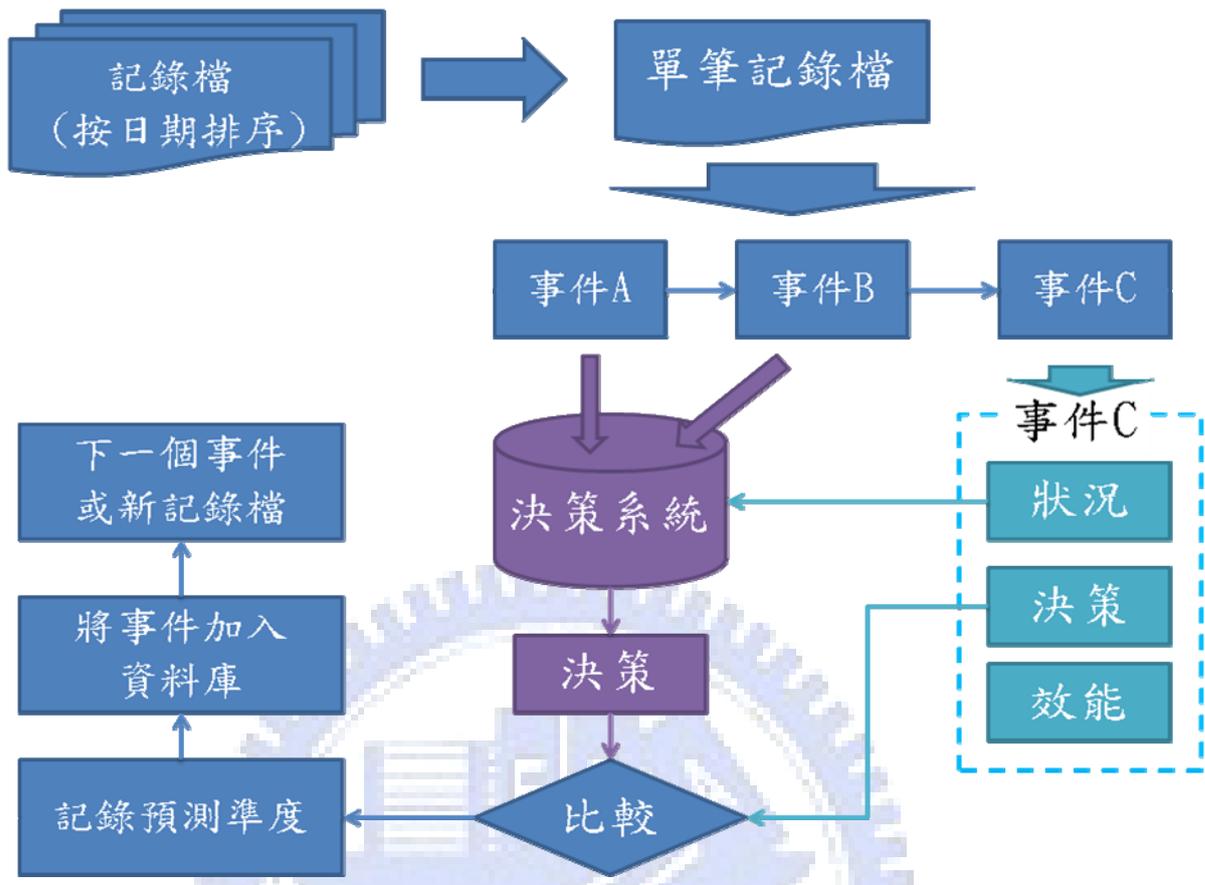


圖 17 策略的預測流程表

## 四、實驗與結果

### 4.1 玩家辨識實驗

玩家辨識的實驗比較難得知效果，因為同一位玩家可能玩上好幾個不同的帳號，不同的玩家也有可能互相交換帳號使用。在此我們採用的做法是從玩家整理好的幾位職業選手資料下手，比對搜尋到的資料量相對於該玩家比賽總量的比例。在門檻設定為平均值加上一倍標準差的距離下大約可以找出 30%-40% 該玩家的紀錄，由於我們將搜尋目標放在該玩家的主要種族上，所以非主要種族的資料會被忽略，這表示我們的搜尋比例應該還要再多一些。在實驗中最讓人驚訝的是，被搜尋出來的資料有些橫跨了三年的時間以上，表示這樣的習慣在玩家的身上多年來一直沒有太大的變動。由於我們最終的目的是建立特定對象的建築策略模型，玩家辨識的目的只是方便我們將目標資料鎖定，所以我們會在下一個章節針對系統搜尋以及玩家社群手動分類的資料做比較，來驗證我們玩家辨識究竟有沒有辦法達到所想要的目的。

### 4.2 玩家策略的預測

#### 4.2.1 建造策略的母空間

在開始策略預測的實驗之前，我們先計算決策空間的特性，方便我們瞭解決策的複雜度與可能的預測效果。在星海爭霸(starcraft)這款遊戲中，三個種族擁有的建築特性截然不同，我們將利用遊戲中的建築科技樹，計算出所有可能的建築狀態、決策數、以及策略平均分支度(表 2)。一個建築狀態是玩家目前擁有的建築種類集合，代表著玩家策略的布局狀況，而相差一種建築的兩個建築狀態間可存在一個決策(建造該建築)，如果把建築狀態視為圖上的點，決策就可以視為兩點之間的邊，策略的分支度則是表示在目前的建築狀態下有幾種決策可以挑選。

表 2 三個種族的策略母空間

| 100場訓練以後 | 人類                | 蟲族              | 神族              |
|----------|-------------------|-----------------|-----------------|
| 狀態數(點)   | 2234              | 901             | 388             |
| 決策數(邊)   | 10887             | 3828            | 1354            |
| 策略平均分支度  | $10887/2234=4.87$ | $3828/901=4.25$ | $1354/388=3.49$ |

從上面的表可以得知，三個種族的可能決策數量上有非常大的差異，人類的策略空間明顯的比蟲族跟神族要來的大。我們由平均分支度大概可以得知，一場比賽平均而言有 3~5 種策略可以選擇，也就是說如果我們任意選擇一種策略，將會出現 20%~30% 的預測準度。為了更了解決策的分布，我們將建築狀態與分支度做個統計(圖 18)，發現決策分支度呈常態分布，大部分的決策都集中在分支度四到五附近，至於實際玩家的策略選擇集中度有多高，我們將由後面的實驗來做說明。

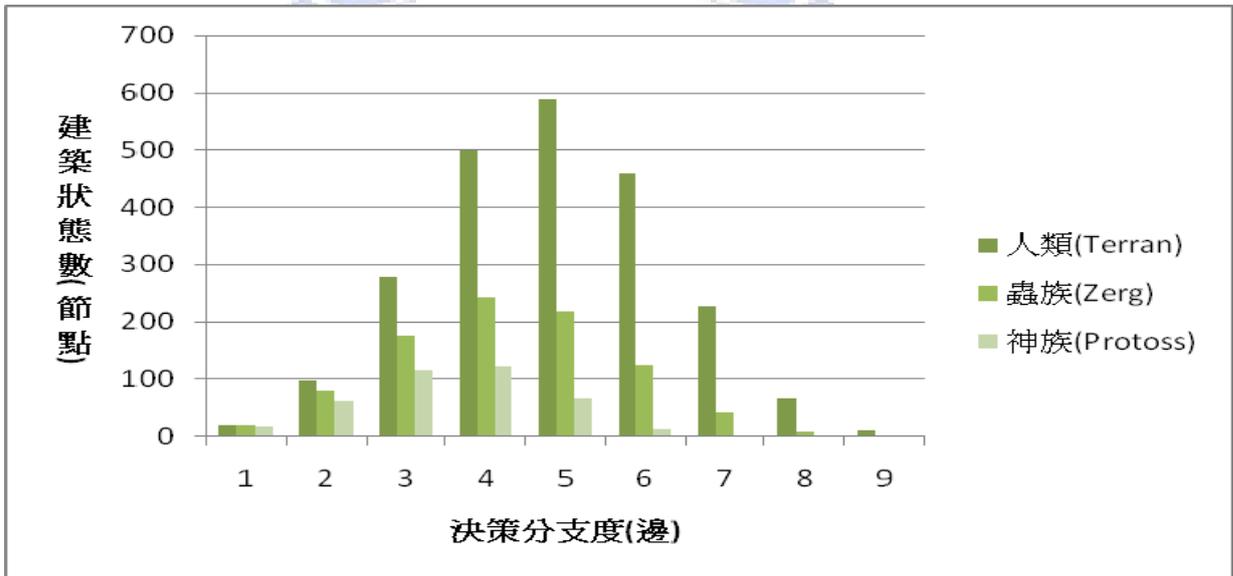


圖 18 「星海爭霸」遊戲設定下的所有可能的策略分支度分佈圖

## 4.2.2 建造策略的觀察值

在實驗中我們觀察以下變數—「玩家本場次使用的策略數」、「本場次策略曾經在過去使用的數目(出現在資料庫中)」、「本場次策略被系統猜中的數目」，並畫出三條曲線分別代表著預測準度上限、目前系統預測準度與上限相較的比例、與系統預測準度。一場比賽下來玩家使用的策略有可能是先前所沒有用過的，這樣子的策略我們的系統並沒有機會預測中，所以該策略是否在資料庫中的比例是我們系統的預測準度上限：

**(1) 該策略在資料庫中存在，表示系統有機會預測中，如(圖 19)的 C/B:**

在訓練的過程當中資料庫中的策略會漸漸囊括玩家使用的所有策略，越到訓練的後期玩家越難出現資料庫所沒有的建築策略，而這條曲線反應的是玩家對於策略的變化度以及廣度。曲線的變化只跟我們選擇的訓練資料有關係，也就是我們可以透過篩選出來的資料看到該(類)玩家的策略究竟是一成不變還是變化多端，但是也有可能因為玩家辨識模型的效果不好，囊括太多玩家的記錄檔導致不同人的策略風格混在一起，所以我們需要透過手動分類的資料來驗證玩家的策略變化風格。

**(2) 該策略在資料庫中存在，而系統也準確的猜中，如(圖 19)的 D/C:**

當策略存在於資料庫中出現，表示我們有機會從過去的經驗得到正確的預測結果，如果我們的策略模型非常成功，那就應該是曲線呈現成長的狀態。但是，在資料庫成長的過程中，策略的數目會增加，很有可能為同一個建築狀態帶來更多種策略的選擇，如果策略模型的準確率成長跟不上玩家策略分支度的成長，曲線的確是有可能下降的。

**(3) 我們最終想得到的猜中率，如(圖 19)的 D/B:**

我們最終想得到的命中率，是玩家一場比賽下來有多少策略被我們猜中，這受到前面兩個因素的影響，一個過去沒使用過的策略是不會被猜中的，必須要先在過去使用過，然後經過精準的預測方式，才有機會被猜中。我們希望這個曲線會呈現穩定成長的狀況。

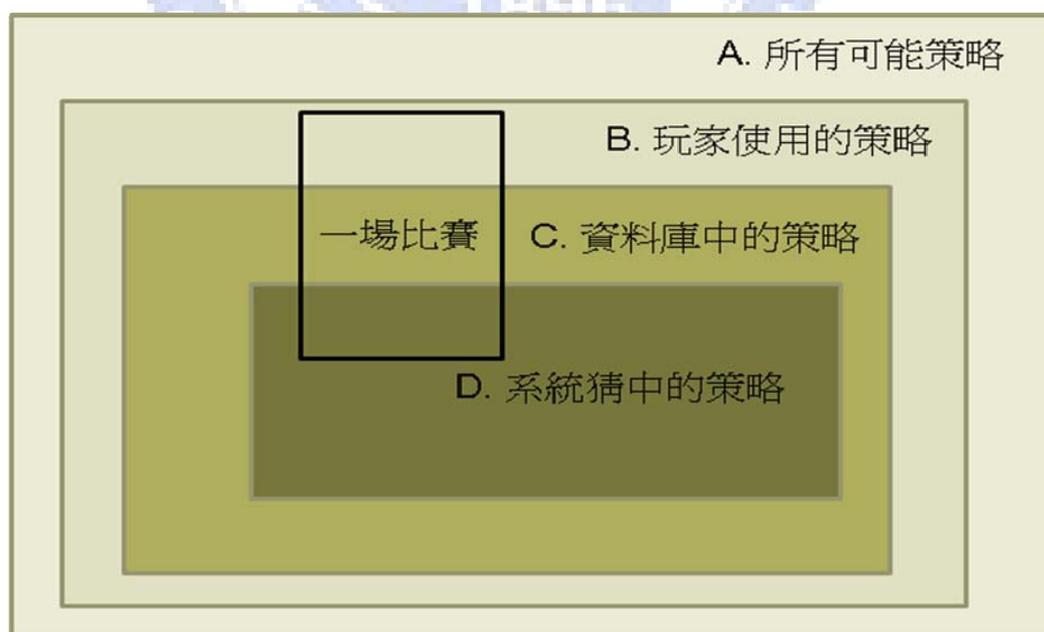


圖 19 檢驗的變數圖示

### 4.2.3 玩家辨識模型 VS.人工整理資料

第一個實驗我們要回應前面所提到的玩家辨識模型的效果，在這裡我們依然採用記錄檔驗證的方式。我們從人工整理的記錄檔中抽出五分之一作為驗證用資料，這將保證我們驗證的資料的確來自於該玩家，而這些資料將會由原本的兩種模型訓練過程中除去，保證不會被拿來訓練。上述的過程將會被重複五次取平均值，而每次我們抽出來的驗證用資料都會不同，以增加實驗的可信度。

當訓練以及驗證的資料選定以後，我們將兩份訓練資料分別拿來訓練各自的決策系統，為了公平性，我們會以相同數量的比賽進行訓練，訓練好的資料庫同時對我們要驗證的記錄檔做預測，(圖 20)是我們的實驗流程，(表 3)是我們的實驗設定與結果。我們從遊戲中的三個種族中各挑出一位具有代表性的玩家，比較我們透過玩家熱鍵搜尋出來的記錄檔以及玩家社群以人工方式收集來的該位玩家記錄檔在訓練決策系統上的效果，透過表中的比較我們可以發現，兩個決策系統有大約七成的策略預測結果是相同的，而透過人工方式整理出來的資料針對玩家實際決策確實有比較高的預測準確度，這是因為人工整理出來的資料是 100%屬於該位玩家的，但是我們系統自動化搜尋的效果也不差，兩者的差異低於百分之四，這表示我們的系統找出來的資料的確有機會夾雜到其他玩家，但影響的比例不大。

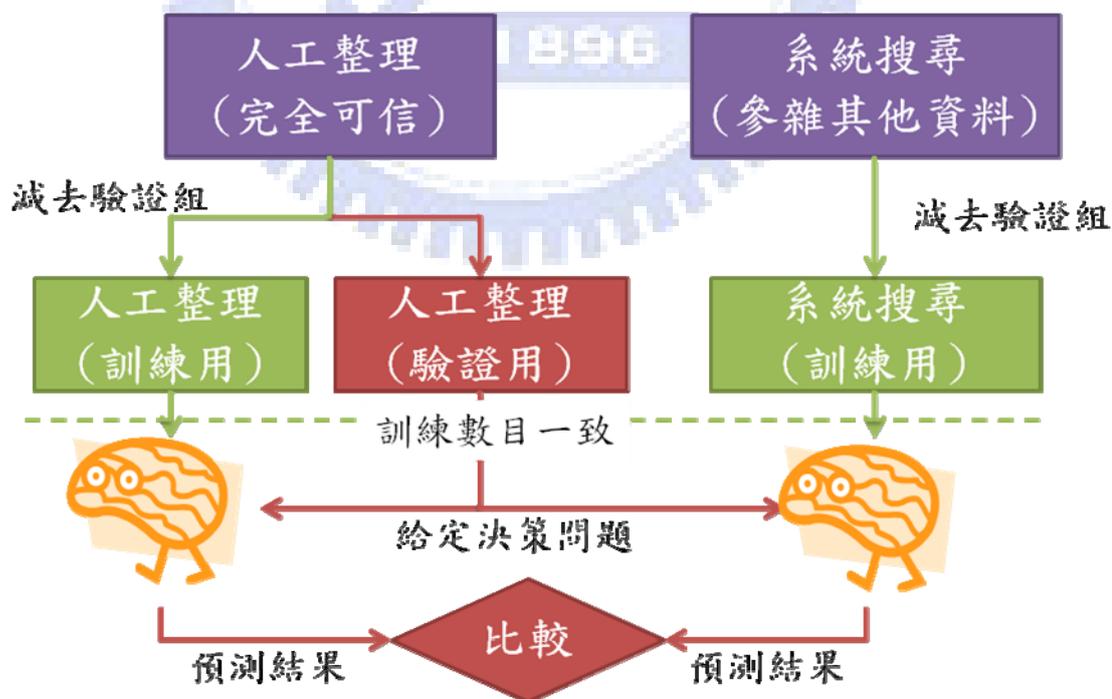


圖 20 玩家辨識實驗流程。

在這圖中我們比較兩者的預測結果有多高的相同度，此外我們也比較兩者預測玩家實際決策的能力。

表 3 玩家辨識實驗結果:

遊戲中每個種族我們各挑選一位玩家做實驗，每次取不同的 1/5 份人工整理資料作驗證組，共進行五次實驗取平均值。我們比較兩份資料訓練出來的決策系統在進行同一份驗證資料的預測時有多少比例是相同的預測結果，並分別與實際的玩家決策做比較，觀察何者在預測玩家決策上比較準確。

| 玩家           | 訓練用記錄檔數 | 驗證用記錄檔數 | 兩種分類預測相同 | 人工分類預測正確 | 系統分類預測正確 |
|--------------|---------|---------|----------|----------|----------|
| 人類玩家(Boxer)  | 227*5次  | 64*5次   | 67%      | 65%      | 61%      |
| 神族玩家(Nal_rA) | 109*5次  | 30*5次   | 72%      | 71%      | 68%      |
| 蟲族玩家(Yellow) | 194*5次  | 55*5次   | 71%      | 68%      | 66%      |

#### 4.2.4 盤面判定法的效果檢測

透過玩家熱鍵的辨識模型我們可以找到該位玩家的可能記錄檔，假設我們的系統在當下完全掌握對手的資訊，那下列三張圖就是決策系統針對三個種族各一位玩家的預測結果。透過趨勢線我們可以看到三位玩家成長的方式很有差異性，蟲族玩家打法呈現兩極化，我們從(圖 22)可以看得出來曲線上下震盪的很厲害，靠近上方是很容易完全預測中的打法，靠近下方則是完全預測不中的打法，為了更了解問題的原因，我們調度另外一個觀察曲線，看看是不是我們系統的關係導致預測能力有差異，或者是該玩家打法本身就是這樣。

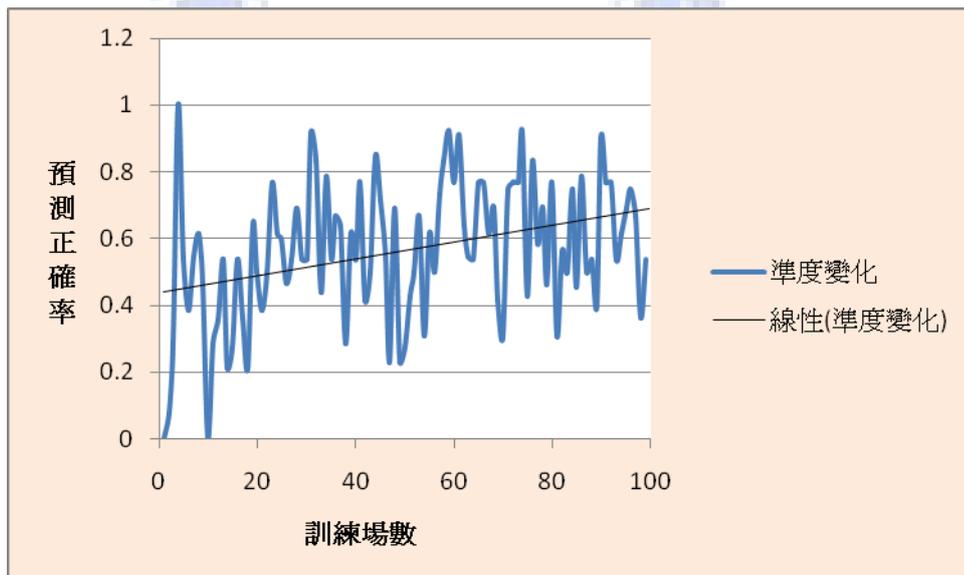


圖 21 系統成長圖-人類玩家

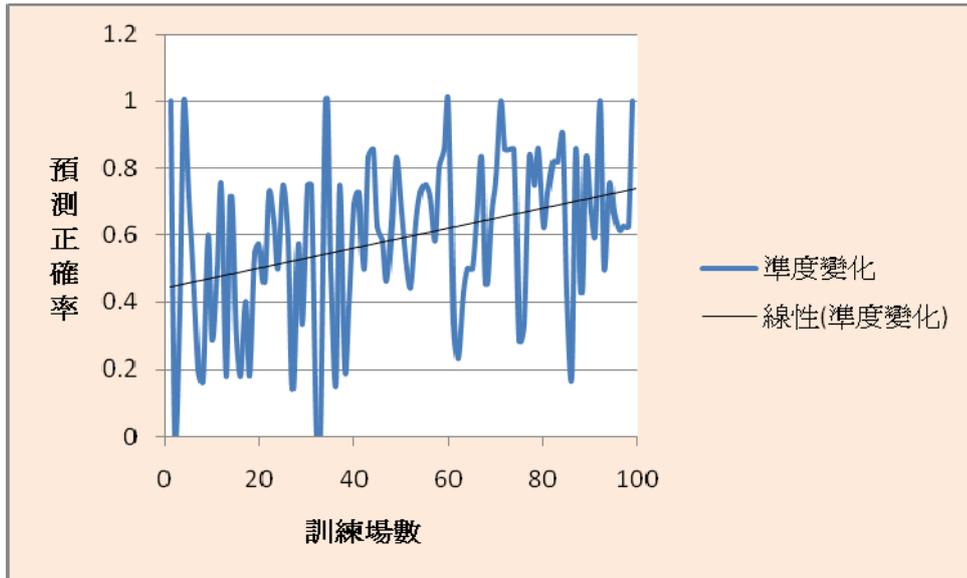


圖 22 系統成長圖-蟲族玩家

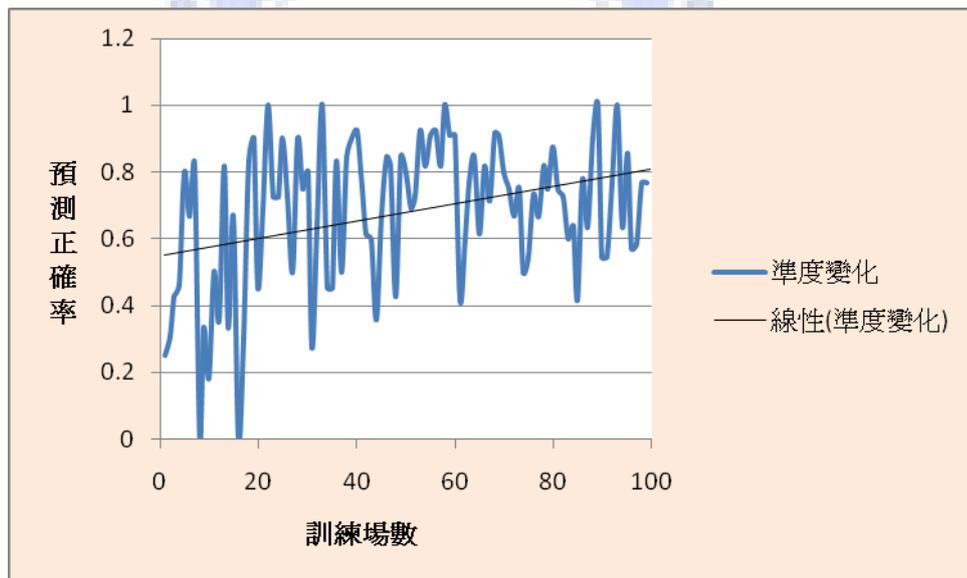


圖 23 系統成長圖-神族玩家

從(圖 24)我們可以看得出來，預測成功率的上限(系統只能夠預測出玩家過去曾經使用過的策略，玩家本次如果用了很多全新策略，那成功率上限就會很低)，在訓練的前中段與最後幾場也都呈現很低迷的狀態，這正好和上面第二張圖的情形不謀而合，我們可以確定那幾場的打法很別於該玩家常用的打法，於是我們猜測會不會是因為那幾筆資料根本就不是該玩家所打的呢？為了澄清這個疑慮，我們又將人工整理的資料也調出來查閱。

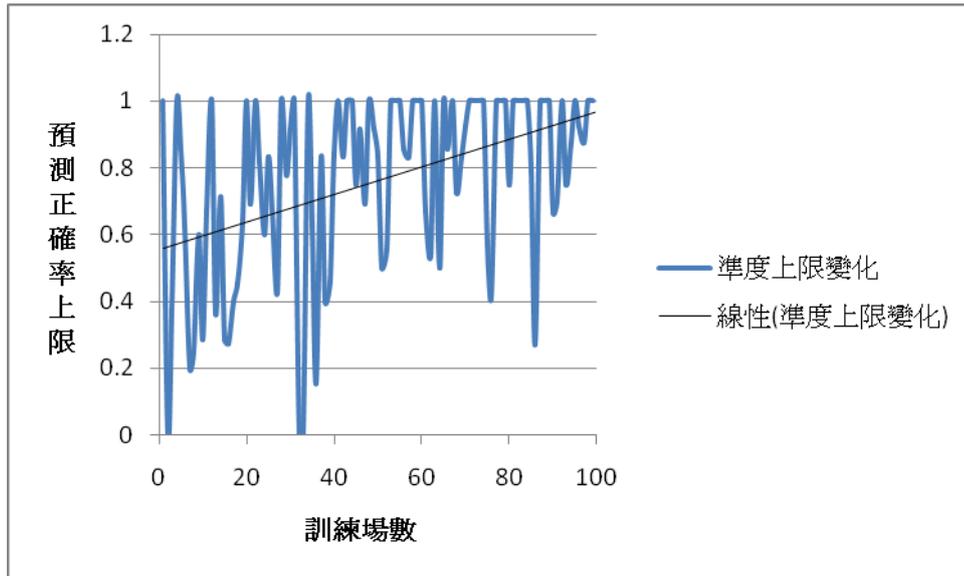


圖 24 蟲族玩家的策略成長圖 (系統搜尋資料)

(圖 25)是我們從人工整理的資料調閱出來的結果，雖然沒有上圖那麼多接近於零地成功率上限，但是前中段的幾場，還是出現有相當低的成功率上限，至此我們可以得到一個結論。該玩家平時大量的類似打法，似乎有為幾場特別的打法作鋪路的作用，一個想在關鍵時刻用剪刀戰勝對手的人，必須佈局讓人家相信他一定會出石頭。

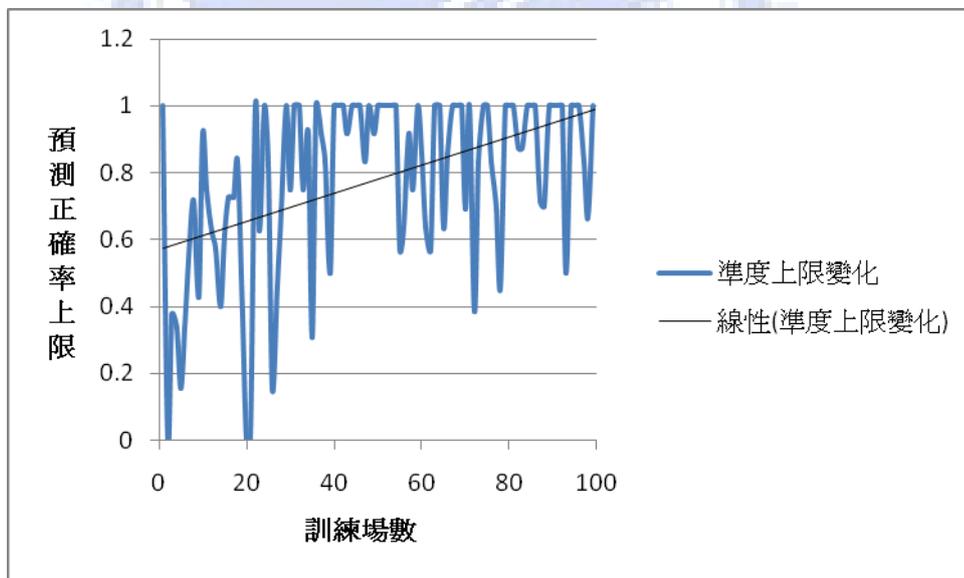


圖 25 蟲族玩家的策略成長圖 (人工整理資料)

#### 4.2.5 玩家的策略集中程度

從(表 4)中我們可以看出玩家地策略集中程度相當的高，尤其是策略母空間最大的種族「人類」，玩家 Slayers\_Boxer 事實上已經是在職業界打滾五六年以上的元老級職業

選手了，不斷創新打法也是該位玩家在大家心目中的印象，可是他在所有可能的策略狀態中只佔了 7% 而在所有可能決策中也只佔了 2%，這是非常驚人的集中。

此外，有趣的是蟲族在遊戲中本來的定義下，擁有的狀態點與決策數都比神族高了快三倍之多，可是在實際打法下，該神族玩家卻比起蟲族玩家有更多的狀態數與決策數。這代表了遊戲策略的發展雖然受到遊戲設計的影響，導致人類玩家比其他兩類玩家的策略更複雜，卻也可能基於玩家本身對打法的理解與變化，導致神族玩家比起蟲族玩具有更複雜的決策空間的結果。

(表 4)的倒數第二列指出了我們系統預測準度期望值應該要有的準度，這是透過玩家策略空間的平均策略分支度算出來的，在人類玩家的策略空間中共出現了 148 個不同的建築狀態(遊戲設定下所有可能狀態點有 2234 個)以及 277 個不同的決策(遊戲設定下所有可能決策有 10887 種)，計算下來平均的策略分支度是  $278 / 148 = 1.87$ ，這表示我們如果透過以往的經驗要來猜測玩家下一個策略的時候，平均而言有 1.87 種選擇，也就是我們不看目前的局勢光從過去的策略中隨意挑一個可用的策略也會有  $1 / 1.87 = 53.4\%$  的理論準確值，這也難怪我們在不了解盤面的狀況下依然可以得到一個不錯的預測值。而我們的系統預測準度又比期望值高，這是盤面判定法下所得到的一個進步。

從(表 5)以及(表 6)的數值變化中我們可以發現，玩家的決策圖在成長的過程中，策略的平均分支度也會稍微的上升，降低了預測準度估計值，然而我們的系統預測準度卻仍然上升，這表示我們的系統在經驗增加的過程當中具有良好的穩定性。

表 4 一百場訓練以後的玩家決策空間大小佔母空間比例表。

| 100 場訓練以後  | 人類玩家(Boxer)         | 蟲族玩家(Yellow)       | 神族玩家(Nal_rA)        |
|------------|---------------------|--------------------|---------------------|
| 狀態數(點)     | 148 ( 7% of 2234 )  | 97 ( 11% of 901 )  | 109 ( 28% of 388 )  |
| 決策數(邊)     | 277 ( 2% of 10887 ) | 169 ( 4% of 3828 ) | 177 ( 13% of 1354 ) |
| 決策平均分支度    | 1.87                | 1.74               | 1.62                |
| 預測準度(不看盤面) | 53.4%               | 57.4%              | 61.6%               |
| 系統預測準度     | 55.9%               | 57.2%              | 67.5%               |
| 系統準度上限     | 68.8%               | 74.4%              | 81.3%               |

表 5 兩百場訓練以後系統的預測準度變化

| 200 場訓練以後  | 人類玩家(Boxer) | 蟲族玩家(Yellow) | 神族玩家(Nal_rA) |
|------------|-------------|--------------|--------------|
| 狀態數(點)     | 177         | 132          | 135          |
| 決策數(邊)     | 355         | 252          | 235          |
| 決策平均分支度    | 2.01        | 1.91         | 1.74         |
| 預測準度(不看盤面) | 49.9%       | 52.4%        | 57.4%        |
| 系統預測準度     | 62.6%       | 58.7%        | 71.2%        |
| 系統準度上限     | 80.3%       | 81.8%        | 86.7%        |

表 6 三百場訓練以後系統的預測準度變化

| 300 場訓練以後  | 人類玩家(Boxer) | 蟲族玩家(Yellow) | 神族玩家(Nal_rA) |
|------------|-------------|--------------|--------------|
| 狀態數(點)     | 202         | 145          | 157          |
| 決策數(邊)     | 421         | 288          | 287          |
| 決策平均分支度    | 2.08        | 1.99         | 1.83         |
| 預測準度(不看盤面) | 48.0%       | 50.3%        | 54.7%        |
| 系統預測準度     | 65.0%       | 60.5%        | 72.0%        |
| 系統準度上限     | 84.9%       | 85.5%        | 88.1%        |

(圖 26)是三百場訓練以後玩家決策平均分支度的分布圖，與前面遊戲設定下的母空間分布比較起來有兩處不一樣，第一是玩家的策略分支度明顯往低的方向移動，這在我們的預料之中。但是策略分支度的分布圖長得並不像原本的常態分佈，反而比較像是無尺度分佈。

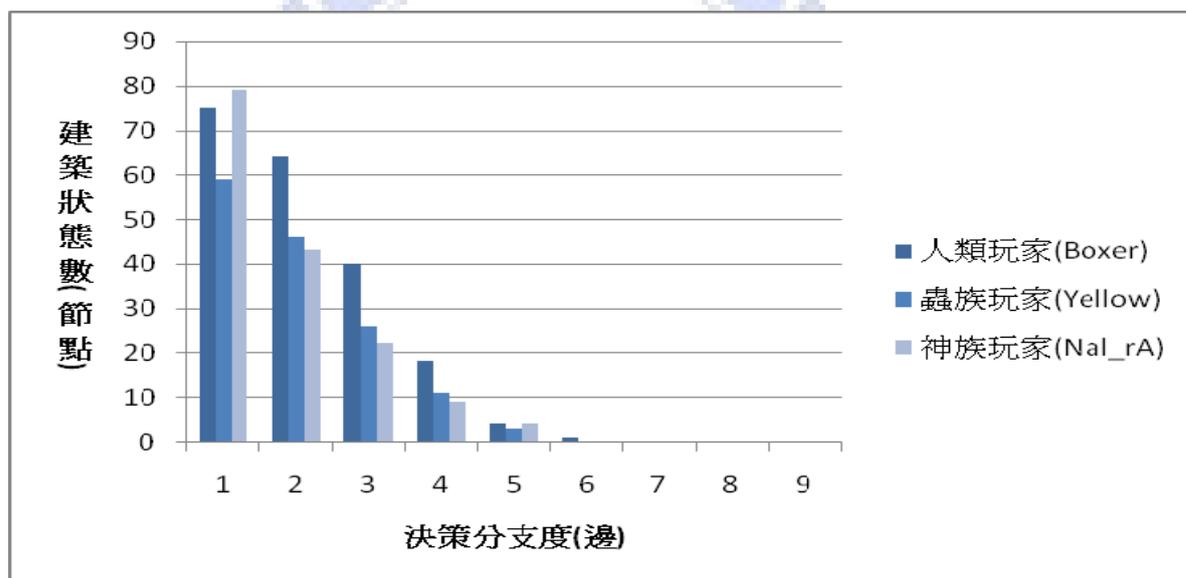


圖 26 玩家決策分支度分佈圖。該統計圖為三百場訓練後的結果

(圖 27)是我們系統自動產生的一張策略圖，玩家的每場比賽就是從圖的上方開始走起，每經過一個邊就會貢獻一點權重，線條的粗細表示採用過此種策略的次數，我們可以視為經驗的多寡。從圖中我們可以看得出來，大多數的策略經驗值並不多，但是少部分地策略卻經驗值異常的多，整體呈現的像是無尺度網路。圖的最上方，也就是遊戲開始的時候，策略選擇的複雜度通常不高，也非常集中在某幾條策略上，這就好像下棋的開局都有固定的幾種方式一樣。值得注意的是在中後段有線條反而變粗的狀況，我們可以視為重要的決策觀念，對玩家來說，有一些決策像是以重要觀念的方式存在，當走到某一個狀態點就觸發，而不太像是背公式從頭走下去。但整體而言玩家的決策看起來是有一定的主要路徑在走的，或許這是因為大家高度交流下存活下來的實戰打

法，但是這中間還是留有一定的自由度，自由度越高的部份，可能是還在探索出有沒有確實比較好的選擇，或者是其實這幾種選擇都差不多了，才沒有形成重要的觀念迫使玩家大量採用同一種決策。

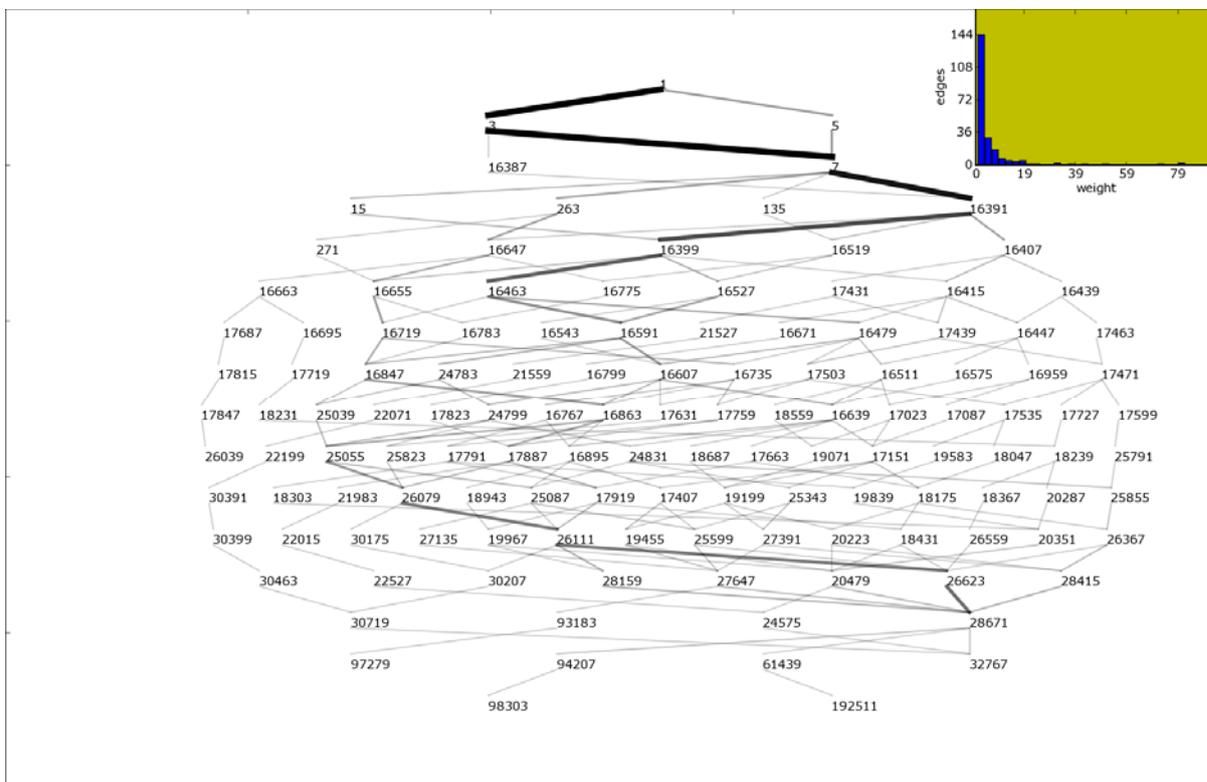


圖 27 人類玩家 Boxer 對上蟲族對手一百場之後的建築策略圖

## 五、結論

我們在第一章所提到的兩個目標「玩家辨識模型」、「玩家策略預測模型」基本上已經初步的完成了。在玩家辨識的部分，我們可以用已知的某場比賽或是已知的帳號下去搜尋出同一位玩家在不同帳號下的比賽，這樣的功能可以應用在目前許多公開網站的記錄檔搜尋上；傳統的記錄檔搜尋方式是透過 AKA (Also-Known-As) 的名單將玩家與帳號聯繫在一起，並透過玩家社群的朋友網路去整理、維護這樣的一個系統；這種方法仰賴長久、大量的人力去維護，而且有許多的記錄檔並沒有人認領，主要的原因在於玩家有時候會想要隱藏自己的真實身分去進行沒有壓力的比賽，而且通常只有一定名氣的玩家才會被列在名單上。透過我們自動化的搜尋功能，可以補足人力所難以達到的廣度，也可以做為輔助工具幫助社群尋建立 AKA 名單。除此之外，由於網路進行遊戲的風氣盛行，許多獎金規模比較小的比賽已經能夠讓世界各地的好手透過網路來進行，但是這種機制下引發出一個難以杜絕的問題——「代打」。登記比賽的玩家可能與實際比賽的玩家是不同一位，這對於另一方來說是一件很不公平的事情，可是規模小的比賽又難以花費金錢再去防治這樣子的行為，我們的玩家辨識模型是一個不錯的輔助工具，不僅可以查出這位玩家究竟是不是登記的那一位，也可以查詢出這位玩家究竟是誰；當然這樣子的機制並非像指紋那樣的具有說服力，因為不同一位玩家有可能擁有同樣的熱鍵習慣，而同一位玩家的熱鍵習慣有可能有輕微的變動，但是身為一個輔助工具可以進行快速的第一階段篩選，找出那些比較有問題的比賽再由人力的方式做更嚴格的比對，增加比賽的公平性，可以花費很少的人力資源就達到一定水平的審查機制。

在玩家策略預測模型的部分，在一百場的訓練下我們的系統已經能夠快速的預測出接近六七成的準度了（準度上限是七八成），這樣的效果其實已經算是不錯，可是要應用在真實的遊戲上的話還需要其他階層性設計的配合。首先要考慮到的就是策略執行的時間點，在我們的預測模型當中我們是已有當時玩家決策的時間點，我們只需要決策系統在當下給定一個好的策略，可是當我們應用在真實遊戲的時候，就要考慮到何時要執行這個策略。其次就是資源分配以及單位操作，在單位操作的部分目前已經有看到用有限自動機的方式做出不錯的成果，如果硬要用學習的方式去學會各種人類玩家所開發出來的操控方式是較為困難的；在資源分配的部分，一些檢視記錄檔的軟體都用比例的方式來顯示，我想這是一個不錯的方向。當這三個方面各自完成的時候，還需要一個階層式的設計來統合三個部分，尤其是資源的分配如何在質與量上做時機的取捨，都是未來需要進一步研究的目標，一旦這些目標完成了，新的人工智慧將會帶給玩家更多有趣的新體驗。

從遊戲的歷史來看，玩家的玩法在一定程度上會受到遊戲設計的影響，如果就即時

戰略遊遊戲來說，自從記錄檔被解碼與檢視的軟體誕生之後，玩家就不斷的熱烈討論著軟體上的數據所帶來的意義，有些玩家開始追逐所謂的手速，認為速度快的玩家才可以稱得上是高手；部分的玩家開始從解碼的指令檔中學習、模仿高手玩家的細部操作，在一些職業比賽中，記錄檔開始禁止外流，原因在於透露了太多職業玩家花費大量努力才研究出的新操作技術，觀眾只能透過影片的方式去欣賞比賽，一邊思考著究竟選手是怎麼做到那些匪夷所思的動作的。本篇研究所提出的兩個模型，如果廣泛的被玩家所運用的話，有可能會出現一些新的玩法、造成衝擊，玩家可以拿著自己的記錄檔搜尋看看跟心目中想要模仿的偶像有多大的相似度，想代打或是想隱姓埋名的玩家也會開始思考著怎麼樣改變自己的習慣讓我們的辨識模型找不出破綻。正如同其他遊戲設計一樣，設計者往往需要面對玩家的挑戰，很難有一個設計是毫無破綻的，如何透過難以一眼看出的特徵找出玩家更深層更難以偽造的習慣，是我們辨識模型進一步可以成長的地方。而在策略辨識模型上，我們還需要對盤面判斷做改進，目前是將建築、部隊、科技的權重一制化，但是玩家或許不是平等的去看待三個面向，不同的玩家可能有自己的比重，未來我們需要有一個自動調整權重的機制來提高我們的辨識效果。



## 參考文獻

1. Tyrrell, T., *Computational Mechanisms for Action Selection*. 1993, University of Edinburgh.
2. Tyrrell, T., *The Use of Hierarchies for Action Selection*. *Adaptive Behavior*, 1993. **1**(4): p. 387.
3. Decugis, V. and J. Ferber, *Action selection in an autonomous agent with a hierarchical distributed reactive planning architecture*. Proceedings of the second international conference on Autonomous agents, 1998: p. 354-361.
4. Bryson, J., *Hierarchy and sequence vs. full parallelism in action selection*. *Intelligent Virtual Agents*, 1999. **2**: p. 113–125.
5. Tzu, S., *The Art of War*. 2005: Digireads. com.
6. Laird, J.E., *Using a computer game to develop advanced AI*. *Computer*, 2001. **34**(7): p. 70-75.
7. Buro, M., *Call for AI research in RTS games*. Proceedings of the AAAI-04 workshop on AI in games, San Jose 2004, 2004: p. 139–142.
8. Buro, M., *ORTS: A hack-free RTS game environment*. Proceedings of the Third International Conference on Computers and Games, 2002: p. 156–161.
9. Molineaux, M. and D.W. Aha, *TIELT: A Testbed for Gaming Environments*. Proceedings of the Twentieth National Conference on Artificial Intelligence (Intelligent Systems Demonstrations), 2005.
10. Aha, D.W. and M. Molineaux, *Integrating learning in interactive gaming simulators*. *Challenges in Game AI: Papers of the AAAI*, 2004. **4**: p. 04-04.
11. Aha, D.W., M. Molineaux, and M. Ponsen, *Learning to win: Case-based plan selection in a real-time strategy game*. Proceedings of the Sixth International Conference on Case-Based Reasoning. Chicago, IL: Springer, 2005.
12. Taylor, T.L., *Power Gamers Just Want To Have Fun?: Instrumental Play In A MMOG*. Paper presented at the 1st Digra Conference: Level Up, 2003.
13. Chee, F., *Understanding Korean experiences of online game hype, identity, and the menace of the “Wang-tta”*. DIGRA 2005 Conference: Changing Views-Worlds in Play.
14. Wallace, B., *Gamer is royalty in S. Korea*, in *LA-Times*. 2007.
15. Maes, P., *Situated agents can have goals*. *ROBOT. AUTONOMOUS SYST.*, 1990. **6**(1): p. 49-70.
16. Lorenz, E.N., *The Essence of Chaos*. 1996: University of Washington Press.
17. Southey, F., W. Loh, and D. Wilkinson, *Inferring Complex Agent Motions from Partial Trajectory Observations*.

18. Aamodt, A. and E. Plaza, *Case-Based Reasoning*. Proc. MLnet Summer School on Machine Learning and Knowledge Acquisition, 1994: p. 1-58.
19. Leake, D.B., *CBR in Context: The Present and Future*. Case-Based Reasoning: Experiences, Lessons, and Future Directions, 1996: p. 3–30.
20. Schmidt, R., et al., *Case-Based Reasoning for Medical Knowledge-based Systems*. International Journal of Medical Informatics, 2001. **64**(2-3): p. 355-367.
21. Likhachev, M. and R.C. Arkin, *Spatio-temporal case-based reasoning for behavioral selection*. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, 2001. **2**.
22. Cheng, D.C. and R. Thawonmas, *Case-based plan recognition for real-time strategy games*. Proceedings of the Fifth Game-On International Conference, 2004: p. 36-40.
23. Molineaux, M., D.W. Aha, and M. Ponsen, *Defeating Novel Opponents in a Real-Time Strategy Game*.

