

銘傳大學

資訊傳播工程學系

碩士論文

使用詢問式支持向量機強化遊戲智慧之
即時戰略系統建置

The construction of a real-time strategy with
game intelligence enhanced by enquiry SVM

指導教授：陳鴻文 教授 蕭立人 教授

研究生：吳秉達

中 華 民 國 1 0 6 年 7 月 3 日

銘傳大學

資訊傳播工程學系碩士班

論文口試委員會審定書

本校 資訊傳播工程學系 碩士班 吳秉達 君
所提論文 使用詢問式支持向量機強化遊戲智慧之即時戰
略系統建置

合於碩士資格水準，業經本委員會評審認可。

口試委員：

(召集人)

江嘉坤
陳鴻文
李明哲

指導教授：

蕭志人

陳鴻文

系主任：

陳善志

中華民國 106 年 7 月 13 日

使用詢問式支持向量機強化遊戲智慧之 即時戰略系統建置

研究生：吳秉達

指導教授：陳鴻文 蕭立人

摘要

隨著人們對於電子娛樂喜好的不斷增長，坊間出現了許多電子遊戲的競技比賽，來滿足大家的娛樂需求；除了人類選手之間的比賽，近年來也產生了即時戰略遊戲(Real-Time Strategy Game)人工智慧對戰比賽。然而，即時戰略遊戲人工智慧的戰略決策系統，由於所處理遊戲資訊的不完整和即時性，仍舊一直被視為是個難題。

為了提昇即時戰略遊戲人工智慧的能力，本研究基於詢問式學習暨支持向量投影理論，嘗試設計了一套詢問式支持向量機(Support Vector Machine, SVM)演算法，來進行遊戲人工智慧的戰略決策規則之訓練。最後透過詢問式 SVM 訓練得出的邊界模型，與原有的專家規則模進行實際對戰，測得對戰勝率可達 63.5%，顯示出本系統學習的成效。

此外，透過遊戲歷程的統計與分析，也可觀察出詢問式 SVM 線性和非線性模型學習到的策略差異性，在對戰表現上是會分別對應到保守型及快攻型的遊戲風格。期望藉由本研究的成果，能夠提供即時戰略遊戲人工智慧的一種有效的新學習機制。

關鍵字：即時戰略遊戲、支持向量機、詢問式學習、人工智慧

The construction of a real-time strategy with game intelligence enhanced by enquiry SVM

Student: Ping-Da Wu Advisor: Hown-Wen Chen Li-Jen Hsiao

Abstract

With the growing popularity of electronic entertainment, there have been many electronic sports to meet everyone's entertainment needs; in addition to the game between human players, there are some new contests for the game intelligence of real-time strategy (RTS) in recent years. However, the strategic decision-making AI of RTS is still regarded as a difficult problem due to the incomplete and immediacy of game information.

In order to enhance the ability of RTS artificial intelligence, this study attempts to design an algorithm of support vector machine (SVM) based on enquiry learning and vector projection theory to carry out the training of game AI rules. Finally, actual 1000 challenge battles were held between the boundary model derived from SVM training against the original expert rules. As a result, the winning rate of up to 63.5% was measured to show the effectiveness of the proposed learning mechanism.

In addition, through the game history analysis, conservative and fast-break styles of game strategies were found individually from the game behaviors made by linear and Radial Basis models for Enquiry SVM. It is expected that an effective new learning mechanism for artificial intelligence in real-time strategy game can thus be provided.

Keywords: Real-Time Strategy Game 、 Support Vector Machine 、 Enquiry Learning,

Artificial Intelligence

誌謝

首先要感謝陳鴻文教授與蕭立人，首先陳鴻文教授願意將其所研究的詢問式支持向量機暨投影理論提供給本人研究使用，並且在研究主題的選擇上願意讓我嘗試電腦遊戲人工智慧研究，並且提供我許多問題解決的方向，以及論文本文的寫作指導。而蕭立人教授則是幫助我快速熟悉遊戲系統建置環境 Unity3D 的學習部分，讓我能順利的將遊戲建置完成。也感謝李明哲教授與江憲坤教授等各位口試委員願意為本次的研究進行口試審查，並且給予本次研究肯定。另外感謝資訊傳播工程學系對我的各種幫助以及專業技術的傳授，非常感謝系上的各位成員對我的教誨。



吳秉達 謹致於

銘傳大學資訊傳播工程所 碩士班

中華民國 106 年 6 月

目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
第一節 研究動機與背景.....	1
第二節 研究目的.....	2
第三節 研究限制.....	3
第四節 論文架構.....	4
第二章 相關文獻.....	5
第一節 即時戰略遊戲介紹.....	5
第二節 即時戰略遊戲的發展趨勢.....	7
第三節 即時戰略遊戲之人工智慧發展現況.....	10
第四節 遊戲建置平台之 UNITY3D 介紹.....	11
第五節 支持向量機.....	12
第六節 詢問式支持向量機暨投影理論介紹.....	14
第三章 遊戲設計與開發.....	18
第一節 本研究建置的戰略遊戲系統架構.....	18
第二節 本研究建置的即時戰略遊戲之參數設計.....	20
第三節 遊戲人工智慧之行為設計.....	26

第四節 詢問式 SVM 演算法	30
第四章 實驗方法及結果	34
第一節 即時戰略遊戲之專家規則設計	34
第二節 儲存資料設計	36
第三節 實驗結果	46
第四節 對戰歷程分析	67
第五章 結論與未來展望	72
第一節 結論	72
第二節 遭遇問題	74
第三節 未來展望	75
參考文獻	78
附錄	81



表目錄

表 2-1 即時戰略與回合制遊戲的差異	5
表 2-2 星海爭霸 2 版本比較表	8
表 3-1 本遊戲中三種部隊的屬性一覽表	20
表 4-1 採用不同紀錄分數門檻來篩選訓練資料之前測對戰勝率...37	
表 4-2 本研究採用之遊戲訓練資料的特徵說明一覽表	39
表 4-3 訓練資料格式 A.....	43
表 4-4 訓練資料格式 B.....	43
表 4-5 訓練資料格式 C.....	44
表 4-6 訓練資料格式 D.....	45
表 4-7 各種訓練資料格式前測比較表	46
表 4-8 各版本之線性分界模型訓練資料集	57
表 4-9 各版本之非線性分界模型訓練資料集	66

圖目錄

圖 2- 1 即時戰略遊戲戰略架構圖	6
圖 2- 2 雪娛樂遊戲爐石戰記	11
圖 2- 3 Unity3D 開發使用語言比例	12
圖 2- 4 線性 SVM 分割效果示意圖	13
圖 2- 5 線性投影	15
圖 2- 6 無雜訊之二分類別的訓練資料集	15
圖 2- 7 分類學習過程中採用了線性 SVM 之 SV 投影點	16
圖 3- 1 本戰略遊戲系統流程圖	18
圖 3- 2 本遊戲中基地進行生產部隊之示意圖	21
圖 3- 3 遊戲中採集資源活動之示意圖	22
圖 3- 4 遊戲中攻擊部隊作戰之示意圖	23
圖 3- 5 遊戲中遊戲地圖設定之示意圖	24
圖 3- 6 本遊戲內戰術與戰略關係圖	26
圖 3- 7 玩家部隊群概念圖	27
圖 3- 8 「世紀帝國 2」遊戲中的玩家分數示意圖	30
圖 3- 9 詢問式 SVM 訓練流程圖	33
圖 4- 1 遊戲人工智慧的決策模組	35
圖 4- 2 Libsvm 資料格式範例	38
圖 4- 3 參數最佳化工具 grid.py 的使用狀況	47
圖 4- 4 SVM 訓練獲得之初版線性模型內容	48
圖 4- 5 第一版投影點經 SVM 訓練所得之線性模型	49
圖 4- 6 第二版投影點經 SVM 訓練所得之線性模型	51

圖 4-7 第三版投影點經 SVM 訓練所得之線性模型.....	52
圖 4-8 第四版投影點經 SVM 訓練所得之線性模型.....	54
圖 4-9 各版本線性模型的勝率	56
圖 4-10 SVM 訓練獲得之初版非線性模型內容.....	58
圖 4-11 第一版投影點經 SVM 訓練所得之非線性模型.....	59
圖 4-12 第二版投影點經 SVM 訓練所得之非線性模型	60
圖 4-13 第三版投影點經 SVM 訓練所得之非線性模型	62
圖 4-14 第四版投影點經 SVM 訓練所得之非線性模型	63
圖 4-15 各版本非線性模型的勝率	64
圖 4-16 各版本非線性模型的預測準確度.....	64
圖 4-17 線性模型中攻擊部隊數量的變化歷程.....	67
圖 4-18 非線性模型中攻擊部隊數量的變化歷程.....	68
圖 4-19 線性模型中工作部隊數量的變化歷程.....	69
圖 4-20 非線性模型中工作部隊數量的變化歷程.....	69
圖 4-21 線性模型中雙方玩家分數評估的變化歷程.....	70
圖 4-22 非線性模型中雙方玩家分數評估的變化歷程.....	70
圖 5-1 星海爭霸 2 天梯宗師排行榜.....	73
圖 5-2 假設的投影點驗證程式流程圖	76

第一章 緒論

本章節透過遊戲人工智慧的相關例證，引出本論文的研究動機；並以即時戰略遊戲人工智慧發展的問題，來設定本研究之研究目的。最後闡述本研究的研究限制以及論文架構。

第一節 研究動機與背景

現今電子遊戲發展蓬勃，人們對於電子娛樂的需求越來越高，甚至產生了電子競技(electronic sports, e-sports)的新產業，雖然內容是以人類遊戲玩家對戰的競技性為賣點，但也同時促成了遊戲人工智慧的發展。從2010年開始，AIIDE (Artificial Intelligence and Interactive Digital Entertainment) [22]每年舉辦即時戰略遊戲人工智慧的設計比賽，其中部份比賽項目就是「星海爭霸」(StarCraft)。並且在多年，透過比賽選手與人工智慧對戰後的結果分析發現，遊戲人工智慧還是不敵人類遊戲高手，主要可歸因於，人類遊戲高手在戰略選擇能力較高的緣故[12]。

然而，隨著 Google DeepMind 所開發的圍棋遊戲人工智慧 Alpha Go 系統[21]，在2016年3月成功擊敗了韓國尖端職業棋士李世石，並造成了轟動，人們對於遊戲人工智慧的能力，開始更有信心和認知。同年暴雪娛樂公司(Bizzard Entertainment)總監 Tim Morten 也表示將提供 Google 公司，其名下遊戲「星海爭霸 2」相關的技術支援，宣布開始即時戰略遊戲人工智慧的相關研究計劃，企圖來挑戰即時戰略遊戲「星海爭霸 2」[11]。由此可窺見，即時戰略遊戲的人工智慧發展，即開進入一個新的紀元。

雖然即時戰略遊戲是一種複雜的競技項目，但還是能歸納出遊戲的核心概念，主要在於包含進攻、經濟及防守三種遊戲戰略的互動。在摒除雙方玩家的遊戲操作反應及熟練因素外，即時戰略遊戲的勝負將建立在玩家的對戰略選擇之適當性；越能適時判斷出遊戲的局勢需求，並成功挑選出對應之戰略，則玩家越容易獲勝[14]。這也促成了本論文，鎖定於進行即時戰略遊戲的策略機器學習之相關研究。

第二節 研究目的

本論文自行建置了一個即時戰略遊戲，以作為研究的訓練資料來源和測試環境。也基於陳鴻文教授[6]所提出的詢問式支持向量機(Support Vector Machine, SVM)投影理論，設計了一套詢問式 SVM 演算法，以作為可接受雜訊訓練資料的遊戲人工智慧學習機制；本研究將針對 SVM 中的線性與非線性兩種核函數，分別進行邊界模型訓練，並且以實際對戰的勝率，來驗證此套學習機制，應用在即時戰略遊戲的戰略規則之效果。故主要的研究目的計有：

- 一、設計及開發一個可真實運作的小型即時戰略遊戲。
- 二、基於詢問式學習暨支持向量投影理論[6]，設計了一套適合即時戰略遊戲的戰略規則學習，且可容忍雜訊訓練資料的 SVM 演算法。
- 三、利用兩個遊戲人工智慧模組實際對戰的方式，來驗證暨探討可容忍雜訊的 SVM 演算法的學習行為，並應用到遊戲戰略學習的細部機制設計(如訓練資料格式等)及學習效能。

第三節 研究限制

本研究之研究限制暨原因說明：

一、遊戲環境自行建置

由於市售的即時戰略遊戲無法提供完整的內容設定修改，因此本研究僅能將研究的環境，建立在自行開發的即時戰略遊戲之上；導致本遊戲系統須適度簡化，但仍需符合即時戰略遊戲的核心概念[3]。

二、簡化遊戲多媒體素材

由於進行遊戲系統開發人數的限制，與欠缺專業的美術專長，本研究放棄了精緻的遊戲多媒體設計，並以能夠觀察遊戲對戰過程變化為目標來進行建置。

三、時間因素

本研究人員因前期需適應遊戲的開發環境、與遭遇 Libsvm 工具[19]的模型檔案解讀錯誤等問題，不幸浪費了許多時間；另外，遊戲建置完成後，還需額外透過人工的方式，慢慢調整遊戲的諸多參數，以期達到即時戰略三種戰略相剋的核心概念。基於以上時間因素，本研究後期只夠完成進攻戰略規則之學習工作。

四、遊戲專家知識

由於本論文所使用的遊戲環境為研究者自行建置而成，因此遊戲所需的專家知識[17]，包括初始的遊戲人工智慧模組運作規則，暨詢問式學習過程的投影點類別修正工作，皆僅能自行透過參考其他即時戰略遊戲攻略，與不斷地實際實驗試錯而獲得。

第四節 論文架構

本論文共分為五章，第一章為「緒論」，介紹了本研究的研究動機、研究背景、研究目標及研究限制。第二章為「相關文獻」，內容分為即時戰略遊戲的定義介紹、即時戰略遊戲的發展趨勢、即時戰略遊戲之人工智慧發展現況與開發環境 Unity3D，以及本論文所使用的分類學習理論支持向量機的原理及兩種 SVM 核函數的算法，最後導入詢問式支持向量機暨投影理論介紹。第三章為「遊戲設計與開發」，詳細介紹了本研究遊戲系統的內容建置，其中包含遊戲系統架構、部隊參數設定原理、專家系統的建置概念，並透過詢問式支持向量機暨投影理論為原理，來設計適用於遊戲的詢問式 SVM 演算法。第四章為「實驗方法及結果」，解釋了訓練資料格式的設計理由，並透過實驗，驗證本論文所設計的詢問式 SVM 演算法之效果。最後透過觀察遊戲歷程的變化，試圖解讀進攻戰略規則透過則詢問式 SVM 演算法的改變。最後為第五章「結論與未來展望」，檢討了本研究的設計與研究成果，並提出可能的修改方式。

第二章 相關文獻

第一節 即時戰略遊戲介紹

即時戰略遊戲有別於傳統的競技類型戰略遊戲，前者採用即時遊戲反應系統，亦即考驗玩家短時間內必須做出判斷和執行動作的能力，如果在遊戲進行中的採集、建造、發展等過程中採用了回合制內容，則此款遊戲就不能被稱為即時戰略遊戲[1]。

表 2-1 即時戰略與回合制遊戲的差異

	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

資料來源：王正宏，2007

在市面上的即時戰略遊戲中，通常有幾種遊戲目標已被採用，例如摧毀所有對方部隊、摧毀敵方所有建築物、在時間內防守基地不被摧毀、摧毀特定目標等；近年來比賽項目較多的即時戰略遊戲「星海爭霸 2」，從 2010 年迄今皆出現在黃金聯賽(Gold Series Pro League ,GPL)、GSL(Global StarCraft II League)等聯賽中，代表該遊戲內容設定已廣為人知，故本研究採用的遊戲目標與「星海爭霸 2」同樣，皆設定為摧毀敵方所有建築物[5]。

即時戰略遊戲中的系統元素可分為建築物、工作部隊、攻擊部隊及資源點，遊戲內的建築物通常有一個主要的基地建築生產工作部隊，並且讓工作部隊採集遊戲中的資源，以提供玩家發展國力，並可生產攻擊部隊，最後擊敗對手[3]。因為即時戰略遊戲考慮到空間跟時間限制(例如:對方國力發展的時間、我方與敵方攻擊部隊的距離與在地圖上的位

置)，一般都是透過玩家在短期時間內一連串的決策與執行動作，來影響到遊戲進行的勝負。

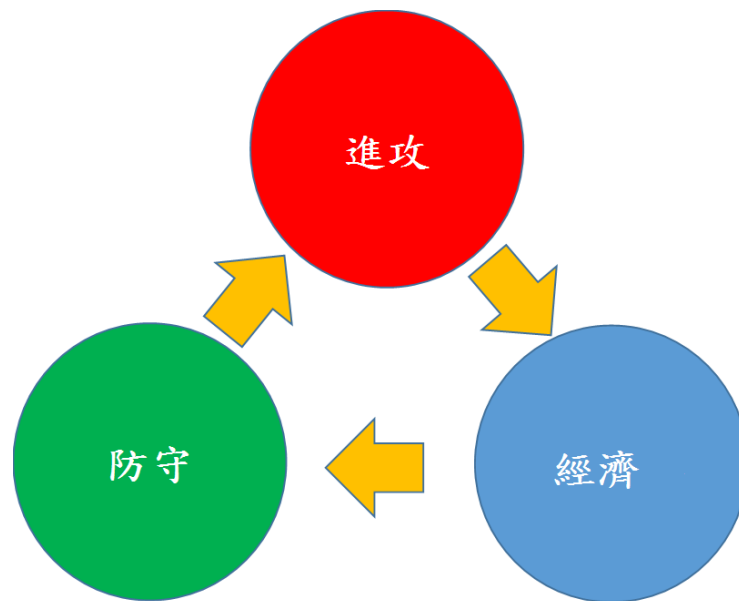


圖 2-1 即時戰略遊戲戰略架構圖

資料來源：幻想戰略譚，2017

雖然即時戰略遊戲種類眾多，但是在遊戲概念上具有共通的遊戲戰略，可區分為如圖 2-1 內的三種戰略行動，且三者會造成一個互相剋制的循環。接下來解說這三種戰略行動的核心內容[9]:

一、進攻戰略

進攻戰略的核心意義在於進行快速的攻擊，捨棄了發展其他經濟國力的機會，企圖以最快的速度生產攻擊部隊來打擊對手。會考慮採用進攻戰略的遊戲戰術，通常是針對以經濟發展為主的對手玩家，因為對手在遊戲初期會因為投入大量資源發展經濟國力，導致沒有多餘的資源來生產防守部隊，所以採取快速進攻的遊戲戰略，一般能順利剋制採取經濟戰略的對手。

二、經濟戰略

經濟戰略的核心意義在於快速發展經濟國力優勢，在經過一定遊戲時間的孕育之後，往往能創造龐大的部隊來摧毀敵人。雖然全力發展經濟國力，有時會導致發展中途時玩家的攻擊部隊數量不足，但是只要能夠順利發展起較對手為高的經濟國力時，就能以龐大資源的優勢來戰勝對手。因此，採取經濟戰略，對於採取遊戲初期不進攻的防守戰略對手，將會產生剋制的效果。

三、防守戰略

防守戰略的核心意義在於利用戰鬥地理位置上的優勢，來成功防守敵方的進攻後，往往能夠獲得兵力上的優勢；之後即可轉而透過敵方的空隙，進行反擊來擊敗對手。原因為即時戰略遊戲的防守方，皆會具有作戰優勢(例如:攻擊方的攻擊部隊需要往前補給的距離優勢、遊戲本身設計的建築物防守機制等)，所以對付採取積極攻擊的進攻戰略對手，會產生剋制的效果。

第二節 即時戰略遊戲的發展趨勢

即時戰略遊戲在西元 1992 年 Westwood Studios 工作室製作的「沙丘魔堡 2」(Dune II)建立了遊戲類型的基礎後，發展至今已有 25 年的歷史[8]，在巔峰期也出現了許多即時戰略遊戲的知名作品，包括了暴雪娛樂推出的「星海爭霸」(StarCraft)、「魔獸爭霸」(WarCraft)，以及微軟推出的「世紀帝國」(Age of Empires)，與 Westwood 之後推出的「終極動員令」(Command & Conquer) 系列作品，皆聞名遊戲界[3]。

從西元 2000 年 10 月韓國龍仁市開始主辦世界電子競技大賽 WCG(World Cyber Games)，此賽事延續至西元 2013 年，其中每一屆的 WCG

都有即時戰略遊戲的比賽項目，數量最多，且比賽項目大宗為「世紀帝國」系列、「魔獸爭霸」系列及「星海爭霸」系列。其中世紀帝國系列從第一屆開始比賽，一直延續到西元 2008 年的 WCG，都有參與項目；而魔獸爭霸與星海爭霸系列，則是延續到西元 2013 年的 WCG。由此可窺見即時戰略遊戲項目，在 WCG 影響之遠大。

然而 WCG 在西元 2013 停辦之後，暴雪娛樂則是開始經營世界總決賽 WCS(World Championship Series)。此賽事以主打暴雪娛樂旗下的各種電子競技遊戲為目的，至今即時戰略項目則以「星海爭霸 2」做為主要項目之一；所帶來的影響以「星海爭霸 2」為例，從西元 2010 年 7 月該遊戲開始發售，迄今已進行過兩次重大資料片改版，改版內容皆在即時戰略遊戲發展歷史上，具有重大意義，詳如表 2-2 所示。

表 2-2 星海爭霸 2 版本比較表

項目 版本	自由之翼	蟲群之心	虛空之遺
遊戲部隊種類	44	50	56
初始工作部隊數	6	6	12
資源點平均產量	1500	1500	1125
資源挖掘速度	慢	慢	快
平均比賽地圖	小	大	大
地圖地形機制	防守優勢	防守優勢	進攻優勢
比賽時間常態	中	慢	稍快

參考攻略網站上的討論[13]，本研究整理出表 2-2，但由於其中部份項目小型遊戲改版頻繁，就缺乏明確數值可供比較。作為電子娛樂熱門

競賽項目，「星海爭霸 2」會依照比賽觀眾喜愛的對戰風格發展改版；從表 2-2 中可發現從 2010 年發售的「星海爭霸 2:自由之翼」(StarCraft II: Wings of Liberty)，比賽用的遊戲地圖普遍較小，並且一開始的工作部隊數量，較最新版本的「星海爭霸 2:虛空之遺」還要少。然而，較小的地圖容易限制比賽選手的戰術發揮，導致比賽選手常用快攻的方式作為遊戲戰略，並且使比賽時間普遍過短，導致比賽精彩的部份較少。於是在西元 2013 年 3 月推出「星海爭霸 2:蟲群之心」(StarCraft II: Heart of the Swarm)，版本內修改了「星海爭霸 2:自由之翼」的比賽取向，將遊戲地圖設定為較大，讓比賽內容可以比較容易推至大規模的國力發展，以提供選手在對戰時發揮諸多的戰術。但是因為資源點平均產量沿用「星海爭霸 2:自由之翼」的 1500 晶礦，加上比賽場地的擴大與防守優勢地形，導致比賽容易陷入僵局狀態；最著名的比賽為 2014 年韓國星海戰隊聯賽 SPL(StarCraft ProLeague)[23]中，比賽選手 Reality 與 SoulKey 進行長達遊戲時間 2 小時 20 分鐘的對戰，這在即時戰略遊戲歷史是非常特別的事件，然而現場觀眾的反應反而不佳。

暴雪娛樂集團在吸取「星海爭霸 2:蟲群之心」的經驗後，開始調整遊戲系統趨向，使其步調加快，並且在西元 2015 年 11 月推出新資料片「星海爭霸 2:虛空之遺」(Legacy of the void)。改版後的初始工作部隊數量與資源採集速度大幅提升，讓遊戲的進行速度有飛越性的成長，並且在地圖機制上，下降了資源點的產量與增加許多地圖的進攻地點，以鼓勵雙方交火，進而提升整體比賽的娛樂效果。由此可見，即時戰略遊戲發展至今，已經開始由傳統節奏較慢，並且賽時較長，發展為鼓勵雙方交火的快節奏比賽。透過以上觀察，導致本研究的即時戰略遊戲系統，著重在中小地圖上，增強進攻戰略適用時機的探索。

第三節 即時戰略遊戲之人工智慧發展現況

自 2010 年開始有即時戰略遊戲相關的 AI 研究機構，舉辦遊戲 AI 設計比賽，例如 AIIDE(Artificial Intelligence and Interactive Digital Entertainment) 舉辦過數次有關知名即時戰略遊戲「星海爭霸」(StarCraft)的 AI 設計比賽 [22]，其中包含了人工智慧之間的對戰，以及人工智慧與真實玩家的對戰等模式。由於即時戰略遊戲是屬於一種複雜的遊戲結構，其中包含了行為的隱藏資訊、廣闊的區域活動及快速的指令執行等，所以對於人工智慧研究者而言，仍是一種巨大的挑戰[12]。

從即時戰略遊戲的人工智慧設計，與真實遊戲玩家對抗的優劣績效項目中分析，遊戲人工智慧的優勢在於比人類玩家更快速下達指令的能力，然而就現實情況而言，這樣的優勢並無法讓人工智慧具有足夠能力，來擊敗頂尖的即時戰略遊戲玩家：因為在即時戰略遊戲中，操作部隊的精細度固然會影響到遊戲的勝負，但真正成就玩家勝負的主要關鍵因素，仍在於對局勢的判斷後的戰略選擇；關於這方面，人類玩家利用經驗及跳躍思考的方式，仍絕大部份遠勝過現在的遊戲人工智慧[22]。

即使如此，2016 年 Google 的圍棋類遊戲人工智慧 Alpha Go，成功以四勝一負的戰績，擊敗了南韓圍棋界的棋王李世乭，這成為人類與遊戲人工智慧對戰史中的一個重要事件；在此之後，暴雪娛樂公司總監 Tim Morten 也在 2016 年 3 月世界總決賽 WCS(World Championship Series)中國站的決賽上表示[11]，確認接下來該公司將會設計遊戲人工智慧，來挑戰即時戰略遊戲「星海爭霸 2」的人類高手。

第四節 遊戲建置平台之 Unity3D 介紹

Unity3D 為目前諸多上市櫃遊戲廠商、與獨立開發遊戲者，愛用的遊戲程式開發環境，已有許多以遊戲開發為訴求的基礎功能，可供使用：例如：A-star 路尋功能、使用者 UI 介面、影格更新率（Frames Per Second ,FPS）等多種功能，皆可直接呼叫使用。這樣的特性，足以滿足絕大部份開發者，對於遊戲建置環境的需求。例如美國知名廠商暴雪娛樂集團旗下，就有實際使用 Unity3D 為基底的商用線上遊戲「爐石戰記」，在上市後獲得了諸多好評，並且在西元 2014 獲得 Unity 全球大賽金立方獎[10]，可作為 Unity3D 應用在遊戲產業的經典例子。



圖 2-2 雪娛樂遊戲爐石戰記

此外，Unity3D 具有跨多元平台的能力[10]，可用於開發 Windows、MacOS、Linux 遊戲，以及 IOS、Android 為作業系統的行動裝置遊戲，也可用於開發 PlayStation、XBox、Wii 主機上的遊戲；並且 Unity3D 只需要網頁瀏覽器支援 HTML5 技術，更可用於網頁線上遊戲的開發工作。

Unity3D 目前支援的開發程式，常用 C#與 JAVA script 兩種程式語言；且從 Unity3D 官方網站統計的數據可以得知，其中採取以 C#作為底層，開發出的 Unity3D 遊戲比例較高。本論文所開發的遊戲環境，也使用了

Unity3D 的 C#語法，因此可以獲得較多的開發資源，以支持本系統的建置工作。

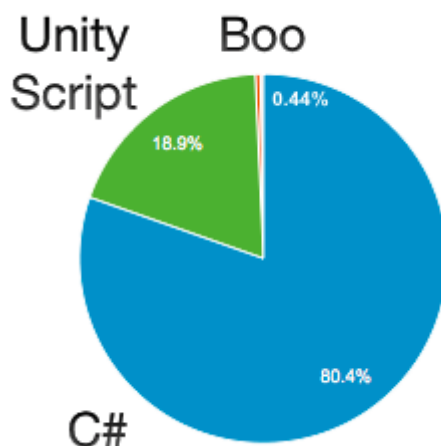


圖 2- 3 Unity3D 開發使用語言比例
資料來源：Unity3D 官方網站，2017

第五節 支持向量機

在現行諸多分類方法中，支持向量機(Support Vector Machine,SVM)已在許多應用中，呈現出非常良好的分類能力[7]。SVM為監督式學習的一種，是利用超平面(hyperplane)將資料分割成兩種類別的分類器；運作時，SVM將訓練集內的各筆資料，透過學習找出重要的支持向量，據以形成邊界模型，並利用此邊界模型對於新的輸入資料進行類別預測。一般對於訓練集中的資料 x_i ，定義其目標類別 Y_i 的方式如公式 2-1:

$$Y_i = \begin{cases} Y_i > 0, x_i = \text{ClassA} \\ Y_i \leq 0, x_i = \text{ClassB} \end{cases} \quad (\text{公式 2-1})$$

SVM 的訓練過程，即嘗試找到所有特殊的訓練資料點，這些又稱為支持向量點，其落於與兩類資料分隔距離最大的邊界(如圖 2-4 中虛線)平行的邊界線上(如圖 2-4 中兩條實線)，故可讓 SVM 的分類準確性達到

理論上的最佳效果[2]。將 SVM 模型中的支持向量帶入核函數中，再乘上對應權重後累加，即形成如同公式 2-2 的分割邊界模型，詳細的說明如後述；之後將資料的輸入訊號帶入分割邊界模型中運算，利用輸出數值的正負值判斷，即可決定該筆資料的輸出歸屬類別。學習過程中可反覆嘗試調整訓練參數，或是修改訓練向量集的內容，都可能會影響到分割邊界模型的形成。

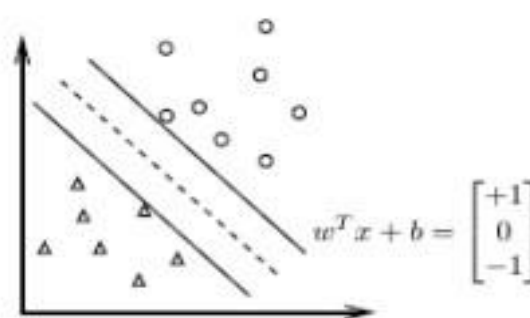


圖 2-4 線性 SVM 分割效果示意圖

資料來源：王榮英、李瑋翰，2012

在本研究採用林智仁(Lin Chih-Jen)教授所開發的 Libsvm[19]程式套件中，就包含了多種可套用的 SVM 分隔邊界模型(公式 2-2)中的核函數 K ，如 Linear、Polynomial、RBF(Radial Basis Function)等；公式 2-2 中的 b 代表了判決函數偏置項，用來定義分割邊界模型在空間中的位置，且在 SVM 的模型文件中以 ρ 數值代表。至於 X_j 為支持向量，將支持向量帶入核函數中，再乘上對應的權重值 w_j ，即形成一個邊界模型。

$$\text{plabel} = \text{sgn}(\sum_{i=0}^n W_i K(x_i, x_j) + b) \quad (\text{公式 2-2}).$$

如果使用線性模型進行訓練，則分界線公式需要套用公式 2-3 之線性核函數來進行分類運算：

$$\text{linear: } K(X_i, X_j) = X_i X_j \quad (\text{公式 2-3})$$

若使用非線性(如 RBF)模型進行訓練，則邊界模型需要套用公式 2-4 之 RBF 核函數來進行分類運算：

$$\text{RBF:}K(X_i, X_j) = \exp\left(-\gamma \left\|X_i - X_j\right\|^2\right), \gamma > 0 \quad (\text{公式 2-4})$$

第六節 詢問式支持向量機暨投影理論介紹

由於 SVM 的特色是標示出距離分類邊界 (boundary line)最近的一些訓練樣本，即所謂的支持向量 SV，並據以形成距離兩類樣本等距的最佳分類邊界模型，所以 SVM 對於 SV 具有的雜訊也會特別的敏感。考量在現實遊戲環境下，由於研究人員所蒐集到的訓練樣本常會具有測量誤差或雜訊干擾，故需額外引用如下的特殊機制來加以處理雜訊訓練資料。

任何分類器模型(包含 SVM)的學習過程，皆可視為類別間分隔邊界不斷調整的過程；且過程中分隔邊界所在的位置，正代表是目前兩類資料最難辨識其類別之所在。所以若能不斷地釐清目前分隔邊界點應該歸屬的類別，就能不斷地微調分界邊界，直到形成最精準的分隔邊界為止。但由於分隔邊界是由無限多個點來形成的，如何有效地取樣並學習到其中的特徵點，便成為分類器訓練過程的重心。

數學投影在線性代數和泛函分析中，為從向量空間映射到投影面(線)的一種線性轉換，造成投射點連接投影點後，可與投影線(面)間形成垂直(正交)關係，亦即投射點與投影點兩點連接成具有最短距離的線段；換言之，投影點就可被視為對應投射點之最靠近投影面(線)的特徵點，如圖 2-5 所示。將這個投影特性應用在 SVM 產生的支持向量及分界線模型上，就可以產生落在目前 SVM 分界線模型上，且對應各個支持向量的投影特徵點；若再能設法透過專家諮詢的協助，賦予該投影特點的正確

類別資訊，即可做為後續微調分類分界線模型的重要依據。為了介紹應用投影理論與詢問式學習法在 SVM 訓練上發生的效果，在此引用了陳鴻文教授的論文[6]中，描述 SVM 線性模型分界線的投影實驗結果來進行介紹。

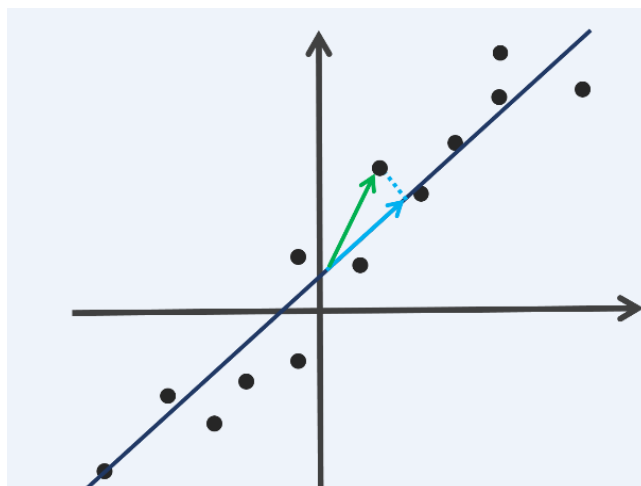


圖 2-5 線性投影

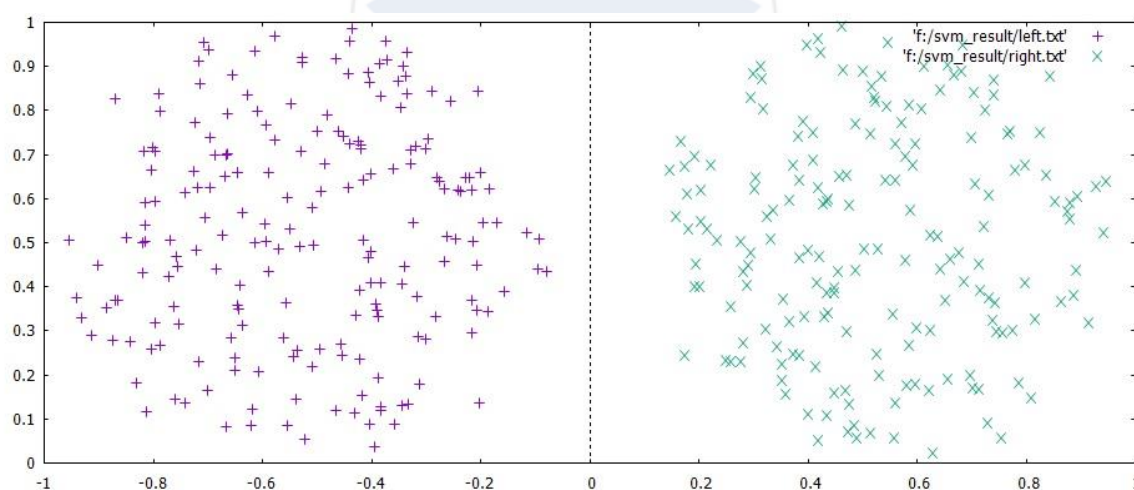


圖 2-6 無雜訊之二類別的訓練資料集

圖 2-6 描述了一個簡單二類別的訓練資料集，其中每一個資料點的特徵值直接以二維座標來表示，正負 x 座標值直接決定其歸屬類別。此資料透過 SVM 學習後，其 SV 對應的線性投影點會集中在非常鄰近中間 y 軸(圖中虛線)上，如圖 2-7 所示。明顯地，若能釐清並學習到這些位於

目前邊界上，卻混雜兩種類別的投影點之真實類別，將有助於獲得更接近最佳分界線的新邊界模型。

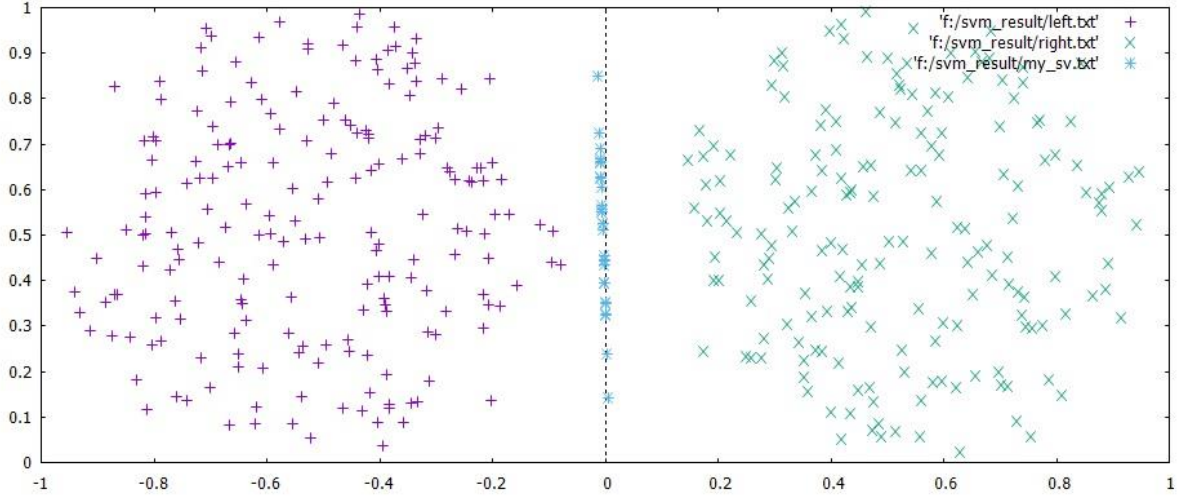


圖 2-7 分類學習過程中採用了線性 SVM 之 SV 投影點

以下公式詳細理論及推導過程請參閱「詢問式支持向量機之推導與應用」[6]，一般分類邊界直線方程式可表示為 $L: a_0 + a_1x_1 + \dots + a_nx_n = 0$ ，其中 $a_i = \sum_{k=1}^m \theta_k l_{ki}$ ， $a_0 = -\rho$ (ρ 是列在 SVM 模型檔案的重要參數之一)。若將分類邊界上對應每一支持向量 SV_p 之投影點 SV_p' 以特徵向量 $(lp_1', lp_2', \dots, lp_n')$ 表示，根據線性投影所形成向量應與直線法向量平行關係，可得 $(lp_1' - lp_1, lp_2' - lp_2, \dots, lp_n' - lp_n) = t * (a_0, a_1, \dots, a_n)$ ， t 為實數，將投影點座標帶回直線方程式，可得 t 值後，故可得對應支持向量之投影點 SV_p' 的座標值為：

$$l'_{pi} = l_{pi} + t * a_i = l_{pi} - a_i * (a_0 + \sum_{k=1}^n a_k l_{pi}) / (\sum_{k=1}^n a_k^2) \quad (\text{公式 2-5})$$

至於在非線性支持向量機模型下，是無法完全運用類似一對一的投影觀念，故需要給定不同的近似假設來進行推導。非線性分類邊界一般可表示為 $\theta_0 + \sum_{k=1}^m \theta_k f_k$ ， $\theta_0 = -\rho$ ，其中支持向量 SV_p 且假設採用的 Kernel function 為 RBF $f_p(x) = e^{-\gamma|x-l_p|^2}$ 。

若假設邊界點 X_p 與其他支持向量 SV_q (特徵向量 l_q) 的距離，可視為近似直角三角形 SV_p ， X_p ， SV_q 的斜邊，可推得：

$$\delta = \sqrt{\frac{1}{n\gamma} * \ln \left\{ \frac{[\theta_0 - \Delta(SV_p)]}{\theta_0} \right\}} \quad (\text{公式 2-6})$$

$$\Delta(SV_p) = \theta_0 + \sum_{k=1}^m \theta_k * f_k(l_p) \quad (\text{公式 2-7})$$

則新投影點座標：

$$l'_k = l_k \mp \delta \quad (\text{公式 2-8})$$

又若假設邊界點 x_p 與其他支持向量(非 SV_p)的距離夠大，故 x_p 在計算與其他支持向量的距離時，可用支持向量 SV_p 來取代(存在些許誤差亦無妨)，可推得：

$$\delta = \sqrt{\frac{1}{n\gamma} * \ln \left\{ \frac{\theta_p}{[\theta_p - \Delta(SV_p)]} \right\}} \quad (\text{公式 2-9})$$

則新投影點座標 $l'_k = l_k \mp \delta$ ，其中加減號可依計算結果，自行選取較合宜者。

第三章 遊戲設計與開發

本章將介紹本論文所建置的即時戰略遊戲內容，其中包含初始的遊戲人工智慧專家模組、系統架構、遊戲部隊參數設定、遊戲地圖設計、及 SVM 學習與實驗步驟。

第一節 本研究建置的戰略遊戲系統架構

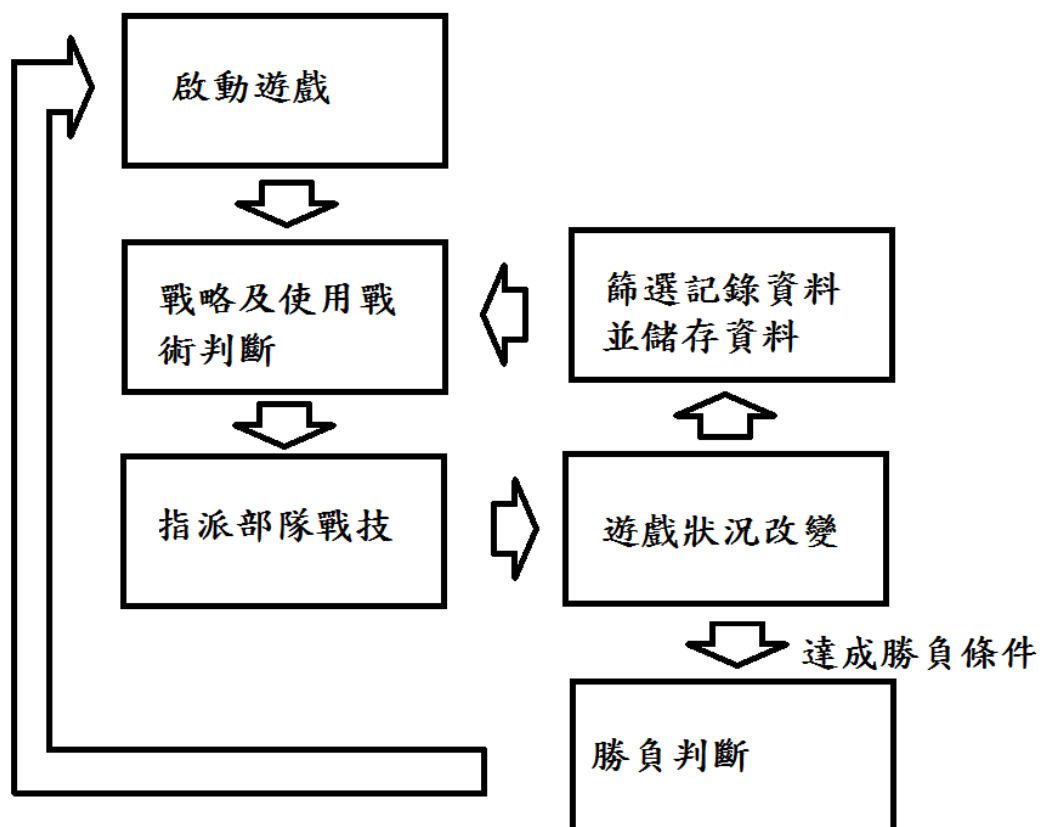


圖 3-1 本戰略遊戲系統流程圖




如圖 3-1，遊戲主程式啟動後遊戲人工智慧模組會進行第一次戰術選擇，且此次戰術選擇將透過完全隨機的方式進行，目的是為了模仿人類玩家在遊戲開局時對戰術的博弈[1]情況，與促使遊戲的情況發展具有多樣性，這樣收集到後續支持向量機的遊戲訓練資料所能包含的訊息，也會比較全面；至於之後的戰術選擇步驟，則是透過專家規則(詳見附錄)以及少部份的隨機方式來進行[18]。在遊戲人工智慧模組決定好採取的戰術後，將會對遊戲中所控管方的部隊發送指令，透過各種戰術目標指派給我方部隊各種戰技行為，這些戰技行為在下一節將會有著更為詳細的說明。

系統在每隔固定時間的推進後，遊戲人工智慧會進行一次目前遊戲局面的評估；也就是透過各種遊戲狀態評估分數，來進行前一個時間點所採用的戰術，迄今的成效評估(分數變化)，及決定接下來要進行的戰術選擇。過程中，如果遊戲狀態評估分數的變化，達到預設的門檻標準，則記錄該次戰術下敵我雙方的狀態評估分數，以期做為後續 SVM 分類器的訓練資料。透過這種不斷地戰術選擇與指派部隊戰技行為的方式，可讓兩個遊戲人工智慧模組在遊戲內互動對戰，最後達到遊戲勝負判斷的條件，即可完成該場遊戲；如有需要，則再重新開啟新的遊戲局面，以繼續蒐集新的 SVM 訓練資料。

第二節 本研究建置的即時戰略遊戲之參數設計

本研究透過建置一款新的即時戰略遊戲，來探索如何增強遊戲的人工智慧之能力。為了方便研究，本系統保留了即時戰略遊戲中，最為核心的部隊基本屬性特色[1]，但簡化了同類型遊戲內的其他內容。例如所創建的遊戲部隊類型，僅包含表 3-1 中的三種部隊，並詳細說明如後：

表 3-1 本遊戲中三種部隊的屬性一覽表

	基地部隊	攻擊部隊	工作部隊
圖示			
生命力	1500	175	70
攻擊力	12	18	6
防禦力	2	2	0
移動速度	0	1	0.75
攻擊範圍	大	小	小
生產消耗資源	1200	450	300
生產時間	45 秒	18 秒	11 秒

一、基地部隊

在遊戲內代表玩家重要的勝負條件之部隊，基地部隊可以透過消耗資源，來生產其他種類的部隊，也包含可以生產其他基地部隊，以擴展基地數量；至於在資源採集方面，則扮演資源回收點的角色。本遊戲系統的基地部隊特性，乃是參考知名即時戰略遊戲「終極動員令」系列的基地車概念[20]，讓基地生產出可以移動的基地部隊，並且在額外的資

源點變形成基地建築，且基地建築設定成具有少量反擊敵方部隊的能力。

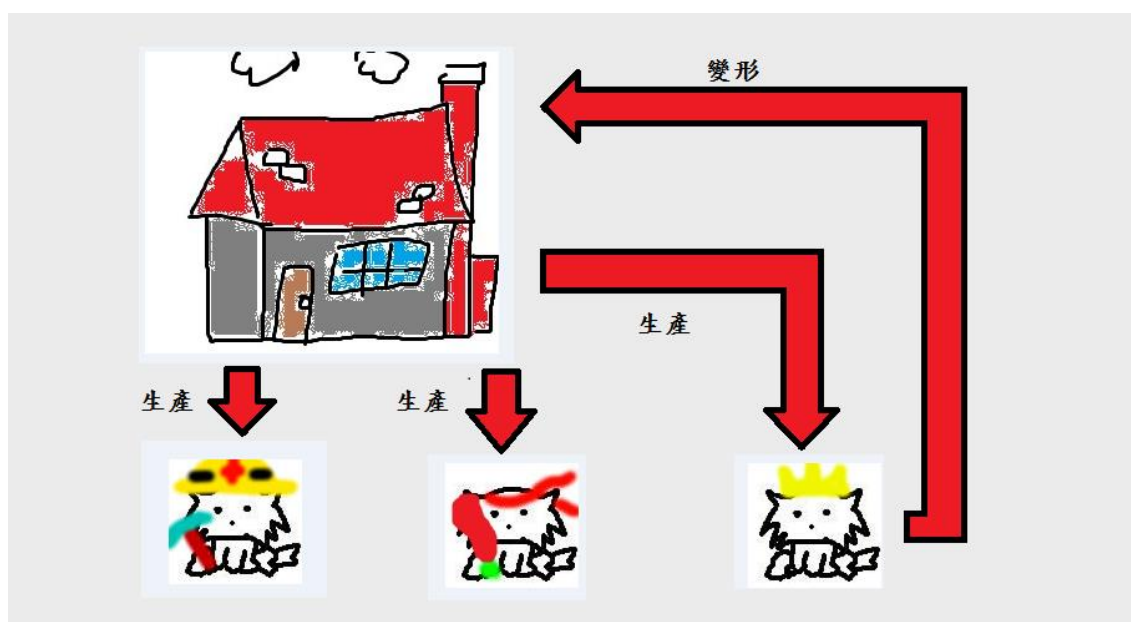


圖 3-2 本遊戲中基地進行生產部隊之示意圖

另外，基地部隊也可以花費 300 單位的資源，來升級生產部隊的速度:本遊戲系統提供 2 種高級等級，每個等級可以減少 4 秒的基地部隊生產需要時間，與 2 秒的其他種類部隊生產需要時間。這樣的升級設計，可以在經濟狀況允許的情況下，增加部隊生產的效率。

二、工作部隊

在各種即時戰略遊戲中，資源採集的目的，皆是為了提供玩家生產更多部隊的籌碼。資源採集在不同遊戲內，進行的方式可能不同，本遊戲系統所採用的是由基礎單位之一的工作部隊，負責採集資源點上的資源，並且繳回到基地內，方能增加玩家的資源數量。每隻工作部隊一次的採集，都可以增加玩家資源 20 點，並且工作部隊所具備的戰鬥能力，是遊戲中所有部隊類型內最差的。



圖 3-3 遊戲中採集資源活動之示意圖

採集資源在遊戲中設定為玩家必需進行的經濟活動，資源點為遊戲中可採集資源的區域，如圖 3-3 中的情況，玩家必須透過工作部隊在資源點收集資源後，再傳回給基地的方式來獲取資源。敵我雙方分別設置了三個資源點，但是其中兩個資源點，是預設沒有配備基地來進行採集；玩家必需在這兩個資源點上，建造新的基地後，方可進行採集，這在後續的地圖介紹會有更為詳細的說明。本遊戲的重要活動之一，就是透過建立更多資源點基地，以增加玩家的經濟能力，方能有足夠資源來生產更多部隊。

三、攻擊部隊

攻擊部隊為即時戰略遊戲內的核心部隊類型，攻擊部隊的作戰為從事軍事活動的核心，該部隊專門用於作戰，擁有最佳的戰鬥能力，擅長於摧毀敵方的部隊或是基地。



圖 3-4 遊戲中攻擊部隊作戰之示意圖

在遊戲中，將攻擊部隊運用在各種不同的場合，將產生不同的意義：如圖 3-4 中的情況，朝向敵人基地移動進行作戰，其遊戲意義為從事攻擊；另外，攻擊部隊在主基地之間進行巡邏，其遊戲意義則為從事防守，這部份會在後續的戰技介紹中補充說明。

四、地圖介紹

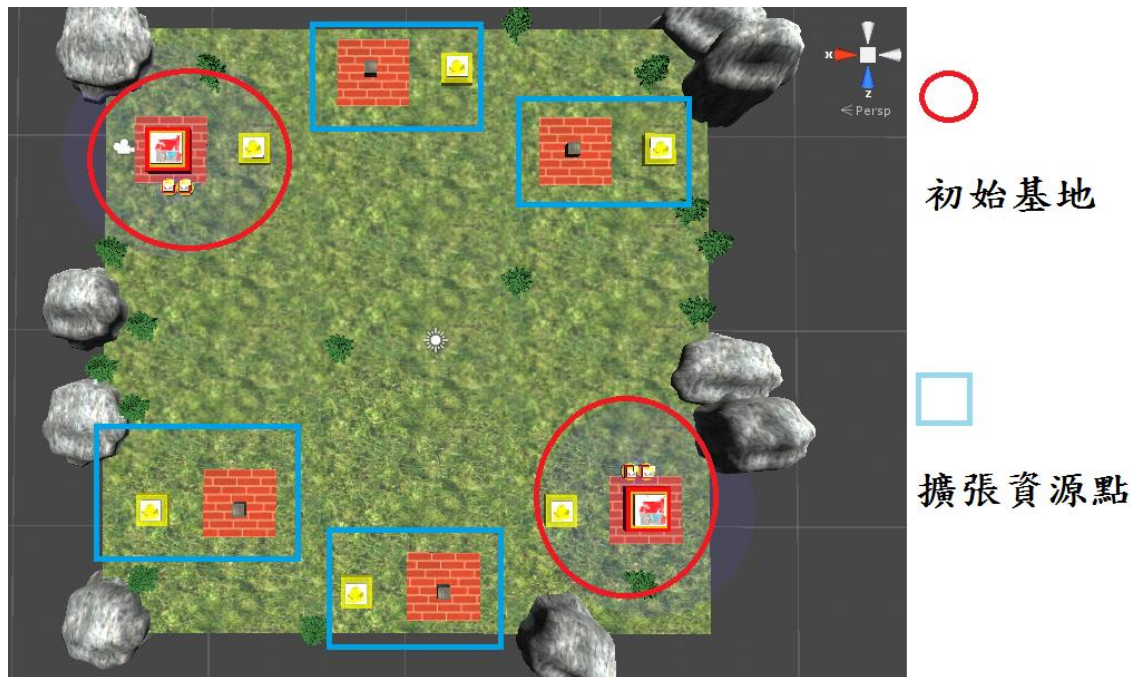


圖 3-5 遊戲中遊戲地圖設定之示意圖

本遊戲系統所採用的對戰地圖如圖 3-5，圈內的部份分別代表兩位玩家起始的出現位置，方框的部份為玩家可以從初始基地以外，設置新基地以增加資源採收的區域。地圖資源點的系統設計，是參照了暴雪娛樂「星海爭霸 2」的遊戲模式，在職業線上比賽的賽評口中，常有一礦、二礦或三礦這樣的說法，以表示選手目前的經濟狀態及地圖擴張的程度 [13]，而本系統則利用這樣的觀念，設計了一位玩家，最多可以擁有三塊資源採集點作為資源生產來源。

五、部隊參數調整

本遊戲系統在初期設定的參數並不能合理的展現即時戰略遊戲的三種戰略的相剋意義，在初期設計的參數設定下，本研究的基地攻擊力過強且攻擊部隊的攻擊力過低的情況下，容易出現防守方過於有利的情況，在這種參數設計下任何快攻的玩法將會沒有效果，也破壞了即時戰

略遊戲內進攻戰略剋制經濟戰略的意義，為了平衡遊戲，本研究修改了基地部隊的攻擊力以及攻擊部隊的攻擊力，基地部隊的攻擊力從原本的 18 修改為 12，並將攻擊部隊的攻擊力 14 修改為 18 以達到平衡，在修改之後進攻戰略的戰略價值提升。

第二次修改參數則著重在攻擊部隊在進攻的方基地時，對工作部隊的威脅能力。在之前的版本中，工作部隊的生命值為 150 點與攻擊部隊的 200 生命值僅相差了 50 點，因此攻擊部隊要消滅工作部隊所需花費時間較多，這代表攻擊部隊在承受基地部隊的攻擊與敵人建造新的攻擊部隊的時間較長，在無法完全摧毀敵人基地的情況下所能打擊對手經濟發展的效果較差。在多次觀察及修改之後，工作部隊的生命值從原本的 150 點修改為 70 點較符合一般即時戰略遊戲的情況。

在修改以上的部分後，容易出現攻擊部隊因為基地部隊的生命力過高，導致雙方基地把所有攻擊部隊及工作部隊都消滅的平手狀態，因此下降基地部隊的生命力，在第三次修改遊戲部隊參數時將基地部隊原有的 2500 生命力修改為 1500，並且有效下降雙方平手導致遊戲無法進行的情況。

第三節 遊戲人工智慧之行為設計

本研究將遊戲人工智慧分類成三個層面，分別為戰略、戰術與戰技，三層的分類可以較有層次地[15]，分類控制遊戲人工智慧的行為，並且讓遊戲人工智慧訓練更有條理。以下分別介紹，在本遊戲中戰略、戰術及戰技的設計。

一、本系統戰術定義：

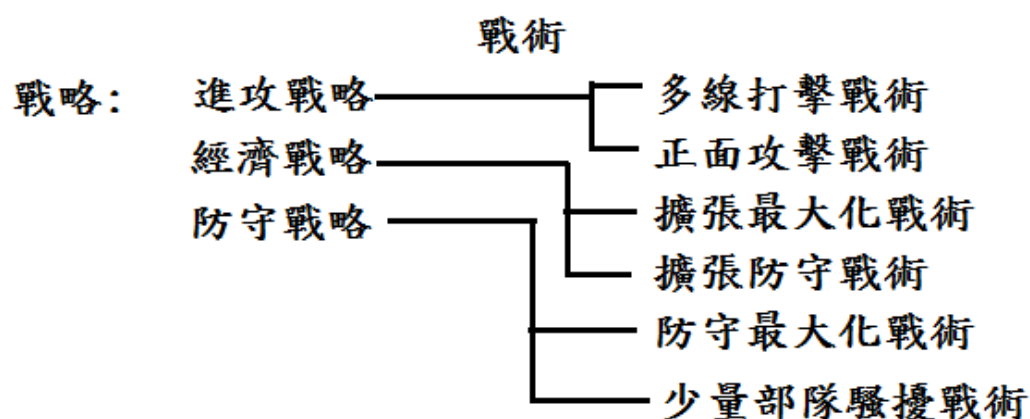


圖 3-6 本遊戲內戰術與戰略關係圖

如同本章第一節的本戰略遊戲系統流程圖(圖 3-6)所示，本遊戲系統建立在三種核心即時戰略的運作規則下，其中通常可以執行一種以上方法，來達成玩家選擇的戰略目標，本研究將這些方法定義為戰術；在戰術層面下，玩家設計了一套操作部隊的流程或是規則來達成目標，就定義為戰技。而戰術的實際內容，常包含了指派各個部隊群進行特定戰技行動，或遇到狀況時，部隊需要作出的反應動作。戰技的部份，將在後續的戰技定義內，詳細介紹。

二、本系統中戰技定義:

即時戰略遊戲中有各種不同種類的部隊，由於遊戲設計時賦予功能的不同，所以各種部隊會有不同的行動可以指派，就算是指派相同的行動，也會因為應用的方法不同，而產生不同的效果。例如：指派攻擊部隊，移動到敵人的基地進行作戰，這樣的行為預期可以對敵人的基地造成損傷，因此可稱此動作隸屬於攻擊戰技；相對地，如果指派攻擊部隊，移動到我方的基地準備作戰，這樣的行動預期可以在敵方攻擊基地前，就先準備好對抗敵人，因此可稱此動作隸屬於防守戰技。

此外，在實際的即時戰略遊戲操作中，要逐一指派各個部隊的戰技會比較繁瑣耗時；因此，一般的即時戰略遊戲會提供編成部隊群的功能，讓玩家得以一次操作多個部隊或多種部隊，以協助玩家操作龐大的戰局，如圖 3-7 所示。當然，有時候為了達到戰術目標，玩家也會將數量較大的部隊群，拆散成多個小的部隊群，以利分別進行不同的戰技。



圖 3-7 玩家部隊群概念圖

本系統的遊戲人工智慧模組，也能透過戰術的層面，來考量群體性的戰技需求，而將攻擊部隊集合編隊成攻擊部隊群，然後儲存相關部隊

的資料，以達到將所屬的部隊群指派特定戰技的效果，得以有效管理多個部隊，免除逐一下達指令至個別部隊的瑣碎情況。

三、本系統戰術概念介紹:

戰術的實施規則，將會在玩家選定特定的戰術後進行切換。以下介紹本遊戲系統中人工智慧模組所具備的戰術概念；如果需要更為具體地瞭解遊戲系統內的實際參數設定，可以參考本論文附錄。

(一)、多線打擊戰術

這是隸屬於進攻戰略之下的戰術，本戰術是指如果敵方基地超過兩個，並且我方戰鬥部隊數量充足的情況下，將會把攻擊部隊平分成兩群，分別朝向敵方的兩個基地進行攻擊。這樣的戰術可以有機會進攻防禦分布較少的敵人基地，藉此達到打擊敵人基地的效果。

(二)、正面攻擊戰術

這是隸屬於進攻戰略之下的戰術，本戰術是指玩家將會將工作部隊控制在最小的生產數量範圍內，並將多餘的資源用於生產攻擊部隊，並且將會指派所有的攻擊部隊，進攻至敵人的主要基地，目的是在攻擊部隊數量優勢的情況下，直接採取武力擊敗對手。

(三)、擴張最大化戰術

這是隸屬於經濟戰略之下的戰術，本戰術是指玩家將所有的資源，皆用於建造工作部隊與發展新的資源點基地，並在短時間內將經濟國力強化至最高；在經濟情況最大化後，則開始生產攻擊部隊進行防守，目的在敵人尚未進攻前，增強本身的經濟國力。

(四)、擴張防守戰術

這是隸屬於經濟戰略之下的戰術，本戰術是指玩家將會在我方攻擊部隊維持在能夠防守敵方攻擊部隊的最低限度數量，發展經濟國力，即

致力於建造工作部隊與發展新的資源點基地。經濟國力發展速度雖會不及擴張最大化戰術，但是在對付敵人較少的攻擊部隊進攻時，也具有相當的防守力量。

(五)、防守最大化戰術

這是隸屬於防守戰略之下的戰術，本戰術是指在經濟國力最低標準的情況下，全力生產攻擊部隊，並且將攻擊部隊放置在基地巡邏防守。雖然放棄了擴張資源點基地，但是若能成功防禦敵人大量攻擊部隊進攻之後，將會獲得攻擊部隊數量的優勢，以進行反擊。

(六)、少量部隊騷擾戰術

這是隸屬於防守戰略之下的戰術，本戰術是指透過分配少量的攻擊部隊，進攻敵人的基地以打擊對手，並且將其他攻擊部隊，分配在我方基地，以對於敵人的攻擊進行防守。在防守成功的情況下，比防守最大化戰術增加了進攻的概念，預期透過少量攻擊部隊的進攻，可更快地降低敵人的經濟國力。

四、玩家分數評分:

在知名的即時戰略遊戲「世紀帝國 2」中，具有輸出遊戲中的玩家分數的功能，玩家分數在「世紀帝國 2」中的評分方式包含探勘地形區域、收集資源、擁有的部隊等項目。「世紀帝國 2」中的評分玩家功能是即時運作的，玩家可以透過快捷鍵指令，得知雙方玩家目前的分數；此功能可以讓玩家評斷自身與對手玩家的遊戲能力強度或是遊戲局勢。本研究認為透過評分玩家，可以作為遊戲人工智慧判斷目前戰局情勢的參考，而本系統也實際將評分玩家功能，拆解成多種玩家遊戲狀態的評估表示，將在第四章第二節的資料儲存格式，進行詳細的探討。



圖 3-8 「世紀帝國 2」遊戲中的玩家分數示意圖

第四節 詢問式 SVM 演算法

本研究嘗試利用林智仁(Lin Chih-Jen)教授所開發的 Libsvm 程式[19]，來將即時戰略遊戲的實際對戰資料，訓練出進攻戰略的邊界模型；再將模型中的支持向量透過投影的方式，找到更接近分類邊界的投影點座標，經由詢問遊戲專家，人工調整這些新訓練點的類別，再重覆訓練，直到人工判斷學習終止或學習效果變差為止。最後將獲得的分類模型，透過本研究開發的程式，將邊界公式輸出，即可做為遊戲人工智慧新的進攻戰略選用依據；再與使用舊有進攻戰略的遊戲人工智慧進行對戰，以比較出機器學習後，進攻戰略選用時機的改進幅度，並進行相關討論。

有關詢問式 SVM 演算法的執行步驟，計有：

一、步驟一：資料蒐集

本研究設計一款符合市面上即時戰略遊戲基本運行概念之遊戲軟體，並且透過加入簡單的專家規則，設計了初步的遊戲人工智慧，以供電腦實際進行多場對戰；研究者則透過過濾對戰過程中，雙方玩家採用

進攻戰略與否的遊戲狀態資料，以形成 SVM 分類器的訓練資料集與測試資料集兩種。

二、步驟二:SVM 訓練

將訓練集輸入至 libsvm 套件中的 svm-train 程式中，來進行支持向量模型的訓練，過程並嘗試優化分類器的相關參數(可利用 libsvm 套件中的 grid.py)，得到最佳的訓練集分類效果，以做為初版 SVM 模型，並輸出對應的邊界模型。

三、步驟三:計算投影點

將邊界模型與 SVM 模型中的支持向量 SV，利用第二章第六節內所對應之詢問式 SVM 投影公式，逐一產生對應 SV 的投影點，以做為新的訓練點資料，並將其預設成與對應支持向量同一類別。關於投影公式的選用，需依照本次所使用的 SVM 核函數，分別參考

<1> 線性投影點 (適用公式 2-5)或

<2> 非線性投影點公式，則需計算包含滿足假設一條件(適用公式 2-6~2-8)、假設二條件(適用公式 2-7~2-9)、線性(適用公式 2-5)共三種不同狀態的投影，再帶回步驟二的邊界模型裏計算距離，保留距離最小者，做為該 SV 的對應投影點。

四、步驟四:專家諮詢暨調整訓練資料

針對各投影點的預設類別，逐一交由專家進行人工確認或修改。同時為獲得到較佳的 SVM 訓練效果，也需檢查最後投影點的兩種類別是否平衡；必要時則隨機選取 SV 的方式，來補充點數較少的類別，合而成為新的訓練資料集。若被專家修改投影點類別的數量超過一半，則進入步驟五，否則直接進入步驟六。

五、步驟五: 隨機選取原始訓練資料進行補充

若投影點預設類別修改數量超過半數，通常意謂著投影點開始集中於邊界的特定一側，往往代表變動較大或是訓練資料集的兩種類別分佈較為懸殊。遇到此現象，可能即使選用所有 SV 來補充較少類別，還是會不足，故可隨機選取原始的訓練資料來進行補充。

六、步驟六:SVM 訓練暨測試

如同步驟二，將所蒐集的訓練資料集輸入至 Libsvm 程式，來進行支持向量模型的訓練，過程並嘗試優化分類器的相關參數，以獲得新的分類模型。再利用 Libsvm 中的 svm-predict 程式對於測試集進行預測，若預測的準確率確實較前次分類模型有所提昇，則進入步驟八。否則，進入下列步驟七。

七、步驟七:專家細部調整訓練資料

要求專家再次確認投影點的類別，與更換補充的 SV，以進行細部調整；若被專家修改投影點類別的數量超過一半，則回到步驟五，否則直接進入步驟六。

八、步驟八:實戰測試成效

將 SVM 的模型轉換成新遊戲人工智慧的戰術判斷規則後，再與使用舊有專家規則戰術的遊戲人工智慧進行 1000 場的對戰。如果新遊戲人工智慧的戰勝機率較高，則可回到步驟三，嘗試再次進行投影訓練，以求得更佳的模型；否則學習中止，確認機器學習後戰術策略的改進幅度，並進行相關討論。

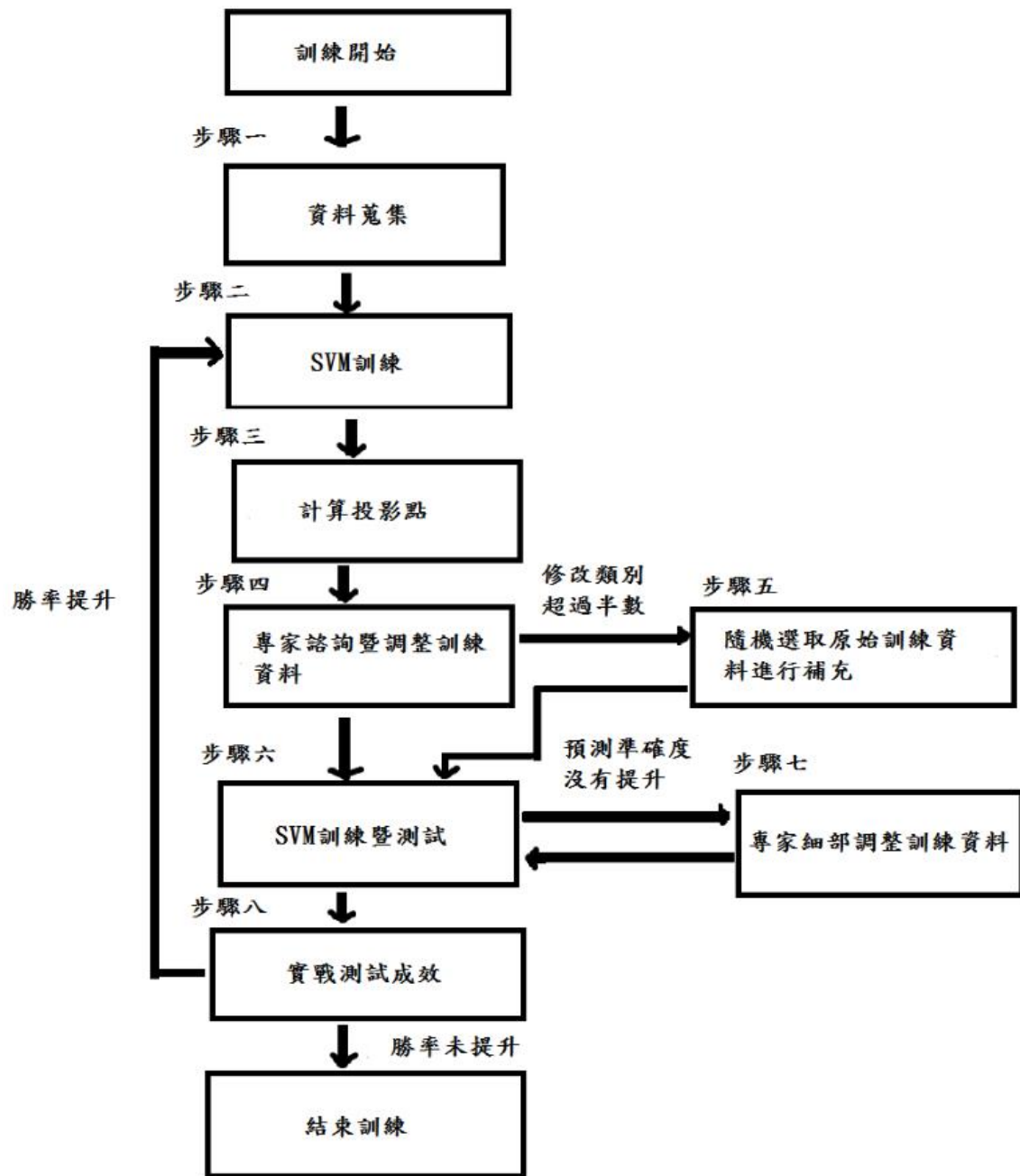


圖 3-9 詢問式 SVM 訓練流程圖

第四章 實驗方法及結果

本章主要說明了遊戲系統人工智慧模組中的專家規則設計理念，及運作方式；也具體說明了透過一些前測實驗，來決定如何從即時戰略遊戲對戰資料裡，擷取適當的特徵指標，來做為後續訓練資料的儲存格式。此外，更詳細說明了如何利用 Libsvm 程式，來將儲存遊戲資料，透過投影及專家詢問方式，訓練出適合的分類模型，再輸出做為遊戲人工智慧新的進攻戰略選用依據。最後透過新舊進攻戰略的遊戲人工智慧進行對戰，來具體瞭解進攻戰略選用時機的改進幅度，並進行相關討論。

第一節 即時戰略遊戲之專家規則設計

本研究建置了一個簡單的即時戰略對戰遊戲，並且參考第三章第三節的戰術內容，設計了一套基本的人工智慧專家規則，來規範電腦人工智慧玩家的行為，以符合戰略遊戲中三種主要戰略的基本涵義。為了區別透過專家規則選擇戰術、與採用完全隨機選擇戰術的不同效果，本研究採取直接進行 1000 場對戰的方式，來驗證本套專家規則的效果；最後專家規則相較於隨機戰術的電腦，勝率達 67%，由此可窺見這些專家規則的有效性。而本研究也將透過這套專家規則的運作，得以進一步地蒐集遊戲資料、訓練及調整遊戲的人工智慧。

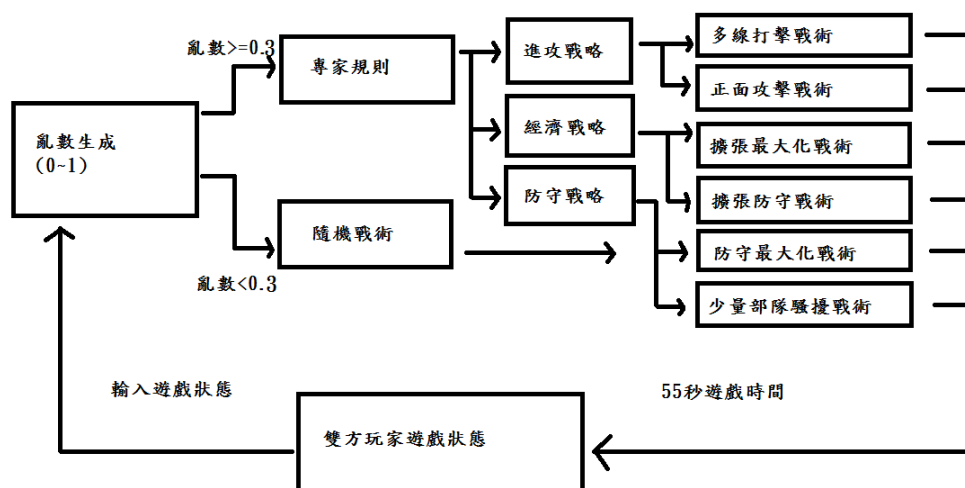


圖 4-1 遊戲人工智慧的決策模組

如圖 4-1 所示，本研究透過輸入雙方玩家遊戲狀態至遊戲人工智慧模組後，為了提昇遊戲進行時的多樣性，系統首先會自動產生一個介於 0 到 1 之間的亂數，以篩選是否啟用專家規則的運作。本研究設定戰術的判斷為 70% 的機率是透過專家規則模組、及 30% 的機率是隨機選擇；因此，亂數大於等於 0.3 時會進入專家規則模組，亂數小於 0.3 時則會隨機選擇戰術。在透過專家模組選擇戰略後，其下皆有兩個達成戰略目標的戰術可以選擇執行，本系統是根據雙方軍事狀態評估、雙方經濟狀態評估、雙方玩家分數評估、雙方個別基地部隊數量、攻擊部隊數量來調整包括基地、工作部隊、採集資源、現有的攻擊部隊、基地巡邏防守及生產攻擊部隊等狀況，而戰術判斷所依據的專家規則詳列於附錄。

第二節 儲存資料設計

一、選擇進攻戰略時機的重要性

在即時戰略遊戲中，評量何時可以進行攻擊，一直是即時戰略遊戲人工智慧與職業玩家相當關心的一個議題。例如由 AIIDE 所舉辦的星海爭霸 AI 設計比賽中，人工智慧 Tcsmoo 的開發者訪問[12]中得知，攻擊部隊的組成與應用是非常重要的議題；因為相較於較為被動的防守戰略與擴張戰略，進攻戰略更能成為一場遊戲進行的勝負關鍵。礙於本研究之實驗進行時間與資源考量，於是將進攻戰略作為本研究的主要探究對象，並且希望透過 SVM 的訓練調整後，得以提昇電腦人工智慧玩家，在選擇進攻戰略時機的準確度。

二、篩選訓練資料的紀錄分數門檻

本研究為探索即時戰略遊戲，資料紀錄的方式與傳統的回合制遊戲不同的是，無法透過回合數來區分遊戲步驟，而是必須要在有限的時間內，透過現有的資訊進行決策；於是本研究在儲存訓練資料上的設計，主要考量為透過時間間隔，來進行遊戲雙方各種狀態的評估。本研究經過各種嘗試後，發現將對戰遊戲資料記錄的時間間隔，設定為 55 秒較為適合；然而，適合的時間間隔設定會依照不同遊戲而改變，55 秒的設定顯然並不是各種即時戰略遊戲都通用的數值。

篩選紀錄的資料方面，本研究透過時間間隔的前後狀態比較作為根據，設計的概念則主要參考知名即時戰略遊戲「世紀帝國 2」，透過玩家擁有的部隊、建築物、科技的數量來表達玩家現在的國力狀態，之後透過前一個時間點的我方國力狀態與對手的國力狀態差距，再和之後時間點的雙方國力差距的改變，來評估這個戰術決策的效果；如果讓能夠國

力差距的數值從上一個狀態提昇，則將此次的戰術選擇以訓練範例的方式紀錄。

有關雙方國力狀態的評估，是依照下列方式來進行計算：

我方國力分數 = 我方基地部隊數 * 25 + 我方工作部隊數 * 5 + 我方攻擊部隊數 * 7

敵方國力分數 = 敵方基地部隊數 * 25 + 敵方工作部隊數 * 5 + 敵方攻擊部隊數 * 7

雙方國力狀態差距 = 我方國力分數 - 敵方國力分數

雙方國力狀態評估 = 現在雙方國力狀態差距 - 前一次雙方國力狀態差距

(公式 4-1)

本研究將各種部隊所代表的國力分數分別設定為：基地部隊 25 分、工作部隊 5 分、和攻擊部隊 7 分，而國力狀態差距為我方軍事國力分數減去敵方軍事國力分數；至於雙方軍事國力狀態評估，則使用現在的雙方軍事國力差距減去前一個時間點(本系統設定為 55 秒前)的雙方軍事國力差距。如果國力分數較前一個時間點提高了 20 分以上，則記錄為正向訓練資料；反之，如果國力分數較前一個時間點下降 20 分以上，則記錄為反向訓練資料。如此選擇的原因說明如下：

表 4- 1 採用不同紀錄分數門檻來篩選訓練資料之前測對戰勝率

結果 項目	遊戲場數	獲勝場數	獲勝機率
10 分紀錄門檻	1000	467	46.7%
15 分紀錄門檻	1000	543	54.3%
20 分紀錄門檻	1000	548	54.8%
25 分紀錄門檻	1000	545	54.5%

本研究利用一般線性 SVM 模型，嘗試學習對象分別為採用不同國力狀態評估差異，來作為紀錄分數門檻，以篩選出不同的訓練資料集。整個訓練階段結束後，再透過與使用基本的專家規則的遊戲模組對戰，發現其中採用 20 分門檻所產生的線性 SVM 模型表現最好，而以使用 10 分門檻的模型表現較差。從表 4-1 不難觀察到，遊戲資料的篩選門檻過低或太高，則蒐集到的資料越難分類，容易導致電腦人工智慧判斷失誤的情況越多，最後會造成獲勝場數比例較低。所以，本研究後續皆採用國力狀態評估差異 20 分，來做為遊戲對戰資料的記錄門檻，以產生 SVM 的訓練資料集。

三、資料格式設計

為了順利利用 Libsvm 套件中 SVM 模型訓練的功能，使用者必須參考其提供的 SVM 資料表格模式來編輯訓練資料，方能透過 SVM-train 的功能進行模型訓練。SVM 資料格式如 4-2 圖，其中 label 欄位為該筆資料的分類種類，index 為訓練資料的特徵參數編號，value 為該特徵參數編號所對應的數值。

```
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...  
[label] [index1]:[value1] [index2]:[value2] ...
```

圖 4-2 Libsvm 資料格式範例

本研究將蒐集的遊戲對戰資料之特徵參數定義如表 4-2 所示。透過雙方進行對戰過程，將所選用的戰略編號先行紀錄；若在下一個時間點

(本研究設定為 55 秒)後，評估此次的戰略選擇可做為訓練範例(國力狀態評估差異為 20 分)，則將這次的紀錄儲存在訓練資料內，依序紀錄了戰略編號，雙方軍事國力狀態評估，雙方經濟國力狀態評估，以及雙方玩家分數評估，與雙方各自的攻擊部隊數量、與雙方各自的基地部隊數量，後續將詳細說明各項特徵指標的計算方式。

表 4-2 本研究採用之遊戲訓練資料的特徵說明一覽表

項目	說明
Label	戰略效果分類
Index1	雙方軍事國力狀態評估
Index2	雙方經濟國力狀態評估
Index3	雙方玩家分數評估
Index4	我方攻擊部隊數量
Index5	敵方攻擊部隊數量
Index6	我方基地部隊數量
Index7	敵方基地部隊數量

(一)、Label 戰略效果分類:

Label 記錄為 SVM 訓練資料格式中的戰略效果分類，由於本研究目前鎖定為僅探討進攻戰略的選擇時機，故以類別'1'用來標示選擇進攻戰略後，國力狀態評估上升 20 分的紀錄(訓練資料)；反之，以類別'-1' 用來標示選擇攻戰略後，國力狀態評估下降 20 分的紀錄。

(二)、 Index1 雙方軍事國力狀態評估:

雙方軍事國力狀態評估為 SVM 訓練資料中的第一個特徵指標屬性，其內容主要表達出遊戲當時雙方國家在戰鬥部隊數量上的強弱狀態。因

為攻擊部隊的數量差異在遊戲部隊規模不同的情況下，所代表的重要性並不盡相同；因此，本系統在定義雙方攻擊部隊數量皆小於為 5 個以下時，使用以下方法來決定雙方軍事國力狀態評估：

當雙方攻擊部隊數量差距 ≥ 2 則差距設為 2 雙方軍事國力狀態評估

$$= 0.5 + 0.1 \times (\text{我方攻擊部隊數量} - \text{敵方攻擊部隊數量})$$

如果雙方攻擊部隊數量皆超過 5 個以上，則使用以下方法來決定雙方軍事國力狀態評估：

$$\text{我方攻擊部隊數量} \div (\text{我方攻擊部隊數量} + \text{敵方攻擊部隊數量})$$

(公式 4-2)

這樣的評估方式，可以避免因為攻擊部隊數量太少，導致雙方軍事國力狀態評估的變動幅度過大的問題，故可讓系統運作更加穩定。

(三)、Index2 雙方經濟國力狀態評估：

雙方經濟國力狀態評估為 SVM 訓練資料中的第二個特徵指標屬性，其內容主要表達出遊戲當時雙方國家在工作部隊與基地數量上的強弱狀態。由於工作部隊數量與基地的數量，在初始狀態下就具備較多，因此本系統採用以下方式，來定義雙方經濟國力狀態評估：

$$(\text{我方工作部隊數量} + \text{我方基地部隊數量} \times 4) \div (\text{我方工作部隊數量} + \text{我方基地部隊數量} \times 4 + \text{敵方工作部隊數量} + \text{敵方基地部隊數量} \times 4)$$

(公式 4-3)

(四)、Index3 雙方玩家分數評估：

雙方玩家分數狀態評估為 SVM 訓練資料中的第三個特徵指標屬性，其內容主要表達出遊戲當時雙方國家所有部隊總和，再轉換為分數權重

來表示成強弱關係。本研究將國力分數狀態設定為每個基地部隊 25 分、工作部隊 5 分、和攻擊部隊 7 分，將這些權重值乘上該類部隊數量，則可以計算出該玩家的分數，並且透過以下方式來定義雙方玩家分數評估：

我方國力分數 ÷ (我方國力分數 + 敵方國力分數) (公式 4-4)

(五)、Index4 我方攻擊部隊數量：

我方攻擊部隊數量為 SVM 訓練資料中的第四個特徵指標屬性，該指標直接紀錄遊戲當時我方攻擊部隊的實際數量；這是由於本遊戲中基地具有一定程度的反擊能力，所以攻擊部隊必需要超過一定的數量，才有辦法有效打擊到敵方的基地部隊，因此有必要將我方攻擊部隊數量加入訓練資料中。但由於部隊數量過大的數值，可能會影響到 SVM 訓練時的效能；於是紀錄時是將我方攻擊部隊的實際數量，再除以 100，使其成為實數。

(六)、Index5 敵方攻擊部隊數量：

敵方攻擊部隊數量為 SVM 訓練資料中的第五個特徵指標屬性，該屬性直接紀錄遊戲當時敵方攻擊部隊的實際數量。相對於進攻方因為基地的反擊能力，需要具備一定數量的攻擊部隊方能有效進攻；我方基地的防守佈署決策，也需要透過敵方攻擊部隊的數量來進行判斷，因此有必要將敵方攻擊部隊數量加入訓練資料中。但由於部隊數量過大的數值，可能會影響到 SVM 訓練時的效能；於是紀錄時是將敵方攻擊部隊的實際數量，再除以 100，使其成為實數。

(七)、Index6 我方基地部隊數量：

我方基地部隊數量為 SVM 訓練資料中的第六個特徵指標屬性，該屬性紀錄遊戲當時我方基地部隊的實際數量。這裡對於我方基地部隊的數

量，將一、二、三個基地分別紀錄為 0.33、0.66、1 的方式來表達，這是因為經由實驗發現我方發展度的屬性跟進攻戰略的選擇較沒有直接關係，反倒是基地部隊的數量，可以較為直接地影響到進攻戰略的選擇；可能的原因為基地部隊是一個花費資源龐大的部隊，而在即時戰略遊戲中，一個國家在剛建立好額外據點的時候，因為沒有辦法快速佈署防守的部隊，以及因為建造基地部隊會花費龐大的資源量，當下國家趨於脆弱，而容易導致被敵人攻擊而受到重創。因此，接下來可能選擇採取防守型戰略會比較合適，所以需要此指標作為訓練資料的一部份。

(八)、Index7 敵方基地部隊數量:

敵方基地部隊數量為 SVM 訓練資料中的第七個特徵指標屬性，該屬性紀錄遊戲當時敵方的基地部隊數量。這裡對於敵方基地部隊的數量，將一、二、三個基地分別紀錄為 0.33、0.66、1 的方式來表達。如同 Index6 我方基地部隊數量的屬性說明，觀察到敵方的基地部隊數量增加，可視為該國家較為脆弱的時候；並且因為國家領土變大，可以被進攻的區域也隨著成長。因此，在雙方攻擊部隊數量差距不多的情況下。進攻方會較為有利。因此，此指標亦可以成為進攻戰略選擇的依據之一。

四、採用不同訓練資料格式的實驗

前面介紹了本研究最終採用的訓練資料儲存格式，其原因是採用該格式的訓練資料集利用 SVM 線性模型訓練之後，在進行模型預測時能顯現出較好的準確率。以下將會介紹本論文嘗試使用過的四種不同訓練資料格式，以及進行比較。

(一)、訓練資料格式 A:

表 4-3 訓練資料格式 A

項目	說明
Label	戰略效果分類
Index1	雙方軍事國力狀態評估
Index2	雙方經濟國力狀態評估
Index3	雙方玩家分數評估

格式 A 僅包含 3 種狀態評估指標，主要是測試本遊戲的設計，是否簡單到僅需使用這三種屬性，就能進行戰略決策。事實上格式 A 由於特徵指標屬性的數量過少，導致對於測試集之預測準確率僅達 62.3%，較為不佳。

(二)、訓練資料格式 B:

表 4-4 訓練資料格式 B

項目	說明
Label	戰略效果分類
Index1	雙方軍事國力狀態評估
Index2	雙方經濟國力狀態評估
Index3	雙方玩家分數評估
Index4	我方攻擊部隊數量
Index5	敵方攻擊部隊數量
Index6	我方基地部隊數量
Index7	敵方基地部隊數量

除了格式 A 採用的三種基礎狀態評估指標外，還增加了雙方的攻擊部隊數量與基地部隊數量，在之後的測試中，也顯示出訓練資料格式 B 對於測試集之預測準確率可達 73.6%，為最高者，故選為後續本遊戲系統的訓練資料儲存格式。

(三)、訓練資料格式 C:

表 4-5 訓練資料格式 C

項目	說明
Label	戰略效果分類
Index1	雙方軍事國力狀態評估
Index2	雙方經濟國力狀態評估
Index3	雙方玩家分數評估
Index4	我方攻擊部隊數量
Index5	敵方攻擊部隊數量
Index6	我方經濟發展度
Index7	敵方經濟發展度

繼承訓練資料格式 B 的各項屬性，並且將最後兩個特徵指標屬性：雙方基地數量，以雙方國力發展度進行取代。而國力發展度的定義為：該玩家基地數量*0.2+該玩家工作部隊數量*0.019。這樣的表示方法，可以在遊戲的經濟飽和型態，亦即達 3 個基地及 21 個工作部隊的情況下(本遊戲所能達到的經濟國力最大化情況)，讓屬性數值趨近於 1，可以更完整顯示該玩家的圓滿經濟發展狀態。但是之後的測試資料顯示準確率僅達 70%，所以這樣的表示方法，並不完全適合做為進攻戰略的決策參考指標，故不採用此格式。

(四)、訓練資料格式 D:

表 4- 6 訓練資料格式 D

項目	說明
Label	戰略效果分類
Index1	雙方軍事國力狀態評估
Index2	雙方經濟國力狀態評估
Index3	雙方玩家分數評估
Index4	我方攻擊部隊數量
Index5	敵方攻擊部隊數量
Index6	我方經濟發展度
Index7	敵方經濟發展度
Index8	我方基地部隊數量
Index9	敵方基地部隊數量
Index10	我方工作部隊數量
Index11	敵方工作部隊數量

格式 D 幾乎完整地包含雙方所有的玩家遊戲狀態，但是透過測試資料顯示準確率僅達 71.6%，並非所有四種格式中最高者。經過分析，可能是因為基地部隊數量與工作部隊數量衝突所導致，主要原因在於基地數量與工作部隊數量不一定具有直接關係，例如：在擴張最大化戰術之專家規則下，有關基地的發展判斷為，只要玩家資源大於 1200 且玩家基地數量不大於等於 3，則會立即生產基地部隊；在這種情況下對方沒有生產基地部隊，而是生產工作部隊，就會造成當時我方基地部隊雖然較多，但是工作部隊相對會較少的情況。另外，經濟發展度在訓練資料格

式 C 的說明中也指出，並不適合成為進攻戰略的判斷依據指標。表 4-7 是整理了本系統在選用訓練資料儲存格式過程的實驗數據。

表 4-7 各種訓練資料格式前測比較表

項目 格式	訓練資料數	模型檔案 SV 數	測試資料驗證
訓練資料格式 A	987	736	62.3%
*訓練資料格式 B	987	652	73.6%
訓練資料格式 C	987	721	70%
訓練資料格式 D	987	706	71.6%

經由測試資料驗證效能，本研究將採用訓練資料格式 B 來進行後續實驗。

第三節 實驗結果

一、最佳化參數工具

林智仁(Lin Chih-Jen)教授所開發的 Libsvm 軟體套件中，提供了 SVM 模型訓練時，所需的參數最佳化工具 grid.py，來進行不同參數組合的格子理論搜尋工作。在本研究中，主要是利用 grid.py 來將 SVM 模型懲罰參數 Cost 與核函數集中參數 γ ，以指數倍增的方式逐步增加，並且將每一種參數組合都進行 SVM 訓練，最後顯示出分類率效果最好的一組參數，故此工具可以節省許多手動進行參數調整所需要的成本。

例如本研究在進行第二版線性模型訓練時，使用了預設參數所獲得的訓練後模型，訓練準確度僅為 77.667%；而使用了 grid.py 工具來進行參數最佳化，可找到最佳參數 C 為 0.128 及 γ 為 0.0078125，使用後針對

同一個訓練資料集，所獲得的訓練後模型，訓練準確度則提昇為 83.667%。
由此可見，參數最佳化工具對於本研究進行訓練時確有助益。

```
[local] -5 -3 50.2439 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 7 -3 99.0244 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 1 -3 99.0244 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -7 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -1 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -13 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 1 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -11 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -5 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -15 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 3 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -9 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
[local] 13 -3 98.7805 <best c=0.125, g=0.0078125, rate=99.0244>
0.125 0.0078125 99.0244
```

圖 4-3 參數最佳化工具 grid.py 的使用狀況

二、線性模型實驗

(一)、初版線性模型:

本遊戲系統擁有初始專家知識的人工智慧，其與完全隨機選擇遊戲策略的對遊戲人工智慧戰，已可達到 67% 的勝率，故足以作為本研究的基礎專家知識。實驗以 70% 機率使用專家規則及 30% 機率隨機的方式進行多次對戰，最後收集了 700 筆紀錄，其中挑選進攻戰略正例 200 筆及進攻戰略反例 200 筆的資料做為訓練集，再將挑選進攻戰略正例 150 筆及進攻戰略反例 150 筆的資料做為測試集。根據 3-3 節的線性模型訓練作法，先透過標準 SVM 訓練，獲得了具有 209 個支持向量 SV(Support Vector) 的初版線性模型，如 4-4 圖所示:

```

1 svm_type c_svc
2 kernel_type linear
3 nr_class 2
4 total_sv 209
5 rho 5.18093
6 label 1 -1
7 nr_sv 105 104
8 SV
9 1 1:0.7 2:0.48 3:0.55 4:0.05 5:0.02 6:0.333 7:0.333
10 1 1:0.6 2:0.39 3:0.44 4:0.02 5:0 6:0.333 7:0.666
11 1 1:0.6 2:0.38 3:0.42 4:0.03 5:0.02 6:0.333 7:0.666
12 1 1:0.6 2:0.43 3:0.47 4:0.02 5:0 6:0.666 7:0.666
13 1 1:0.6 2:0.56 3:0.58 4:0.03 5:0.01 6:0.666 7:0.666
14 1 1:0.5 2:0.62 3:0.58 4:0.02 5:0.02 6:0.333 7:0.333
15 1 1:0.7 2:0.36 3:0.44 4:0.03 5:0 6:0.333 7:0.666
16 0.8656911962154178 1:0.7 2:0.38 3:0.46 4:0.04 5:0.01 6:0.333 7:0.666
17 1 1:0.5 2:0.62 3:0.58 4:0.02 5:0.02 6:0.333 7:0.333
18 1 1:0.7 2:0.36 3:0.44 4:0.03 5:0 6:0.333 7:0.666

```

圖 4-4 SVM 訓練獲得之初版線性模型內容

透過本研究建置之分界線運算程式，可得出此線性模型之分界線如公式 4-5，其中 x_1, \dots, x_7 分別對應 4-2 節訓練資料格式 B 中的 $index_1, \dots, index_7$ 。

$$\begin{aligned}
 f(x) = & 5.00888849399165 * x_1 + 0.578620328965528 * x_2 + \\
 & 2.22847584811801 * x_3 + 0.367397272480561 * x_4 - \\
 & 0.626986134566677 * x_5 - 1.66500000657141 * x_6 + \\
 & 2.15631724417095 * x_7 - 5.18093
 \end{aligned}
 \quad (\text{公式 4-5})$$

使用 `libsvm-predict` 工具對此線性模型進行測試集預測，可得知初版線性模型的預測準確率為 80.333%。將此分界線公式取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 54.1%；故可稱此傳統 SVM 線性分界線模型，對於原有的專家規則對戰效果，可以提昇約 4.1% 的勝率。

(二)、加入第一次投影點的第一版線性模型:

將初版線性模型內的支持向量 SV(Support Vector)，輸入至本研究建置之線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 209 個新

的投影點，且透過專家檢查調整了其中 32 個投影點的類別。由於被修改的投影點類別並沒有超過投影點總數的一半，且投影點正例的數量較多，為了平衡正反兩類的資料筆數，本系統透過隨機取樣的方式，從初版線性模型 SV 中選取了 37 個反例，再加上計算所得的第一次投影點，最後形成了數量為 246(=209+37)筆資料的訓練集，並進行了 SVM 線性模型訓練，結果產生了支持向量數目為 206 個的第一版線性模型，如圖 4-5 所示：

```

1 svm_type c_svc
2 kernel_type linear
3 nr_class 2
4 total_sv 206
5 rho 3.8335
6 label 1 -1
7 nr_sv 104 102
8 SV
9 1 1:0.61 2:0.39 3:0.44 4:0.02 5:0 6:0.33 7:0.67
10 1 1:0.62 2:0.38 3:0.43 4:0.03 5:0.02 6:0.33 7:0.67
11 1 1:0.67 2:0.44 3:0.5 4:0.03 5:-0.01 6:0.64 7:0.7
12 1 1:0.63 2:0.56 3:0.59 4:0.03 5:0.01 6:0.66 7:0.68
13 1 1:0.61 2:0.63 3:0.63 4:0.03 5:0.01 6:0.3 7:0.38
14 1 1:0.65 2:0.35 3:0.42 4:0.03 5:0.01 6:0.35 7:0.64
15 1 1:0.64 2:0.37 3:0.43 4:0.04 5:0.02 6:0.35 7:0.64
16 1 1:0.61 2:0.63 3:0.63 4:0.03 5:0.01 6:0.3 7:0.38
17 1 1:0.65 2:0.35 3:0.42 4:0.03 5:0.01 6:0.35 7:0.64
18 1 1:0.64 2:0.5 3:0.57 4:0.02 5:-0.02 6:0.29 7:0.39
19 1 1:0.61 2:0.39 3:0.44 4:0.02 5:0 6:0.33 7:0.67
20 1 1:0.61 2:0.39 3:0.44 4:0.02 5:0 6:0.33 7:0.67
21 0.125418182207303 1:0.69 2:0.47 3:0.58 4:0.04 5:0 6:0.34 7:0.33

```

圖 4-5 第一版投影點經 SVM 訓練所得之線性模型

透過本研究建置之分界線運算程式，可得出此線性模型之分界線如公式 4-6:

$$\begin{aligned}
 f(x) = & 5.24934212628949 * x_1 + 2.2945797399513 * x_2 + \\
 & 2.84320407185266 * x_3 + 0.207981561923324 * x_4 - \\
 & 0.819207944068135 * x_5 - 2.5292221366679 * x_6 + \\
 & 1.75505308314985 * x_7 - 3.8335
 \end{aligned}
 \tag{公式 4-6}$$

使用 `libsvm-predict` 工具對此線性模型進行測試集預測，可得知第一版線性模型的預測準確率為 82.333%，預測準確度較初版線性模型的準確度提升了 2%，因此直接進行對戰驗證。

將此分界線公式取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 56.3%；故可稱此第一版投影線性分界線模型，對於原有的專家規則對戰效果，可以提昇約 6.3% 的勝率。

(三)、加入第二次投影點的第二版線性模型：

將第一版線性模型內的支持向量 SV，輸入至本研究建置之線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 206 個新的投影點，且透過專家檢查調整了超過半數以上(178 個)投影點的類別；究其原因為線性投影開始集中至理想分界線的一邊附近，導致產生的投影點大部份都偏向反例。由於反例數量已經超過原先的第一版線性模型的正例數量，無法透過只使用第一版線性模型中的 SV 補充訓練資料，為了平衡正反兩類的資料筆數，本系統透過隨機取樣的方式，從原始訓練集適當選取 232 個正例與 28 個反例，加入第二次投影點的訓練集內，最後產生 466(=206+232+28)筆資料的訓練集。接著進行了 SVM 線性模型訓練，並透過 `grid.py` 工具找到訓練優化參數 `cost` 為 0.125 和 `gamma` 為 0.0078125，結果產生了支持向量數目為 350 個的第二版線性模型，如圖 4-6 所示：

```

1 svm_type c_svc
2 kernel_type linear
3 nr_class 2
4 total_sv 350
5 rho 4.2198
6 label 1 -1
7 nr_sv 174 176
8 SV
9 0.125 1:0.35 2:0.52 3:0.49 4:0.02 5:0.05 6:0.42 7:0.29
10 0.125 1:0.42 2:0.58 3:0.51 4:0.07 5:0.1 6:0.66 7:0.34
11 0.125 1:0.43 2:0.58 3:0.53 4:0.07 5:0.08 6:0.7 7:0.31
12 0.125 1:0.61 2:0.39 3:0.44 4:0.02 5:0 6:0.33 7:0.67
13 0.125 1:0.62 2:0.38 3:0.43 4:0.03 5:0.02 6:0.33 7:0.67
14 0.125 1:0.67 2:0.44 3:0.5 4:0.03 5:-0.01 6:0.64 7:0.7
15 0.125 1:0.63 2:0.56 3:0.59 4:0.03 5:0.01 6:0.66 7:0.68
16 0.125 1:0.65 2:0.35 3:0.42 4:0.03 5:0.01 6:0.35 7:0.64
17 0.125 1:0.64 2:0.37 3:0.43 4:0.04 5:0.02 6:0.35 7:0.64
18 0.125 1:0.65 2:0.35 3:0.42 4:0.03 5:0.01 6:0.35 7:0.64
19 0.125 1:0.64 2:0.5 3:0.57 4:0.02 5:-0.02 6:0.29 7:0.39

```

圖 4-6 第二版投影點經 SVM 訓練所得之線性模型

透過本研究建置之分界線運算程式，可以得出此線性模型之分界線公式 4-7 為：

$$\begin{aligned}
 f(x) = & 4.76294155523237 * x_1 + 0.843343476441962 * x_2 + \\
 & 2.079210376544484 * x_3 + 0.529025469683165 * x_4 - \\
 & 0.418614873990716 * x_5 - 1.72727132379922 * x_6 + \\
 & 1.54748578244087 * x_7 - 4.2198
 \end{aligned}
 \quad (\text{公式 4-7})$$

使用 libsvm-predict 工具對此線性模型進行測試集預測，可得知第二版線性模型的預測準確率為 83.667%，預測準確度較第一版線性模型的準確度提升了 1.3%，因此直接進行對戰驗證。

將此分界線公式取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 60.8%；故可稱此第二版投影線性分界線模型，對於原有的專家規則對戰效果，可以提昇約 10.8%的勝率，並且提昇的效果相較第一版線性模型的 6.3%的勝率還要高些。

(四)、加入第三次投影點的第三版線性模型：

將第二版線性模型內的支持向量 SV，輸入至本研究建置之線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 350 個新的投影點。

透過專家檢查，發現許多投影點都會對應到即時戰略遊戲中的平均戰局狀況，也就是戰略的選擇絕大部份是依靠雙方玩家進行博弈，因此極難明確定義其類別；在此情況下，專家還是盡可能地調整了其中 185 個投影點的類別，並且由於被修改的投影點類別，超過投影點總數的一半，為了平衡正反兩類的資料筆數，經過多次實驗後，本系統透過隨機取樣的方式，從原始訓練集適當選取 190 個正例與 160 個反例，加入第三次投影點的訓練集內，最後產生 700(=350+190+160)筆資料的訓練集。接著進行了 SVM 線性模型訓練，發覺此時即使透過 grid.py 工具，也找不到其他的訓練優化參數；甚至從原始訓練資料集中嘗試透過多次隨機取樣，來改變訓練資料集，發覺效能也無法提昇，最後結果產生了支持向量數目為 571 個的第三版線性模型，如圖 4-7 所示：

```

1 svm_type c_svc
2 kernel_type linear
3 nr_class 2
4 total_sv 571
5 rho 1.94951
6 label 1 -1
7 nr_sv 285 286
8 SV
9 1 1:0.39 2:0.54 3:0.51 4:0.02 5:0.04 6:0.4 7:0.3
10 1 1:0.46 2:0.6 3:0.53 4:0.07 5:0.09 6:0.64 7:0.35
11 1 1:0.46 2:0.32 3:0.36 4:0.01 5:0.02 6:0.4 7:0.62
12 1 1:0.47 2:0.31 3:0.35 4:0.02 5:0.04 6:0.4 7:0.62
13 1 1:0.5 2:0.29 3:0.34 4:0.02 5:0.03 6:0.42 7:0.59
14 1 1:0.49 2:0.31 3:0.35 4:0.03 5:0.04 6:0.42 7:0.59
15 1 1:0.5 2:0.29 3:0.34 4:0.02 5:0.03 6:0.42 7:0.59
16 1 1:0.45 2:0.42 3:0.47 4:0.01 5:0.01 6:0.38 7:0.33
17 1 1:0.46 2:0.32 3:0.36 4:0.01 5:0.02 6:0.4 7:0.62
18 1 1:0.46 2:0.32 3:0.36 4:0.01 5:0.02 6:0.4 7:0.62
19 1 1:0.5 2:0.39 3:0.48 4:0.03 5:0.03 6:0.43 7:0.27
20 1 1:0.46 2:0.39 3:0.46 4:0.01 5:0.01 6:0.37 7:0.34
21 1 1:0.51 2:0.41 3:0.46 4:0.01 5:0.01 6:0.72 7:0.64
22 1 1:0.44 2:0.34 3:0.38 4:0.03 5:0.04 6:0.41 7:0.62

```

圖 4-7 第三版投影點經 SVM 訓練所得之線性模型

透過本研究建置之分界線運算程式，可以得出此線性模型之分界線公式 4-8 為：

$$f(x) = 4.48284713713968 * x_1 + 1.84926539686255 * x_2 + 2.36051556751921 * x_3 + 0.259941108190839 * x_4 -$$

$$0.683867875297713 * x5 - 2.1004360109805 * x6 + 1.51139240514789 * x7 - 1.94951 \quad (\text{公式 4-8})$$

使用 libsvm-predict 工具對此線性模型進行測試集預測，可得知第三版線性模型的預測準確率為 72.667%，雖然預測準確度不佳，還是嘗試進行對戰驗證。

將此分界線公式取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 55.5%；故可稱此第三版投影線性分界線模型，相比於第二版投影點的線性模型，效果下降了 4.6%。即使之後進行多次嘗試，也未能提昇其分類預測及對戰效果。

(五)、加入第四次投影點的第四版線性模型：

將第三版線性模型內的支持向量 SV，輸入至本研究建置之線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 571 個新的投影點。透過專家檢查，發現許多投影點相較於第三版線性模型所採用的投影點，恢復為較不靠近分界模型的資料點，並且投影點投影前的正例反例互相對調的情況較多；在此情況下，專家盡可能地調整了其中 289 個投影點的類別，並且由於被修改的投影點類別，超過投影點總數的一半，為了平衡正反兩類的資料筆數，經過多次實驗後，本系統透過隨機取樣的方式，從投影模型檔案內適當選取 61 個正例，加入第四次投影點的訓練集內，最後產生 632(=571+61)筆資料的訓練集。接著進行了 SVM 線性模型訓練，並透過 grid.py 工具找到訓練優化參數 cost 為 8 和 gamma 為 0.0078125，結果產生了支持向量數目為 169 個的第四版線性模型，如圖 4-8 所示：

```

1 svm_type c_svc
2 kernel_type linear
3 nr_class 2
4 total_sv 169
5 rho 18.1393
6 label 1 -1
7 nr_sv 86 83
8 SV
9 8 1:0.6 2:0.32 3:0.4 4:0.02 5:0.02 6:0.36 7:0.57
10 8 1:0.6 2:0.32 3:0.4 4:0.02 5:0.02 6:0.36 7:0.57
11 8 1:0.55 2:0.38 3:0.45 4:0.04 5:0.04 6:0.34 7:0.6
12 1.269074366930006 1:0.58 2:0.55 3:0.61 4:0.02 5:0.01 6:0.61 7:0.62
13 8 1:0.54 2:0.4 3:0.43 4:0.01 5:0.02 6:0.31 7:0.62
14 8 1:0.55 2:0.33 3:0.44 4:0.09 5:0.1 6:0.29 7:0.65
15 8 1:0.56 2:0.36 3:0.44 4:0.02 5:0.02 6:0.34 7:0.6
16 8 1:0.55 2:0.38 3:0.45 4:0.04 5:0.04 6:0.34 7:0.6
17 8 1:0.6 2:0.32 3:0.4 4:0.02 5:0.02 6:0.36 7:0.57
18 8 1:0.58 2:0.55 3:0.61 4:0.02 5:0.01 6:0.61 7:0.62
19 8 1:0.54 2:0.4 3:0.43 4:0.01 5:0.02 6:0.31 7:0.62
20 8 1:0.58 2:0.38 3:0.42 4:0.02 5:0 6:0.35 7:0.68
21 8 1:0.58 2:0.38 3:0.42 4:0.02 5:0 6:0.35 7:0.68
22 8 1:0.58 2:0.38 3:0.42 4:0.02 5:0 6:0.35 7:0.68

```

圖 4-8 第四版投影點經 SVM 訓練所得之線性模型

透過本研究建置之分界線運算程式，可以得出此線性模型之分界線公式 4-9 為：

$$\begin{aligned}
 f(x) = & 17.7411200256149 * x_1 + 5.11342249040678 * x_2 + \\
 & 4.45240095007654 * x_3 + 6.39473278896644 * x_4 + \\
 & -2.01073728328785 * x_5 + -4.14235003626877 * x_6 + \\
 & 9.25724507755265 * x_7 - 18.1393
 \end{aligned}
 \quad (\text{公式 4-9})$$

使用 libsvm-predict 工具對此線性模型進行測試集預測，可得知第四版線性模型的預測準確率為 75.333%，雖然預測準確度不佳，還是嘗試進行對戰驗證。

將此分界線公式取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 57.1%；故可稱此第四版投影線性分界線模型，相比於第三版投影點的線性模型，效果上升了 1.6%，但是由於研究限制的時間問題，所以決定終止了進攻戰略的線性分界線學習程序。

(六)、線性模型總結:

經過多次地針對投影點進行線性模型訓練，以充當電腦玩家的進攻戰略選用規則的實驗後，不難發覺以使用兩次投影點進行訓練的效果最好，如圖 4-8 所示，可以達到將遊戲人工智慧的勝率提昇至 60.1%。在即時戰略遊戲中，由於遊戲中具有相當程度的博弈成分，因此透過機器學習的機制，能夠達到提昇 10.1% 的勝率，已足可稱為具有顯著的學習成效[16]。另外觀察圖 4-9 與圖 4-10，也不難發現到實際對戰的勝率和預測準確度的趨勢走向是相符的。



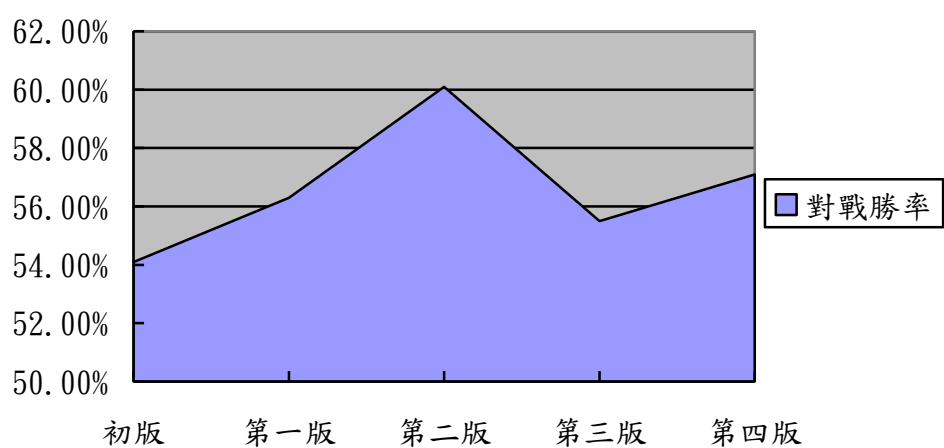


圖 4-9 各版本線性模型的勝率

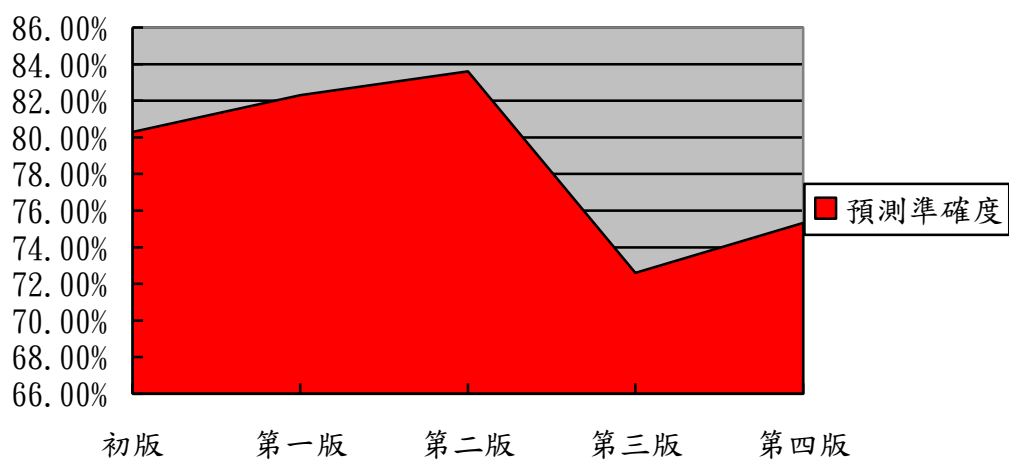


圖 4-9 各版本線性模型的預測準確度

表 4- 8 各版本之線性分界模型訓練資料集

	初版線性 模型	第一版線 性模型	第二版線 性模型	第三版線 性模型	第四版線 性模型
原始訓練集 資料	400	0	260	350	0
投影程式產 生	0	209	206	350	571
上一版本分 界模型支持 向量	0	37	0	0	61
專家修改投 影點類別數	0	32	178	185	289
訓練集總數	400	246	466	700	632
邊界模型支 持向量數	209	206	350	571	169

二、非線性模型實驗

(一)、初版非線性模型:

非線性模型實驗與線性模型實驗使用相同的 400 筆進攻戰略訓練集以及相同的 300 筆測試集資料。根據 3-3 節的非線性模型訓練作法，先透過標準 SVM 訓練，獲得了具有 262 個支持向量 SV 的初版非線性模型，如圖 4-10 所示。根據 2-5 節公式 2-2 及公式 2-4，我們同時也獲得了非線性 SVM 的分割邊界模型。：

```
1 svm_type c_svc
2 kernel_type rbf
3 gamma 0.142857
4 nr_class 2
5 total_sv 262
6 rho -0.145918
7 label 1 -1
8 nr_sv 131 131
9 SV
10 1 1:0.7 2:0.4 3:0.47 4:0.04 5:0.01 6:0.333 7:0.666
11 1 1:0.7 2:0.48 3:0.55 4:0.05 5:0.02 6:0.333 7:0.333
12 1 1:0.6 2:0.39 3:0.44 4:0.02 5:0 6:0.333 7:0.666
13 1 1:0.6 2:0.38 3:0.42 4:0.03 5:0.02 6:0.333 7:0.666
14 1 1:0.6 2:0.43 3:0.47 4:0.02 5:0 6:0.666 7:0.666
15 1 1:0.6 2:0.56 3:0.58 4:0.03 5:0.01 6:0.666 7:0.666
16 1 1:0.5 2:0.62 3:0.58 4:0.02 5:0.02 6:0.333 7:0.333
17 1 1:0.7 2:0.38 3:0.48 4:0.04 5:0 6:0.333 7:0.666
```

圖 4- 10 SVM 訓練獲得之初版非線性模型內容

使用 libsvm-predict 工具對此非線性模型進行測試集預測，可得知初版非線性模型的預測準確率為 82%。將此邊界模型取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 56.4%；故可稱此傳統 SVM 非線性邊界模型，對於原有的專家規則對戰效果，可以提昇約 6.4% 的勝率。不難發現，此勝率提昇幅度也較線性初版模型為佳。

(二)、加入第一次投影點的第一版非線性模型:

將初版非線性模型內的支持向量 SV，輸入至本研究建置之非線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 262 個新的投影點，且透過專家檢查調整了其中 97 個投影點的類別。由於被修改的投影點類別並沒有超過投影點總數的一半，且投影點正例的數量較多，為了平衡正反兩類的資料筆數，本系統透過隨機取樣的方式，從初版非線性模型 SV 中選取了 45 個反例，再加上計算所得的第一次投影點，最後形成了數量為 307(=262+45)筆資料的訓練集，接著進行了 SVM 非線性模型訓練，並透過 grid.py 工具找到訓練優化參數 cost 為 32768 和 gamma 為 0.125，並進行了 SVM 非線性模型訓練，結果產生了支持向量數目為 78 個的第一版非線性模型，如圖 4-11 所示:

```
1 svm_type c_svc
2 kernel_type rbf
3 gamma 0.125
4 nr_class 2
5 total_sv 78
6 rho 300.975
7 label 1 -1
8 nr_sv 44 34
9 SV
10 32768 1:0.53 2:0.56 3:0.55 4:0.04 5:0.04 6:0.36 7:0.36
11 2875.854224373976 1:0.64 2:0.56 3:0.64 4:0.02 5:0 6:0.32 7:0.35
12 32768 1:0.53 2:0.56 3:0.55 4:0.04 5:0.04 6:0.36 7:0.36
13 12647.63182896363 1:0.54 2:0.52 3:0.55 4:0.05 5:0.04 6:0.37 7:0.37
14 32768 1:0.51 2:0.57 3:0.56 4:0.02 5:0.02 6:0.35 7:0.35
15 32768 1:0.53 2:0.56 3:0.55 4:0.05 5:0.05 6:0.36 7:0.36
16 10954.14382997721 1:0.52 2:0.41 3:0.44 4:0.03 5:0.02 6:0.35 7:0.69
17 32768 1:0.53 2:0.56 3:0.55 4:0.04 5:0.04 6:0.36 7:0.36
```

圖 4-11 第一版投影點經 SVM 訓練所得之非線性模型

使用 libsvm-predict 工具對此非線性模型進行測試集預測，可得知第一版非線性模型的預測準確率為 83%，預測準確度較初版非線性模型的準確度提升了 1%，因此直接進行對戰驗證。

將此邊界模型取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 57.9%；故可稱此第一版投影非線性

邊界模型，對於原有的專家規則對戰效果，可以提昇約 7.9% 的勝率。不難發現，此勝率提昇幅度仍較線性第一版模型為佳，且 SV 使用數目也遠少於線性第一版模型。

(三)、加入第二次投影點的第二版非線性模型:

將第一版非線性模型內的支持向量 SV，輸入至本研究建置之非線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 78 個新的投影點，且透過專家檢查調整了 42 個超過半數以上投影點的類別；究其原因為非線性投影開始集中至理想邊界模型的一邊附近，導致產生的投影點大部份都偏向正例。由於正例數量已經超過原先的第一版非線性模型的反例數量，無法透過只使用第一版非線性模型中的 SV 補充訓練資料，為了平衡正反兩類的資料筆數，本系統透過隨機取樣的方式，從原始訓練集適當選取 5 正例與 61 個反例，加入第二次投影點的訓練集內，最後產生 144(=78+5+61) 筆資料的訓練集。接著進行了 SVM 非線性模型訓練，並透過 grid.py 工具找到訓練優化參數 cost 為 32768 和 gamma 為 2，結果產生了支持向量數目為 48 個的第二版非線性模型，如 4-12 圖所示:

```
1 svm_type c_svc
2 kernel_type rbf
3 gamma 2
4 nr_class 2
5 total_sv 48
6 rho -58.9593
7 label 1 -1
8 nr_sv 21 27
9 SV
10 123.0517668622605 1:0.6 2:0.5 3:0.52 4:0.02 5:0.01 6:0.67 7:0.67
11 138.2528818030468 1:0.6 2:0.83 3:0.8 4:0.03 5:0.01 6:0.67 7:0.33
12 402.4994532441313 1:0.62 2:0.72 3:0.72 4:0.02 5:0.01 6:0.36 7:0.3
13 32768 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33
14 325.3004817085969 1:0.6 2:0.73 3:0.7 4:0.02 5:0.01 6:0.33 7:0.33
15 14812.73789860136 1:0.48 2:0.52 3:0.5 4:0.1 5:0.11 6:0.33 7:0.33
16 8609.718573830702 1:0.5 2:0.45 3:0.49 4:0.01 5:0 6:0.33 7:0.33
17 18317.50942689909 1:0.3 2:0.52 3:0.49 4:0.01 5:0.02 6:0.33 7:0.33
```

圖 4-12 第二版投影點經 SVM 訓練所得之非線性模型

使用 `libsvm-predict` 工具對此非線性模型進行測試集預測，可得知第二版非線性模型的預測準確率為 87.333%，預測準確度較第一版非線性模型的準確度提升了 5.3%，因此直接進行對戰驗證。

將此邊界模型取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 63.5%；故可稱此第二版投影非線性邊界模型，對於原有的專家規則對戰效果，可以提昇約 13.5% 的勝率，並且提昇的效果相較第一版非線性模型的 7.1% 的勝率還要高些。不難發現，此勝率提昇幅度仍較線性第二版模型為佳，且 SV 使用數目也遠少於線性第二版模型。

(四)、加入第三次投影點的第三版非線性模型

將第二版非線性模型內的支持向量 SV，輸入至本研究建置之非線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 48 個新的投影點。透過專家檢查，如同線性模型實驗發生的情況，發現許多投影點都會對應到即時戰略遊戲中的平均戰局狀況，因此極難明確定義其類別；在此情況下，專家還是盡可能地調整了其中 26 個投影點的類別，並且由於被修改的投影點類別，超過投影點總數的一半，為了平衡正反兩類的資料筆數，經過多次實驗後，本系統透過隨機取樣的方式，從初始訓練集適當選取 25 個正例與 70 個反例，加入第三次投影點的訓練集內，最後產生 $143(=48+25+70)$ 筆資料的訓練集。接著進行了 SVM 非線性模型訓練，發覺此時即使透過 `grid.py` 工具，也找不到其他的訓練優化參數；甚至從原始訓練資料集中嘗試透過多次隨機取樣，來改變訓練資料集，發覺效能也無法提昇。最後結果產生了支持向量數目為 34 個的第三版非線性模型，如圖 4-13 所示：


```

1 svm_type c_svc
2 kernel_type rbf
3 gamma 8
4 nr_class 2
5 total_sv 34
6 rho -9.34418
7 label 1 -1
8 nr_sv 17 17
9 SV
10 10.09575223212684 1:0.6 2:0.5 3:0.52 4:0.02 5:0.01 6:0.67 7:0.67
11 2273.931168502189 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33
12 735.7336755539585 1:0.48 2:0.52 3:0.5 4:0.1 5:0.11 6:0.33 7:0.33
13 3977.34519176507 1:0.5 2:0.47 3:0.48 4:0.01 5:0.01 6:0.33 7:0.33
14 1130.520903343421 1:0.3 2:0.52 3:0.49 4:0.01 5:0.02 6:0.33 7:0.33
15 58.17845613855383 1:0.6 2:0.47 3:0.54 4:0.02 5:0 6:0.33 7:0.33
16 5.54659938958404 1:0.5 2:0.58 3:0.59 4:0.01 5:0 6:0.67 7:0.67
17 8192 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33

```

圖 4-13 第三版投影點經 SVM 訓練所得之非線性模型

使用 libsvm-predict 工具對此非線性模型進行測試集預測，可得知非線性模型的預測準確率為 84%，預測準確度較第二版非線性模型的準確度下降了 2.7%。雖然預測準確度無法提升至第二版非線性模型之上，還是嘗試進行對戰驗證。

將此非線性邊界模型取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 58.6%；故可稱此第三版投影非線性邊界模型，相比於第二版投影點的非線性模型，效果下降了 4.9%。即使之後進行多次嘗試，也未能提昇其分類預測及對戰效果。

(五)、加入第四次投影點的第四版非線性模型:

將第三版非線性模型內的支持向量 SV，輸入至本研究建置之非線性投影程式內進行運算，因與 SV 逐一對應，故共產生了 34 個新的投影點，且透過專家檢查調整了 5 個投影點的類別，為了平衡正反兩類的資料筆數，本系統透過隨機取樣的方式，從原始訓練集適當選取 6 正例，加入第三次投影點的訓練集內，最後產生 40(=34+6)筆資料的訓練集。接著進行了 SVM 非線性模型訓練，並透過 grid.py 工具找到訓練優化參數 cost

為 2048 和 gamma 為 0.5，結果產生了支持向量數目為 21 個的第四版非線性模型，如 4-14 圖所示：

```

1 svm_type c_svc
2 kernel_type rbf
3 gamma 0.5
4 nr_class 2
5 total_sv 21
6 rho 23.5965
7 label 1 -1
8 nr_sv 10 11
9 SV
10 2048 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33
11 13.48428548360448 1:0.48 2:0.52 3:0.5 4:0.1 5:0.11 6:0.33 7:0.33
12 2048 1:0.5 2:0.47 3:0.48 4:0.01 5:0.01 6:0.33 7:0.33
13 2048 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33
14 2048 1:0.5 2:0.48 3:0.51 4:0.02 5:0.01 6:0.33 7:0.33
15 2048 1:0.5 2:0.48 3:0.51 4:0.02 5:0.01 6:0.33 7:0.33
16 9.963150289881501 1:0.5 2:0.5 3:0.53 4:0.02 5:0.01 6:0.33 7:0.33
17 944.6787011436847 1:0.5 2:0.5 3:0.53 4:0.02 5:0.01 6:0.33 7:0.33
18 2048 1:0.5 2:0.38 3:0.41 4:0.02 5:0.01 6:0.33 7:0.67
19 2048 1:0.5 2:0.45 3:0.46 4:0.01 5:0.01 6:0.33 7:0.33
20 -2048 1:0.5 2:0.45 3:0.49 4:0.01 5:0 6:0.33 7:0.33
21 -2048 1:0.5 2:0.45 3:0.49 4:0.01 5:0 6:0.33 7:0.33
22 -1953.869676070974 1:0.49 2:0.37 3:0.4 4:0.01 5:0 6:0.32 7:0.66

```

圖 4- 14 第四版投影點經 SVM 訓練所得之非線性模型

使用 libsvm-predict 工具對此非線性模型進行測試集預測，可得知第二版非線性模型的預測準確率為 84.333%，預測準確度較第三版非線性模型的準確度提升了 0.3%，因此直接進行對戰驗證。

將此邊界模型取代遊戲人工智慧原有的專家系統之進攻戰略決策規則，再與舊有的專家系統進行 1000 場對戰，數據顯示使用新的進攻戰略規則的遊戲人工智慧戰勝率為 58.8%；故可稱此第二版投影非線性邊界模型，相比於第三版投影點的非線性模型，效果上升了 0.2%但是並不顯著與研究限制的時間問題，所以決定終止了進攻戰略的非線性邊界學習程序。

(六)、非線性模型總結：

經過多次地針對投影點進行非線性模型訓練，以充當電腦玩家的進攻戰略選用規則的實驗後，如同線性模型訓練，使用兩次投影點進行訓練的效果最好。

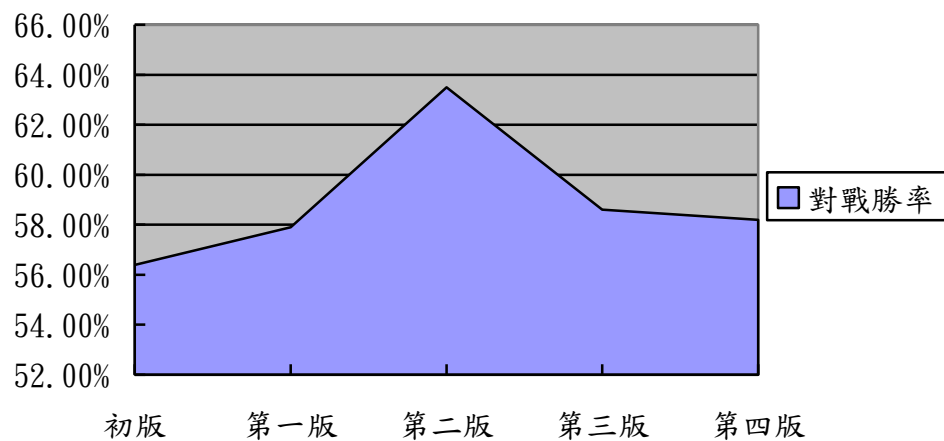


圖 4-15 各版本非線性模型的勝率

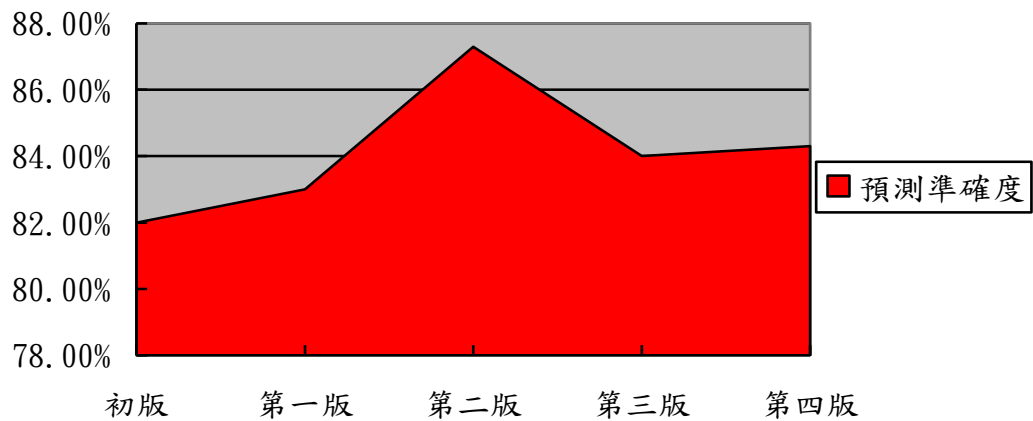


圖 4-16 各版本非線性模型的預測準確度

如圖 4-15 所示，可以達到將遊戲人工智慧的勝率提昇至 63.5%。在即時戰略遊戲中，由於遊戲中具有相當程度的博弈成分，因此透過機器學習的機制，能夠達到提昇 13.5% 的勝率，已足可稱為具有顯著的學習成效[16]。並且由對戰勝率與預測準確度的走勢圖發現，與線性模型的實驗結果類似，兩者具有同昇同降的效果，這足以驗證本實驗所採用的測試資料集，確實足以代表實戰母體本身。此外，兩種 SVM 投影模型的專家調整困難情況都在第三次投影時發生，並且第三版模型的預測準確度與對戰勝率也都較第二版模型為低，究其原因可能是已達到投影方式

或是專家辨識類別的極限。如果是後者，或許可透過額外設計的遊戲對戰程式，來協助或取代人類專家進行困難的投影點類別的判斷，此問題將會在最後的章節再進行深入的探討。



表 4-9 各版本之非線性分界模型訓練資料集

	初版線性 模型	第一版線 性模型	第二版線 性模型	第三版線 性模型	第四版線 性模型
原始訓練集 資料	400	0	66	95	0
投影程式產 生	0	262	78	48	34
上一版本分 界模型支持 向量	0	45	0	0	6
專家修改投 影點類別數	0	97	42	26	5
訓練集總數	400	307	144	143	40
邊界模型支 持向量數	262	78	48	34	21

第四節 對戰歷程分析

經過前述的 SVM 邊界模型訓練後，各自從線性與非線性模型實驗中，獲得準確率最高的邊界模型，分別為第二版線性與第二版非線性模型。為了觀察兩者在對戰過程的差異，本研究將這兩種邊界模型與原有的專家規則，分別進行並記錄了 1000 場對戰；為了表現出遊戲歷程，本研究從遊戲開始到結束，每經過遊戲時間 10 秒，就紀錄一次整場遊戲的各項指標變化。

由於每場遊戲的對戰時間不一定相同，為了方便觀察，本研究將所有對戰場次的資料時間軸，都透過正規化處理，亦即皆以遊戲進行歷程百分比的方式來呈現。

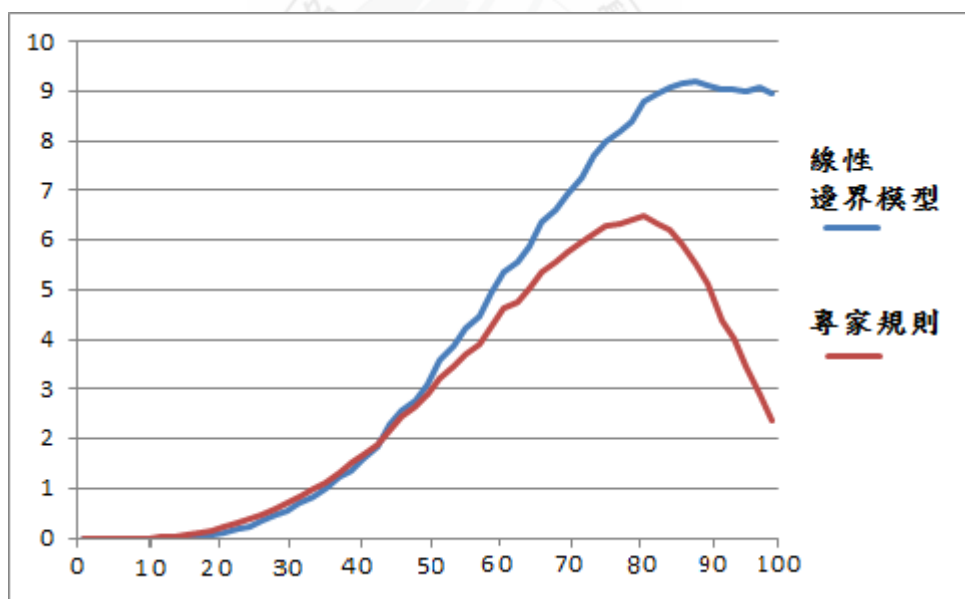


圖 4- 17 線性模型中攻擊部隊數量的變化歷程

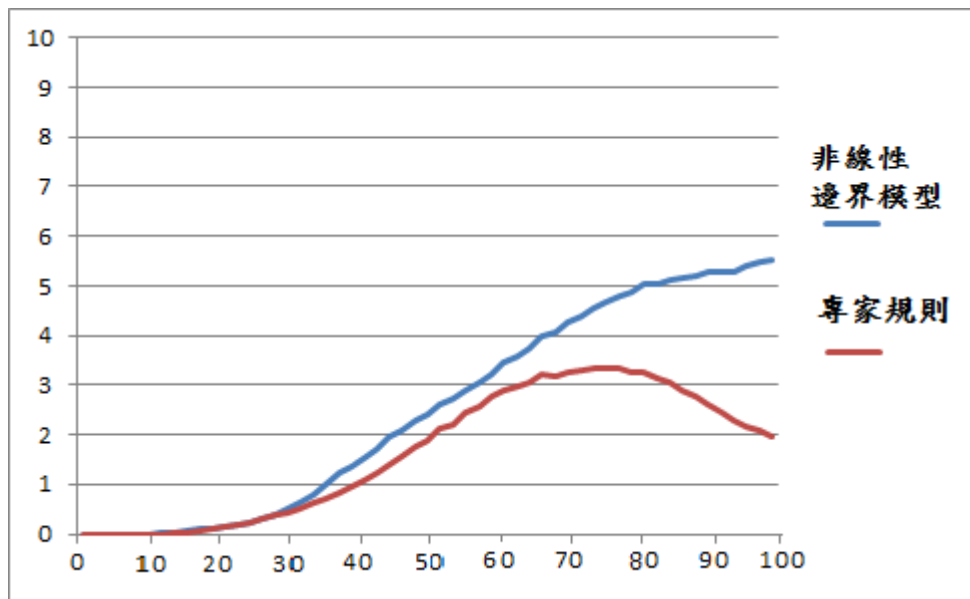


圖 4-18 非線性模型中攻擊部隊數量的變化歷程

為了理解兩種不同的邊界模型的差異，本研究將會透過變化較明顯的遊戲指標來進行分析，其中 X 軸為對戰歷程百分比，Y 軸為特定指標數值：

如圖 4-17 與 4-18 所示，線性模型的攻擊部隊的數量，從遊戲歷程 80% 左右(進入遊戲後期)，傾向保持在平均數值為最高 9 的狀態，持續到遊戲結束；相對地，非線性模型將會讓遊戲的平均攻擊部隊數量壓低至 5.5 以下，並且從能在遊戲時間經過 30% 的時候，就開始保持攻擊部隊數量領先的狀況，並且透過不斷地進攻來限制住敵人攻擊部隊的發展。

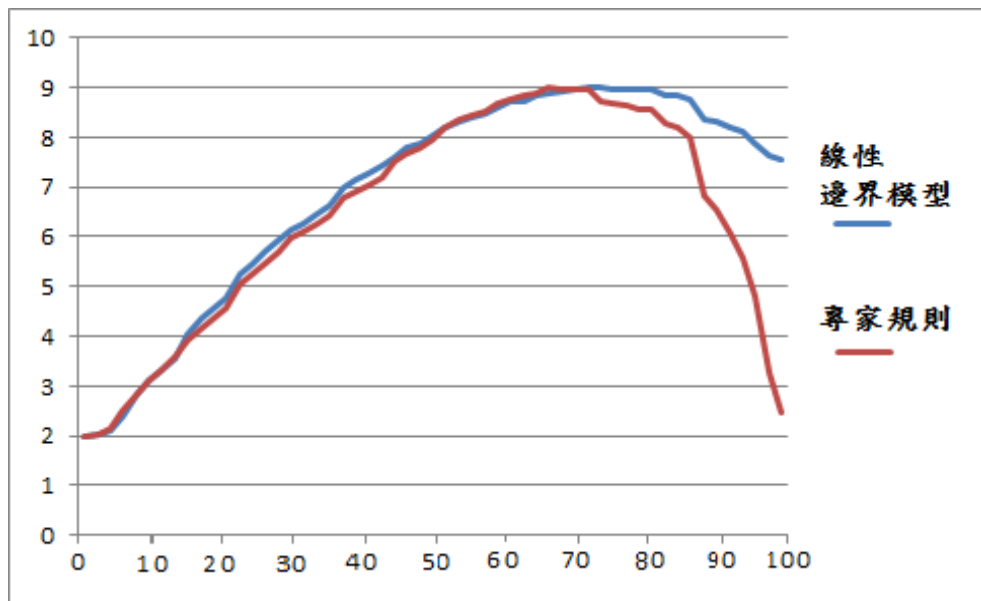


圖 4- 19 線性模型中工作部隊數量的變化歷程

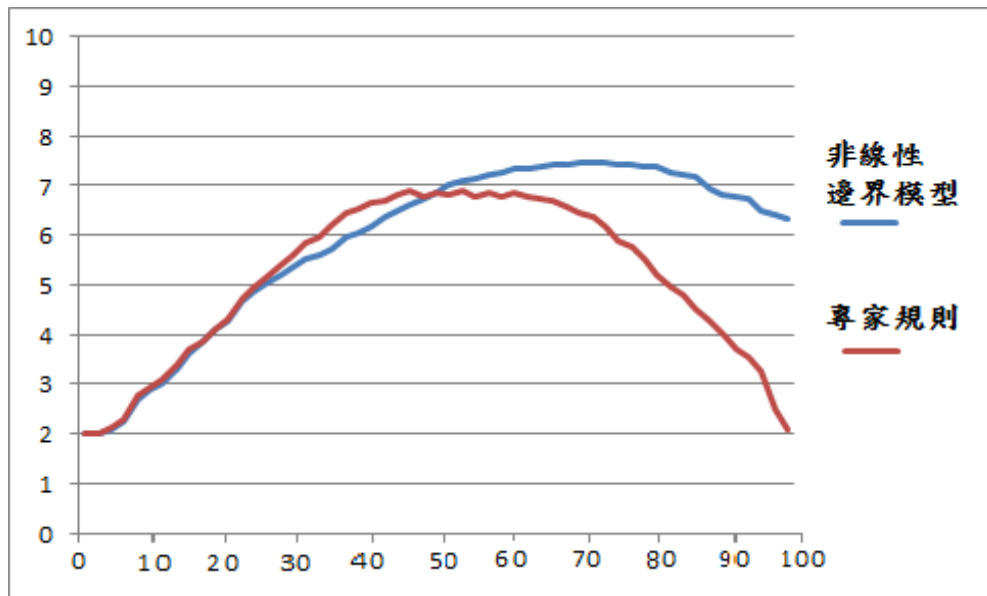


圖 4- 20 非線性模型中工作部隊數量的變化歷程

先行觀察圖 4-20，可以發現在遊戲歷程經過 30% 的時候，非線性模型的工作部隊數量，雖然少於專家規則模型；但是在遊戲歷程經過 50% 之後，將會開始超越後者，這或許與圖 4-18 顯示的非線性模型攻擊部隊數量，在遊戲早期就開始壓制專家規則模型的現象有關；

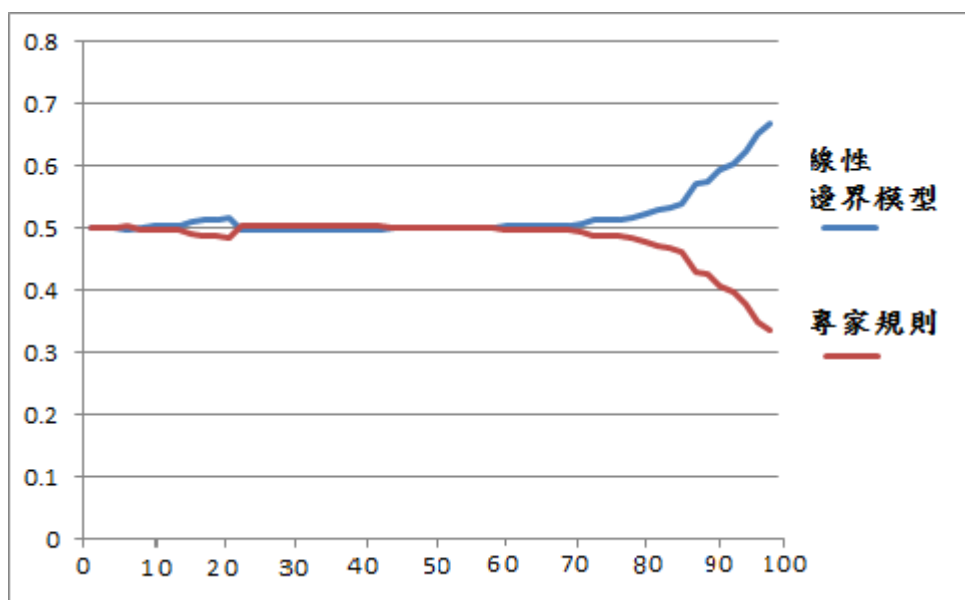


圖 4-21 線性模型中雙方玩家分數評估的變化歷程

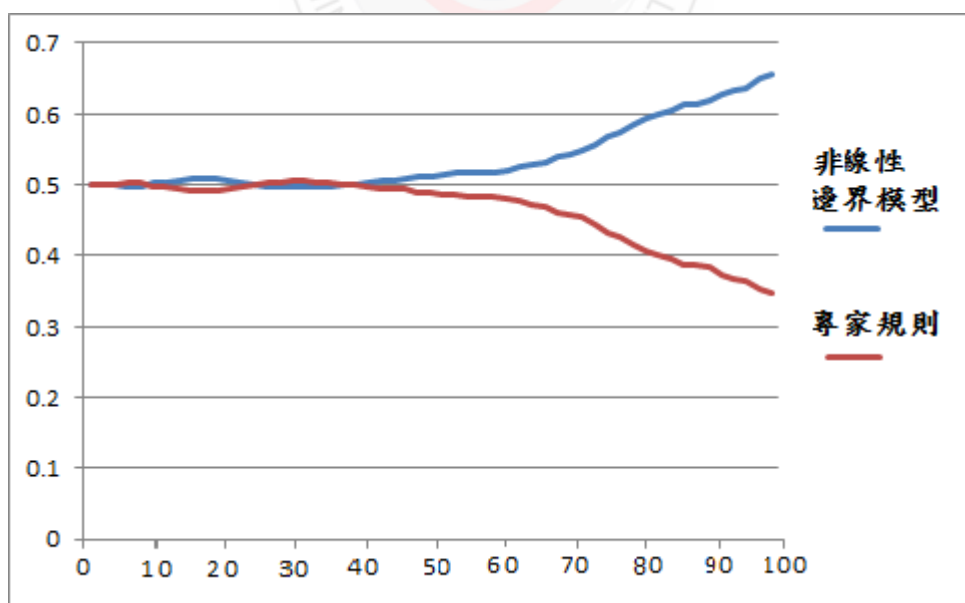


圖 4-22 非線性模型中雙方玩家分數評估的變化歷程

換言之，非線性模型可能採取，遊戲前期犧牲經濟國力來進行快攻，並得以在遊戲中期，壓制專家規則模型工作部隊的發展，之後工作部隊數量反超後者，以終於獲取遊戲優勢的情況。

從圖 4-19 顯示，反觀線性模型在遊戲前中期，並不會限制專家規則模型的工作部隊數量，兩邊的工作部隊規模一直發展到遊戲後期(遊戲歷程 70%)，才開始嘗試消滅對手的工作部隊。所以參考圖 4-17 與圖 4-19 的情況，發現線性模型會讓遊戲的總部隊數量規模，發展至較大的情況。

經過統計，非線性模型對戰的平均遊戲時間為 208 秒且最大遊戲時間為 550 秒，而線性模型對戰的平均遊戲時間為 224 秒且最大遊戲時間為 780 秒，不難看出線性模型的對戰時間通常需時較久。觀察圖 4-21，發現線性模型的雙方玩家分數評估在遊戲後期(遊戲歷程 70%)，才會出現明顯的變化；並參考最久遊戲時間可以推測，線性模型容易讓遊戲，陷入雙方長時間不進攻的僵局狀況。相對地，如圖 4-22 所示，非線性模型則是經常在遊戲中期(遊戲歷程 45%)就開始拉開雙方玩家分數評估的差距。在綜合以上的觀察之後，不難推測出非線性模型採取的策略，較偏向快攻型的遊戲玩家；而線性模型採取的策略，較偏向保守型玩家，會比較謹慎地培養攻擊部隊，等到數量達到優勢程度後，才會開始進攻。

第五章 結論與未來展望

本章將敘述本研究的結論，和進行實驗所遭遇的問題、及探討，也針對本研究提出未來展望和建議。

第一節 結論

為了能完整控制即時戰略遊戲內的所有程式模組，本研究自行設計暨開發了一個即時戰略遊戲離型系統，並先行實作出專家規則，來作為初始的遊戲人工智慧模組；在其運作之下，得以蒐集了利用兩個人工智慧模組，進行實際對戰的遊戲歷程資料，並以七種遊戲狀態指標來呈現。接著再導入了線性與非線性兩種核函數之支持向量機的訓練機制，並嘗試結合了支持向量投影和專家諮詢等觀念[6]，設計出適合本遊戲的訓練流程，得以在嚴重雜訊干擾下的環境下，來訓練出本遊戲進攻戰略適用時機的邊界模型。最後，與使用初始進攻戰略的遊戲人工智慧模組，進行了 1000 場對戰；實驗結果顯示，第二版非線性進攻戰略模型的對戰效果最好，勝率可達 63.5%，圖 5-1 為星海爭霸 2 宗師級玩家天梯排行榜，而宗師級玩家為全遊戲伺服器遊玩級別最高的玩家，從圖中可以看出就算以遊戲伺服器前 1% 的宗師玩家所經歷的對戰歷程來看，前位玩家的勝率大部分也是坐落於約 63% 勝率的對戰歷程，由此來看即時戰略遊戲在同遊戲級別的玩家對戰中，勝率到達約 63% 的情況及為同級別中的前位玩家，故可稱體顯示出整個學習機制的有效性。此外，遊戲歷程的指標分析，也可觀察出線性和非線性模型學習到的策略，會分別對應到保守型及快攻型的遊戲玩家。在電子競技娛樂發達的年代，希望本研

究可提供即時戰略遊戲人工智慧研究者的一種新作法，也期望能夠提昇人類玩家的遊戲體驗。

2017年第2賽季 - 1對1 宗師爭奪者				
排名	名稱	對戰積分	勝場	敗場
50		5328	57	45
51	 valhalla	5328	474	404
52		5326	32	21
53	 bbO	5320	377	300
54		5316	265	236
55	 <906> 九零六	5316	297	272
56	 <인투더> IntoTheSonni	5313	188	163
57		5312	108	97
58	 King	5307	548	511
59		5302	209	189
60		5300	79	69
61	 <Kholin> 玛哈念	5299	81	74
62		5294	132	125

圖 5-1 星海爭霸 2 天梯宗師排行榜

整體而言，雖然本遊戲環境和市面上的即時戰略遊戲相較之下，設計稍嫌簡單，但是透過實驗的遊戲紀錄數據顯示，還是符合了基本的三種即時戰略相互相剋的特性。可惜由於研究的諸多限制，本研究只完成了三種戰略中進攻戰略適用時機的訓練，至於戰術跟戰技層級的訓練，也來不及進行進一步的探索。

此外，本研究在訓練資料的格式與資料收集的門檻上也都曾進行多種實驗，由於雙方國力分數差異性門檻越低，所蒐集的訓練資料數量會越多，對於結果也會產生具體的影響，所以雙方國力分數差異性門檻是一個需要注意的部份。

另外，本研究也透過比較各種雙方國力分數門檻所蒐集的資料進行 SVM 初版線性模型訓練，透過數據可以猜測在初次的 SVM 線性模型訓練時，蒐集資料的雙方國力分數門檻太低或許會造成邊界模型的不適合狀況，反而下降對戰勝率，以本研究的看法而言，可能的原因為初次線性模型訓練所使用的訓練資料分的正反例辨別度不高，導致學習效果可能下降，但是本研究還無法證實這部份。

另外，為了遊戲進行的多樣性，本研究在遊戲人工智慧模組運行時，尚且加入了相當程度(30%)的隨機要素。雖然很可能導致最後進攻戰略適用時機的邊界模型，與初始的遊戲人工智慧模組，實際對戰勝率未臻理想；但若與使用完全隨機的遊戲人工智慧模組對戰，實際勝率可高達 81%；本研究認為此現象或許跟訓練資料裏隱含的隨機性有關。

第二節 遭遇問題

本遊戲系統初期版本，在對戰平衡上的設定並不是很成功，導致無法具體呈現出，即時戰略遊戲中的三種戰略彼此相剋的特性。因此研究者需透過不斷地實際觀察遊戲人工智慧對打的過程，進而多次調整容易導致失衡問題的因素，主要計有：攻擊部隊的移動速度、工作部隊的生命力、基地所擁有的反擊能力與其他較小的項目，最後才能確定各種遊戲部隊的參數，以呈現出較佳的遊戲系統對戰平衡效果。

此外，研究人員在過往的研究議題上，並沒有使用 Unity3D 的程式開發經驗，所以在適應開發環境上，花費不少時間；研究人員也未具備美術專長，因此遊戲系統在美術效果方面，僅能達到最低門檻要求。

由於本論文所使用的即時戰略系統是自行建置，並非市面上擁有眾多玩家的商用即時戰略遊戲，因此要找到對本遊戲精通的專家，成為本

研究推進時的一大難題；幸而在多次嘗試下，使用初始專家規則模組與完全隨機對戰時，實驗數據顯示還是具備了 67% 的戰勝率，代表初版的專家規則，已經具有相當的遊戲能力。然而，也正由於領域專家匱乏的問題，造成在後續進行 SVM 訓練，需要檢查並且修改投影點類別時，專家的能力更受到侷限，甚至可能將投影點類別修改失誤，反而降低了原來 SVM 的準確度；因而造成研究者需要花費許多時間，來反覆進行不斷的嘗試和修改測試。在後續的未來展望中，本研究也提出了一種可能有助於判斷投影點類別的改善方法。

此外，本研究嘗試利用林智仁(Lin Chih-Jen)教授所開發的 Libsvm 程式套件[19]，來進行遊戲人工智慧戰略規則的訓練；然而，對於研究者而言，是首次接觸到 SVM 分類模型。在研究初期，在 SVM 模型檔的解讀方面，研究者曾遭遇到解讀錯誤的情況，因此也浪費了不少時間；但在透過實際撰寫預測準確率的計算程式後，得以確認了 SVM 模型檔正確的解讀和應用方法，也加深了對於 SVM 分類模型的認知和操作能力。

第三節 未來展望

研究者發現在特定情況下，本研究採用的資料格式，所能呈現的資訊仍不夠完整。例如原格式僅保留攻擊部隊的數量，卻沒有紀錄攻擊部隊的生命值；所以當一個生命滿值的攻擊部隊，與兩個生命值只剩下一點的攻擊部隊進行對戰，正常戰鬥的結果，反而應該是數量較少卻生命滿值的攻擊部隊獲取勝利，但是本研究並未考慮到此現象。建議在往後的軍事國力分數評估，與攻擊部隊數量的紀錄項目內，可以涵蓋到攻擊部隊生命值狀況的資訊；或是將各攻擊部隊群，用一個以上的紀錄指標儲存，並且將每一個攻擊部隊受過的損傷，乘上權重值後再行紀錄。

另外，關於需要專家人工來判斷，SVM 模型產生的投影點類別，對於遊戲的討論熱門度較低，且沒有專業玩家的即時戰略遊戲而言，非常困難；但假如專家在判斷投影點時，修改的分類失誤，反而會造成學習上的反效果。本研究建議一種可以輔助專家，進行投影點類型分類的方法，其流程如圖 5-2 所示：作法為在遊戲開始時，就將投影點的相關資料，

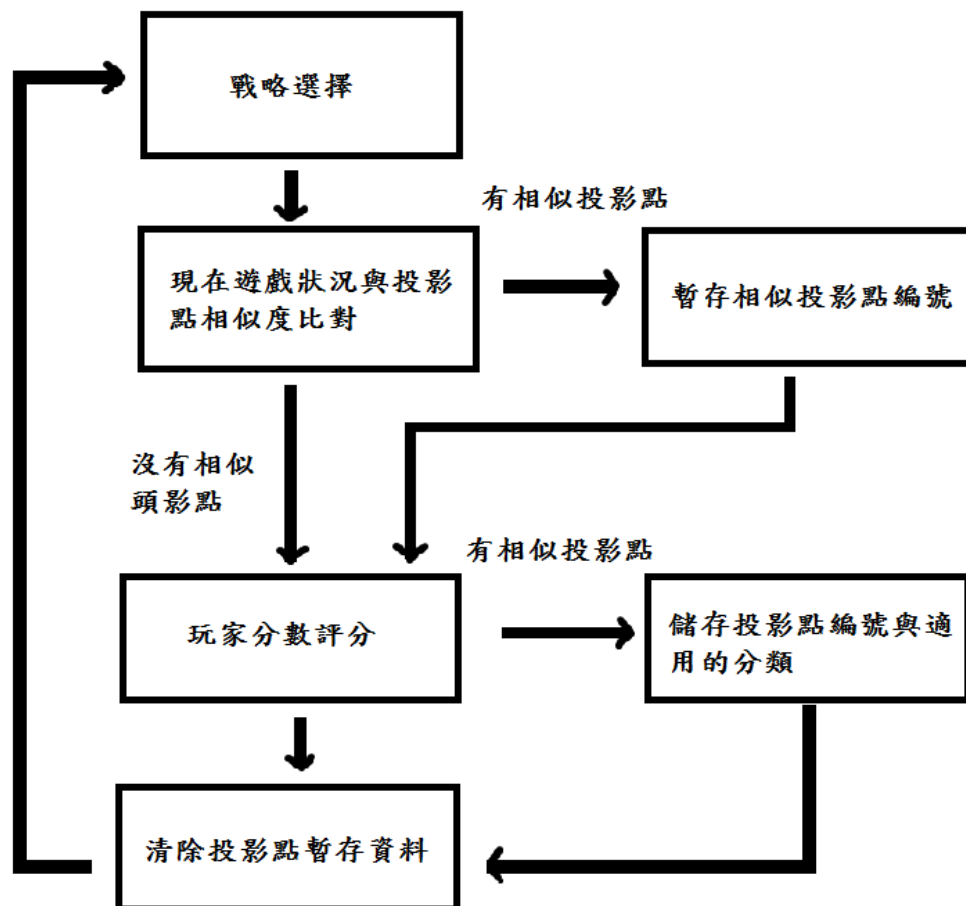


圖 5-2 假設的投影點驗證程式流程圖

載入遊戲人工智慧的程式內，並且在遊戲過程中，進行戰略選擇時，也同步搜尋是否具有相似或是吻合投影點資料的情況；如果發現類似投影點的遊戲情況發生，則將此次戰略選擇之後的對戰效果，作為該投影點

資料的正確類別，提供專家參考，以協助專家對於難以分類的投影點資料的判斷工作。



參考文獻

中文部份

1. 王正宏，(民 96)，在即時戰略遊戲中探勘與預測玩家的策略風格，國立交通大學資訊科學與工程研究所論文。
2. 王榮英、李瑋翰，《使用機器學習理論建構遊戲中非玩家角色之情緒變化》，龍華科技大學學報，第 31 期，民國 101 年 7 月，頁 61~72。
3. 吳忠霖，(民 93)，即時戰略連線遊戲基礎架構之研究，國立台灣科技大學機械工程系碩士論文。
4. 邱菊添、洪宗貝，《人工智慧技術於電腦遊戲之應用》，全球商業經營管理學報，第 3 期，民國 100 年 9 月，頁 33~741。
5. 黃世瑋，(民 102)，即時戰略遊戲之使用性研究，國立東華大學藝術與設計學系碩士論文。
6. 陳鴻文，(民 106)，詢問式支持向量機之推導與應用，私立銘傳大學資訊傳播工程學系技術文件。
7. 游源惠、黃建軒，《導入 SVM 之棋類人工智慧-以四連環遊戲為例》，2006 數位科技與創新管理國際研討會，頁 685~695，華梵大學，2006 年。
8. 維基百科，即時戰略遊戲條目，
<https://zh.wikipedia.org/zh-tw/%E5%8D%B3%E6%97%B6%E6%88%98%E7%95%A5%E6%B8%B8%E6%88%8F/>，上網日期：2017 年 6 月 15 日。

9. 幻想戰略譚 The Touhou Empire RTS 攻略，

https://www27.atwiki.jp/th_empire/pages/34.html/，上網日期：2017 年 6 月 15 日。

10. Unity3D 官方網站，網址：<https://blogs.unity3d.com/>，上網日期：2017 年 6 月 15 日。

11. 科技報橘 TechOrange，《Alpha Go 將挑戰《星海爭霸 2》，當人工智慧也能秒反應，代表人類智力正被步步逼近》，

<https://buzzorange.com/techorange/2016/11/07/google-deepmind-challenge/>，上網日期：2017 年 6 月 15 日。

12. 机器之心，《人工智能玩《星际争霸》，水平赶上顶尖人类玩家》，

https://baijia.baidu.com/s?old_id=253102/，上網日期：2017 年 6 月 15 日。

13. 遊戲基地，星海爭霸 2 專區，<http://sc2.zone.gamebase.com.tw/zone/sc2/>，

上網日期：2017 年 6 月 15 日。

英文部份

14. Tyrrell, T., Computational Mechanisms for Action Selection. 1993, University of Edinburgh.

15. Tyrrell, T., The Use of Hierarchies for Action Selection. Adaptive Behavior, 1993. 1(4): p.387.

16. 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

17. Burges, C.J.C., “A tutorial on support vector machine for pattern recognition,” Data mining and Knowledge Discovery, Vol. 2, No. 2, pp. 121-167, June 1998.

18. 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.
19. Chih-Chung Chang and Chih-Jen Lin, LIBSVM -- A Library for Support Vector Machines, online resource, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> 【June ,2017】
20. EA Video Games, Command & Conquer Red Alert, <http://www2.ea.com/cnc> 【June ,2017】
21. AlphaGo | Deepmind, <https://deepmind.com/research/alphago/> 【June ,2017】
22. AIIDE StarCraft AI Competition , <http://www.cs.mun.ca/~dchurchill/starcraftaicom/> 【June ,2017】
23. Proleague - Liquipedia - The StarCraft II Encyclopedia , <http://wiki.teamliquid.net/starcraft2/Proleague/> 【June ,2017】

附錄

1.進攻戰略規則:

雙方軍事狀態評估 ≥ 0.6 且 雙方經濟狀態評估 ≥ 0.3

或 雙方軍事狀態評估 ≥ 0.7

A.多線打擊戰術規則:

敵方基地部隊數量-我方基地部隊數量 ≥ 1 且 雙方經濟狀態評估 ≥ 0.35

部隊戰技:

基地:

if (資源 ≥ 300 & 該基地區域工作部隊 < 5) ==>生產工作部隊

else if (資源 ≥ 700 & 該基地區域工作部隊 > 4) ==>升級建造速度科技

else if (資源 ≥ 450 & 該基地區域工作部隊 ≥ 5) ==>生產攻擊部隊

工作部隊:

採集資源

現有的攻擊部隊:

敵方最新生產基地(最外側基地)

之後生產的攻擊部隊:

攻擊敵方主基地(一開始的基地)

B.正面攻擊戰術規則：

在進攻戰略選擇下除了多線打擊戰術規則以外的情況

部隊戰技：

基地：

if (資源 \geq 300 & 該基地區域工作部隊 $<$ 5) ==>生產工作部隊
else if (資源 \geq 700 & 該基地區域工作部隊 $>$ 4) ==>升級建造速度科技
else if (資源 \geq 450 & 該基地區域工作部隊 \geq 5) ==>生產攻擊部隊

工作部隊：

採集資源

現有的攻擊部隊：

攻擊敵方主基地(一開始的基地)

之後生產的攻擊部隊：

攻擊敵方主基地(一開始的基地)



2.擴張戰略規則:

雙方軍事狀態評估 ≥ 0.5 且 雙方經濟狀態評估 ≥ 0.4

A,擴張最大化戰術規則:

雙方軍事狀態評估 > 0.55 且 雙方經濟狀態評估 ≥ 0.5

部隊戰技:

基地:

if (資源 ≥ 1200 & 我方基地數 < 3) \Rightarrow 生產基地部隊

else if (資源 ≥ 300 & 該基地區域工作部隊 < 7) \Rightarrow 生產工作部隊

else if (資源 ≥ 450 & 我方基地數 ≥ 3) \Rightarrow 攻擊部隊

工作部隊:

\Rightarrow 採集資源

現有的攻擊部隊:

\Rightarrow 在基地巡邏防守

之後生產的攻擊部隊:

\Rightarrow 在基地巡邏防守

B.擴張防守戰術規則：

在擴張戰略選擇下除了擴張最大化戰術規則以外的情況

部隊戰技：

基地：

if (資源 \geq 450 & 我方攻擊部隊數-敵方攻擊部隊數量 \leq 2) \Rightarrow 生產攻擊部隊

else if (資源 \geq 1200 & 我方基地數 $<$ 2) \Rightarrow 生產基地部隊

else if (資源 \leq 1200 & 該基地區域工作部隊 $<$ 7) \Rightarrow 生產工作部隊

else if (資源 \geq 450 & 該基地區域工作部隊 \geq 7) \Rightarrow 生產攻擊部隊

工作部隊：

\Rightarrow 採集資源

現有的攻擊部隊：

\Rightarrow 在基地巡邏防守

之後生產的攻擊部隊：

\Rightarrow 在基地巡邏防守



3.防守戰略規則

除了進攻戰略與擴張戰略之外的情況

A.少量部隊騷擾戰術規則：

敵方基地部隊數量-我方基地部隊數量 ≥ 1 且雙方軍事狀態評估 ≥ 0.45

部隊戰技：

基地：

if (資源 ≥ 300 & 該基地區域工作部隊 < 5) ==> 生產工作部隊

else if (資源 ≥ 700 & 該基地區域工作部隊 ≥ 4) ==> 升級建造速度

else if (資源 ≥ 450 & 該基地區域工作部隊 ≤ 6) ==> 生產攻擊部隊

工作部隊：

==> 採集資源

現有的攻擊部隊：

==> 敵方最新生產基地(最外側基地)

之後生產的攻擊部隊：

==> 在基地巡邏防守

B.防守最大化戰術規則:

在防守戰略選擇下除了少量部隊騷擾戰術規則以外的情況

部隊戰技:

基地:

if (資源 \geq 450 & & 我方攻擊部隊數-敵方攻擊部隊數量 \leq 1) ==>生產攻擊部隊

else if (資源 \geq 700 & 該基地區域工作部隊 \geq 4) ==>升級建造速度

else if (資源 \geq 300 & 該基地區域工作部隊 \leq 6) ==>生產工作部隊

else if (資源 \geq 450) ==>生產攻擊部隊

工作部隊

==>採集資源

現有的攻擊部隊:

==>在基地巡邏防守

之後生產的攻擊部隊:

==>在基地巡邏防守

