

國立交通大學

電機與控制工程研究所

碩士論文

以關聯式法則建立共生式人工免疫系統之抗體



**Using Association Rules for Creating the Assembly
Algorithm of a Symbiotic Artificial Immune System**

研究生：周家興

指導教授：林昇甫 博士

中華民國一百年七月

以關聯式法則建立共生式人工免疫系統之抗體
集合

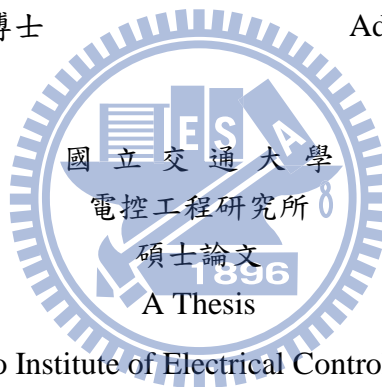
**Using Association Rules for Creating the Assembly
Algorithm of a Symbiotic Artificial Immune System**

研究生：周家興

Student: Chia-Hsing Chou

指導教授：林昇甫 博士

Advisors: Dr. Sheng-Fuu Lin



Submitted to Institute of Electrical Control Engineering

College of Electrical Engineering

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Electrical Control Engineering

June 2011

Hsinchu, Taiwan, R.O.C

中華民國一〇一年七月

以關聯式法則建立共生式人工免疫系統之抗體集合

研究生：周家興

指導教授：林昇甫 博士

國立交通大學 電機與控制工程學系

摘要

在人工免疫系統中，抗體代表其問題解。而由基因演算法中合作式的概念衍伸出了共生式人工免疫系統，而抗體基元(partially specified antibody)在這當中則是代表其部分解。當抗體基元間非線性獨立而有相關聯性或者面對高維度的問題時，將抗體基元全部放在同一族群一起演化，容易使演算法效能低落。本論文提出以頻繁樣式成長法尋找抗體基元各維度間的關聯性，其關聯性的強弱將決定其分類規則。合理的分類規則可以使演算法在效能上有更佳的表现。因此對於共生式人工免疫系統來說，抗體基元分類為整個演化過程中相當重要的一部分。本論文由抗體基元分類根據維度關聯性訂立分類規則，可視問題不同做調整，在使用上可適用於多種問題類型。

關鍵字：原始人工免疫系統、共生式人工免疫系統、頻繁樣式成長。

Using Association Rules for Creating the Assembly Algorithm of a Symbiotic Artificial Immune System

Student: Chia-Hsing Chou

Advisor: Dr. Sheng-Fuu Lin

Department of Electrical and Control Engineering

National Chiao-Tung University

Abstract

In artificial immune system, antibodies represent the solutions. With the cooperative concept in genetic algorithm, the symbiotic artificial immune system has been developed, and partially specified antibodies are defined as the partial solutions. When the partially specified antibodies are not linear independent; have correlation with the others, or face the high-dimensional problem, put all partially specified antibodies in the same swarm which result in low efficiency in the algorithm. Frequent pattern growth algorithm is adopted in this thesis to find the correlation between each dimension, in which the strength of the connectivity determines whether the classification rules are valid or not. Generally, reasonable classification rules cause the algorithm work more effectively. Hence, for the algorithm, the partially specified antibody classification plays a very important role in the evolutionary process. The classification rules defined in this thesis are determined according to the correlation between each dimension, and they are adjustable under different problem, which greatly enhance the applicability of many kinds of problems.

Keywords : AIS, SymbAIS, FP-growth.

誌謝

首先以最誠摯的謝意，感謝我的指導教授林昇甫博士，感謝他在研究上的啟迪與指導，以及細心校閱本論文的謬誤之處。同時也要感謝本實驗室的學長永吉、建焜、逸章、俊偉、啟耀、晉嘉、國育；一起同甘共苦同學世雄、煒清、家昌、以及學弟妹們，有如此龐大的研究團隊，使得我在遇到問題時都能很充裕的能找到人來對我伸出援手，讓論文的研究能夠順利完成，並且在碩士生涯中有好大一票的好伙伴一起成長、一起學習；不只是書本上的知識，還有更多更多有趣有用的技能以及想法。過程中也許有苦有樂，但是在回憶中卻是既燦爛又豐富。

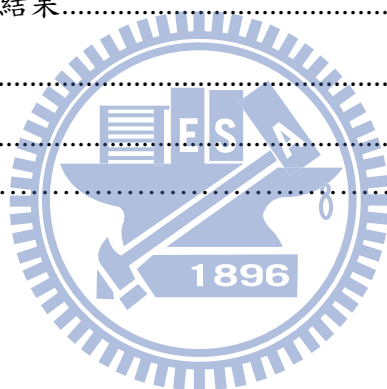
最後要感謝我親愛的父母，感謝他們的養育之恩以及在我求學生涯中給我最大的鼓勵與支持，使我得以在精神與生活上無後顧之憂，順利完成學業。



目錄

摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖目錄	vi
表目錄	viii
第一章 緒論.....	1
1.1 相關研究	2
1.2 研究動機	3
1.3 論文架構	4
第二章 背景知識	5
2.1 免疫系統.....	5
2.1.1 生物免疫系統.....	5
2.1.2 免疫機制.....	6
2.2 人工免疫系統.....	8
2.3 共生式人工免疫演算法.....	13
2.4 資料探勘.....	14
2.4.1 發掘關聯式規則.....	16
2.4.2 頻率項目成長演算法.....	18
2.5 模糊類神經網路.....	20
2.5.1 模糊系統.....	20
2.5.2 模糊化類神經網路.....	21
第三章 基於頻繁樣式成長法的共生式人工免疫系統.....	24
3.1 演算法概述	24
3.2 基於頻率項目成長的共生式人工免疫系統.....	25
3.2.1 抗體基元分類規則確立.....	26

3.2.2分類規則確立後最佳化演算法.....	33
3.3函數模擬架構.....	34
3.3.1架構模糊類神經網路.....	35
3.3.2最佳化過程.....	38
第四章 實驗數據	39
4.1函數模擬.....	39
4.1.1函數類型選取.....	39
4.1.2函數模擬實驗結果.....	40
4.2 函數最佳化	47
4.2.1最佳化方程式選取.....	47
4.2.2函數最佳化結果.....	49
4.3 模擬分析	60
第五章 結論	62
參考文獻.....	63



圖目錄

圖 1.1株落選擇原理示意圖。	3
圖 2.1生物免疫系統架構。	6
圖 2.2免疫應答示意圖。	7
圖 2.3人工免疫系統流程圖。	9
圖 2.4抗體群架構圖。	11
圖 2.5 產生抗體集合演算法。	12
圖 2.6共生式人工免疫系統流程圖。	13
圖 2.7資料探勘流程圖。	15
圖 2.8發掘關聯式規則簡例。	17
圖 2.9資料指定並排序。	18
圖 2.10建構頻率項樹。	19
圖 2.11 TSK形式模糊類神經網路的架構。	23
圖 3.1基本流程。	25
圖 3.2實際分類情況。	27
圖 3.3本文測試資料分類情況。	28
圖 3.4族群分類詳細流程。	30
圖 3.5 族群分類簡例。	31
圖 3.6族群分類簡例頻繁項樹。	32
圖 3.7基於頻繁樣式成長法的共生式人工免疫系統演化示意圖。	34
圖 3.8 本論文所使用TSK模糊類神經網路架構。	36
圖 3.9函數模擬流程圖。	38
圖 4.1函數一模擬結果:(a)以原始人工免疫系統調整TSK參數(b)以共生式人工免疫系統調整TSK參數(c)以本論文調整TSK參數。	41

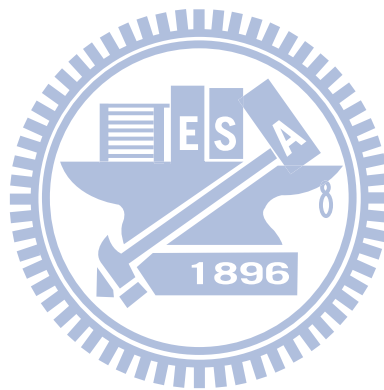
圖 4.2三種方法誤差值。.....	42
圖 4.3函數二模擬結果：(a)實際圖型，(b)以原始人工免疫系統調整TSK 參數，(c)以共生式人工免疫系統調整TSK參數，(d)以本論文調整 TSK參數。.....	44
圖 4.4函數三模擬結果：(a)實際圖型，(b)以原始人工免疫系統調整TSK 參數，(c)以共生式人工免疫系統調整TSK參數，(d)以本論文調整 TSK參數。.....	45
圖 4.5 Rosenbrock function實際圖形。.....	47
圖 4.6 Ackley's function之實際圖形。.....	48
圖 4.7 generalized Rastrigin function實際圖形。.....	48



表目錄

表 2.1 生物免疫系統與實際應用問題名詞對照表。	8
表 4.1 函數初始設定。	40
表 4.2 函數一模擬結果。	41
表 4.3 函數模擬中函數一 λ 不同時之數據。	43
表 4.4 函數一利用歸屬函數之重疊性調整規則數前後之時間比較。	43
表 4.5 函數二模擬結果。	44
表 4.6 函數模擬中函數二 λ 不同時之數據。	44
表 4.7 函數二利用歸屬函數之重疊性調整規則數前後之時間比較。	45
表 4.8 函數三模擬結果。	45
表 4.9 函數模擬中函數三 λ 不同時之數據。	46
表 4.10 函數三利用歸屬函數之重疊性調整規則數前後之時間比較。	46
表 4.11 實驗參數初始設定。	49
表 4.12 f_0 抗體基元數目測試結果。	50
表 4.13 f_1 抗體基元數目測試結果。	51
表 4.14 f_2 抗體基元數目測試結果。	52
表 4.15 f_0 使用本論文方法實驗結果。	53
表 4.16 f_0 使用 AIS 方法實驗結果。	54
表 4.17 f_0 使用 SymbAIS 方法實驗結果。	54
表 4.18 方程式 f_0 當 λ 不同時之數據。	55
表 4.19 f_0 利用歸屬函數之重疊性調整規則數前後之時間比較。	55
表 4.20 f_1 使用本論文方法實驗結果。	56
表 4.21 f_1 使用 AIS 方法實驗結果。	56
表 4.22 f_1 使用 AIS 方法實驗結果。	57

表 4.23 方程式 f_1 當 λ 不同時之數據。	57
表 4.24 f_1 利用歸屬函數之重疊性調整規則數前後之時間比較。	57
表 4.25 f_2 使用本論文方法實驗結果。	58
表 4.26 f_2 使用AIS方法實驗結果。	58
表 4.27 f_2 使用SymbAIS方法實驗結果。	59
表 4.28 方程式 f_2 當 λ 不同時之數據。	59
表 4.29 f_2 利用歸屬函數之重疊性調整規則數前後之時間比較。	59
表 4.30 各方法應用於三種函數結果對照表。	61



第一章 緒論

近年來，人工智慧成為電腦資訊科學中熱門的研究領域，人工智慧應用在非線性的近似最佳化求解、傳統非線性控制系統、函數模擬等方面，還有其它愈來愈廣泛的應用，使得愈來愈多的學者投入研究這塊領域，探究如何更有效率、更準確的讓結果收斂到最佳解附近。

人們不斷從生物系統中獲得靈感，提出了若干採用計算途徑實現的學習系統，包括基因演算法(genetic algorithm, GA)[1]-[3]、類神經網路(artificial neural network, ANN)、人工免疫系統(artificial immune system, AIS)等，它們分別從自然界演化過程、人體神經系統和生物免疫系統所發展而來的最佳化理論。基因演算法是依據優勝劣敗的機制，仿效基因遺傳的法則；即擇優(selection)，交配(crossover)、突變(mutation)，達到優良基因可永遠遺傳的目的，而劣等基因在世代中漸次被淘汰的原則，用在尋求近似最佳化的解題上。類神經網路更是仿人類神經系統的工作原理而發展出學習力強大的神經網路。生物免疫系統是一個高度進化的生物系統，它旨在區分外部有害抗原(antigen)和自身組織，進而清除病原並保持有機體的穩定。從計算的角度來看，生物系統是一個高度平行、分布、自適應的系統，具有很強的學習、記憶、判別和特徵抽取能力。人們自然希望從生物免疫系統的運行機制中獲得靈感，進而開發出應用層面的免疫系統計算機模型—人工免疫系統，來解決工程上的實際問題。目前，人工免疫系統已經發展成為人工智慧中一個嶄新的分支。

本章節中，將對本論文做整體架構介紹，1.1 節為介紹人工免疫系統的相關研究；1.2 節為研究動機；1.3 節會介紹本論文的整體架構，讀者可以藉此了解整篇論文的編排架構。

1.1 相關研究

本論文將以人工免疫系統(artificial immune systems, AIS)做為整篇論文主軸，本小節中將提及此演算法的起源與簡介。另外本論文另一重點為關聯性法則的尋找，將透過資料探勘找出最佳化過程中的抗體的關聯項目。

生物免疫系統所具有的多樣性、免疫記憶、平行式處理、自我組織、自我學習、自適性和強健性等特點，激發人們探索其運行原理，並建構人工系統來模擬生物免疫系統的特性。1958年，澳大利亞學者Burnet先提出了株落選擇原理(clonal selection theory)，1978年他本人又對該原理予以了更完整的闡述。株落選擇原理是免疫系統用來說明對抗原刺激所產生的免疫應答(immune response)基本特徵的演算法理論，只有識別抗原的細胞能夠進行克隆擴增(clonal expansion)。Hunt和Cooke[4]於1996年提出一個以抗體基因V-J 和V-D-J重組機制為主的人工免疫演算架構，並成功地將其演算法應用於DNA序列的識別上。1997年Dusgapta[5]說明神經網路和免疫網路兩者都是利用生物機制發展出來的技術，他們都具有學習、記憶與聯想的功能，所以常常被應用於解決辨識或分類的任務。但是辨識和學習是不同的機制，免疫網路提供了較不一樣的全域或區域的收斂的最佳解，免疫網路處理自組織的記憶並且記住本身的分類有較長的時效性。Fukuda 等人[6]於1999年提出一個兼顧多樣性、免疫網路理論與株落選擇的類免疫演算法，應用於解決多值域最佳化之問題上；作者更在其中導入了訊息熵(information entropy)的概念，用於抗體與抗體間的相似度判定上；其最主要之特徵為抗體之多樣性與有效地平行搜尋解答，並將多值域之解，存於系統之記憶細胞庫內。2000年，巴西學者de Castro 提出在人工免疫系統當中株落選擇演算法(clonal selection algorithm, CSA)[7]以及其學習和記憶的功能，最後再加入成熟(maturation) 的概念，並將該演算法應用於二元辨識、多模式最佳化與旅行推銷者問題。圖1.1為株落選擇原理示意圖，當抗原碰到某特定抗體時，抗體會大量增值。2009年Halavati和Shouraki[8]將共生合作的概念導入了人工免疫系統內，提出了共生式

人工免疫系統(symbiotic artificial immune system, SymbAIS)，並將演算法應用於函數最佳化、圖形識別…等。

近年來，人工免疫系統引起了很多學者的投入研究，最佳化過程為參考生物免疫系統，發展至今受到廣泛的應用，主要是因為人工免疫系統演算法彈性大、適用範圍廣且種類眾多，除了可用於函數最佳化、圖形識別、機器人學、自動控制...等，顯示出人工免疫系統強大的信息處理和問題求解能力以及廣闊的研究前景。

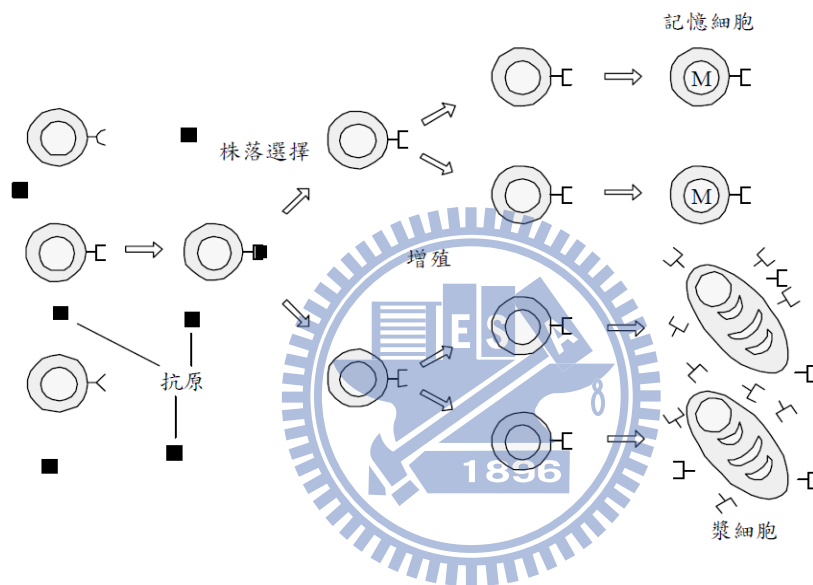


圖 1.1 株落選擇原理示意圖。

1.2 研究動機

基因演算法與人工免疫系統是目前常用的兩種最佳化理論，基因演算法因提出時間較早，因此最佳化的過程中較常以基因演算法做為工具，但近年人工免疫系統最佳化演算法發展迅速，許多改良後版本陸續被提出，這些改良方式使人工免疫系統演算法更具有競爭力，改良的種類中以共生式人工免疫系統所提出的抗體基元的概念最受青睞，它是將演算法中之解；亦為抗體，視為多個抗體基元組合而成，而利用這些抗體基元去進行演化以得到更佳的结果，也因此合作式人工

免疫系統更受到重視且廣泛的使用。

但共生式人工免疫系統所提出的合作式概念在面對所產生的抗體基元間相互有關連性時，演算法效能不佳，面對維度大的問題時，更是明顯，因此本論文希望透過關聯法則的搜尋，找出相關的維度項目，以做為共生式人工免疫系統中抗體基元分群的標準，因此透過這組訂立的規則，可幫助未來使用者在分群上有跡可循。

1.3 論文架構

以下為本篇論文章節的安排，第二章將介紹各類型人工免疫系統內容和演算方式，並將資料探勘做較詳細的介紹，包含資料探勘模式、流程、關聯法則以及建構頻繁樣式樹；第三章為本文核心，提供本篇論文的想法及架構；第四章為實驗數據，所有重要的資訊將詳列於此；第五章為本文的結論。



第二章 背景知識

本文中所提及相關知識，包含人工免疫系統、資料探勘以及模糊類神經網路三大領域。在最佳化演算法部分，使用人工免疫系統(AIS)；資料探勘的部分，利用頻繁樣式成長法(FP-Growth)尋找所需關聯性；而模糊類神經網路，本文使用 TSK 模糊類神經系統，此部分是用於函數模擬 (function approximation)，以下章節將詳細介紹各理論及其相關知識。2.1 節為生物免疫系統介紹；2.2 節為人工免疫系統的介紹；2.3 節為共生式人工免疫系統最佳化部分；2.4 節為資料探勘，介紹搜尋關聯法則的工具；2.5 節介紹 TSK 模糊類神經系統，本論文中函數模擬實驗需以此模糊類神經網路做為架構，在此節中將詳細介紹。

2.1 免疫系統

生物免疫系統具有多樣性、免疫記憶、平行式處理、自我組織、自我學習、自適性和強健性等優點，人工免疫系統的研究與發展是由生物免疫系統得到啟發，所以在本小節將先介紹生物免疫系統。

2.1.1 生物免疫系統

生物免疫系統主要目的是保護生物體免受外來感染性病原體的入侵[11]，當病原體入侵生物體時便會產生免疫反應(immune response)。在生物免疫系統之組成成分及相關反應關係如圖 2.1 所示，相互反應可分為兩階段，第一階段是抗原與非專一性吞噬細胞結合成的吞噬抗原對抗感染，屬於先天性免疫反應(innate responses)；第二階段則是淋巴細胞能專一性的辨識各種抗原進而消滅抗原，屬於適應性免疫反應(adaptive immune responses)。淋巴細胞又可分為 B 淋巴細胞(B lymphocytes)與 T 淋巴細胞(T lymphocytes)兩類，其中 B 淋巴細胞主要任務是製造抗體以消滅抗原，經辨認為專一性抗原時，B 細胞便開始增殖製造大量抗體與

抗原相結合，進而消滅抗原；T 淋巴細胞又可細分為兩種類型，第一類稱為輔助型 T 細胞(T-helper cells, TH)，能刺激 B 淋巴細胞發育與製造抗體，並協助吞噬細胞消滅吞入之病原體；第二類為毒殺性 T 細胞(T-cytotoxic cells, TC)，能辨認及消滅被病毒感染的細胞。

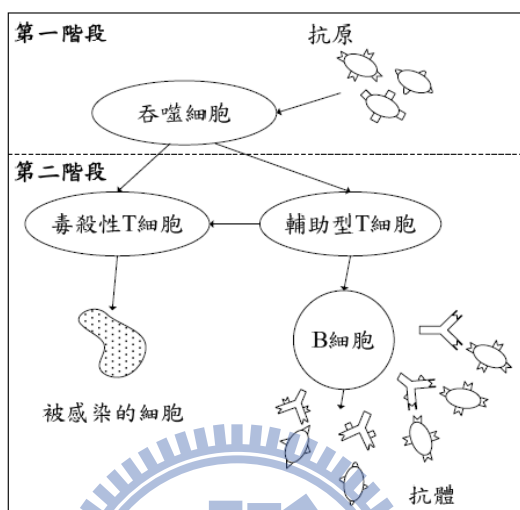


圖 2.1 生物免疫系統架構。

2.1.2 免疫機制

免疫系統是一個分散式、自組織和具有動態平衡能力的適應性複雜系統。它面對外界入侵的抗原，能夠產生相應的抗體，其抗體上有特定的物質，可以結合或消除抗原，以維持免疫平衡。其主要運行機制可歸納為以下三個方面：

(1) 免疫進化(immune evolution)：

免疫細胞識別抗原並啟動、分化與增殖，通過變異選擇和相似性抑制等過程逐步進化。一種抗原有多個不同型態，而一類抗體只有一種受體，因此一個抗原能夠和多個抗體結合。抗體與抗原之間的結合程度稱為親合度(affinity)，隨著抗原驅動的體細胞變異和對不同親合度抗體不斷的進行免疫反應，抗體對抗原的親合度也不斷增加一直達到親合度成熟。抗體並非各自獨立存在於生物體內，而是抗體之間相互結合，以抗體數量展現對抗抗原的能力，稱為抗體濃度

(concentration)。

(2) 免疫記憶(immune memory)：

抗原首次侵入生物體時引發初次反應，免疫系統通過學習抗原，產生優秀抗體並對有優良特性的抗體給予獎勵產生記憶細胞。當相同類型的抗原再次入侵時，觸發二次反應，通過喚醒記憶細胞，在更短的時間內產生大量抗體消滅抗原。這表現了進化過程的精英群體保留的策略，其作用如圖 2.2 所示。

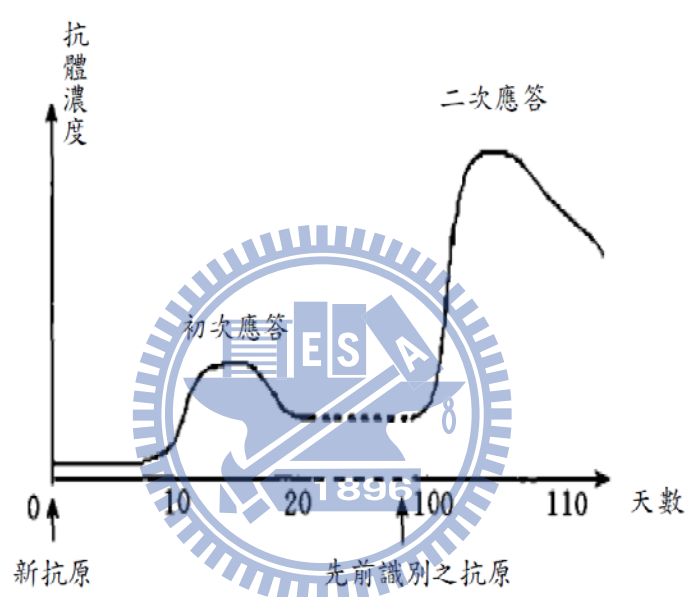


圖 2.2 免疫應答示意圖。

(3) 免疫調節(immune regulation)：

在免疫反應過程中，免疫系統內部各免疫細胞之間、抗原與抗體、抗體與抗體之間形成一個相互作用的動態平衡網路，當有外界抗原入侵時，透過免疫系統的調節，可達到新的免疫平衡。即使在無抗原入侵時，透過抗體之間的相互刺激與抑制反應，也能自我調節維持免疫平衡。即以上論述是免疫系統的獨特型網路假說(idiotypic network hypothesis)。由上述機制，如果把進化計算對應為整個免疫系統，把外來入侵的抗原對應為求解問題的目標函數，把免疫系統產生的抗體對應為問題的解，則經進化計算的求解過程與生物免疫系統的進化機制是十分類

似的。表 2.1 為生物免疫系統與實際應用問題名詞對照表。

表 2.1 生物免疫系統與實際應用問題名詞對照表。

免疫系統	實際應用問題
抗原(antigen)	問題
抗體(antibody)	解答
記憶細胞(memory cell)	過去的最佳解
親和力計算(affinity calculation)	計算出的最佳解
細胞複製(cell reproduction)	問題解自我複製
株落選擇(clonal selection)	交配、突變產生新解答

2.2 人工免疫系統

一般的工程最佳化設計問題都很複雜，很難用人力去達成運算的工作，所以利用計算機來處理運算的工作是必須的。在處理問題時，都要先瞭解問題並分析問題。分析問題是要將問題的輸出、輸入清楚地定義出來，有了輸出輸入才有辦法針對不同的目標值作處理。分析完問題之後，就需要一個有效的演算法快速地求得問題的最佳解，本節將介紹人工免疫系統用來搜尋最佳設計解。圖2.3為人工免疫系統流程圖。

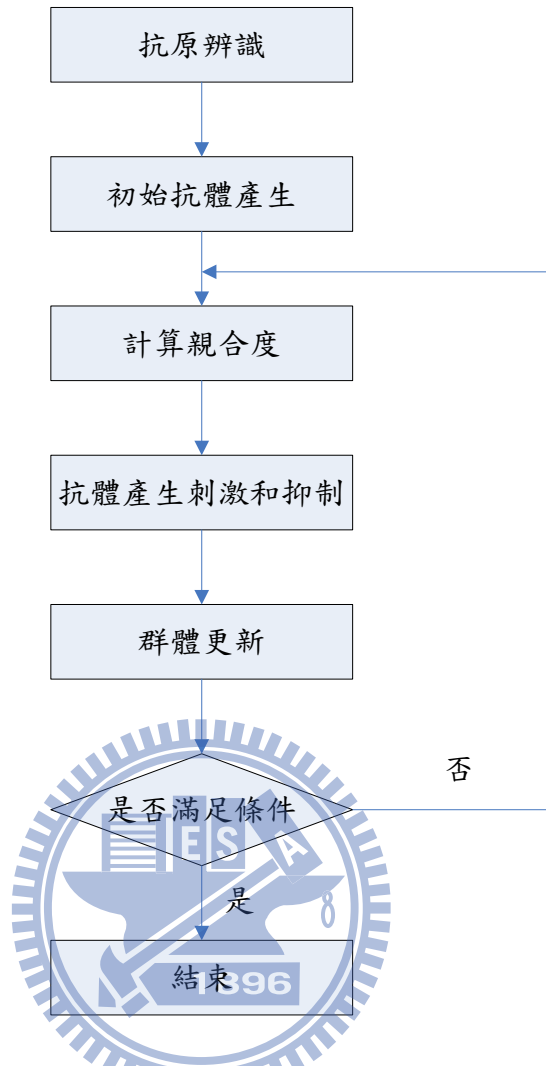


圖 2.3 人工免疫系統流程圖。

(1) 抗原辨識：

輸入目標函數(抗原)和各種約束作為人工免疫網路演算法則的抗原。

(2) 初始抗體群產生：

刺激記憶細胞產生初始抗體群，清除過去出現過的抗原，並從包含較佳抗體(較佳解)的數據中選取一些抗體。初始通常為隨機產生，編碼方式主要有二進制編碼(binary code)、實數編碼(real code)、灰色編碼(gray code)等。

(3) 計算親合度(affinity)：

抗體之間和抗體對抗原之間的相互作用是利用親合度來描述，能與抗原有很好結合性的抗體會具有較高的親合度，抗體之間的親合度越大表示特性越相似。其中在抗體 v 與抗體 w 之間的親合度計算，用 $E(2)$ 表示抗體 v 與抗體 w 之間的訊息熵，當 $E(2)=0$ 時表示抗體 v 與抗體 w 之間的基因相同，其定義如(2.1)式：

$$(A_b)_{vw} = \frac{1}{1+E(2)}, \quad (2.1)$$

在抗體 v 與抗原之間的親合度定義如(2.2)式：

$$(A_g)_v = \frac{1}{1+t_v}, \quad (2.2)$$

其中 $(A_g)_v$ 是抗原和抗體 v 之間的親合度， t_v 是抗原和抗體 v 之間的抗體濃度。當 $(A_g)_v=1$ 時表示抗體與抗原完全匹配，即得到最佳解。

計算每個抗體對抗原之間的親合度和抗體之間的親合度。在抗體群中抗體具有多樣性，可用平均訊息熵(entropy)來計算，假設免疫系統由 N 個抗體，每個抗體有 M 個基因組成，如圖 2.4 所示。每個抗體的每一個基因 g_{ij} 共有 S 個字母可選擇： k_1, k_2, \dots, k_s (對二進制編碼來說， $S=2$ ，即使用 0、1 兩種字母)。則 N 個抗體的平均訊息熵可計算為 $E(N)$ ，如(2.3)式：

$$E(N) = \frac{1}{M} \sum_{j=1}^M E_j(N), \quad (2.3)$$

其中 $E_j(N)$ 為 N 個抗體計算第 j 個基因的訊息熵，如(2.4)式：

$$E_j(N) = -\frac{1}{M} \sum_{i=1}^S P_{ij} \log P_{ij}, \quad (2.4)$$

其中 P_{ij} 是字母 k 出現在 N 個抗體第 j 個基因的機率。當位置 j 上的 N 個

抗體的基因都相同時，則 $E_j(N) = 0$ 。

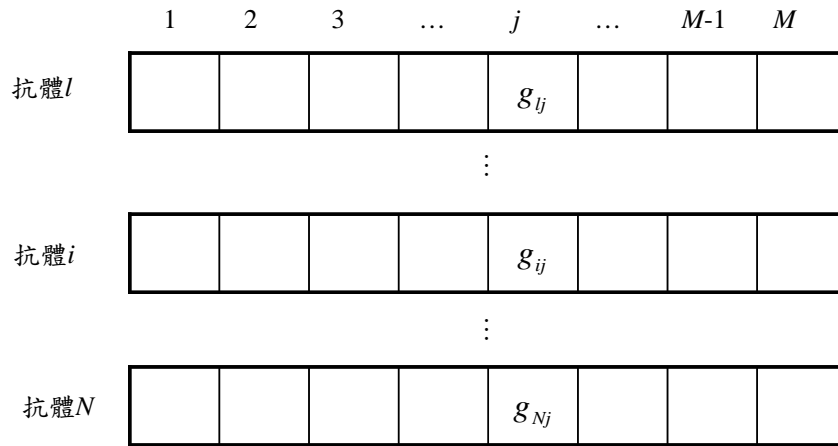


圖 2.4 抗體群架構圖。

抗體濃度表示相同抗體在整體抗體群中所佔的比例，抗體 i 的濃度 C_i 可定義為如(2.5)式：

$$C_i = \frac{1}{N} \sum_{w=1}^N (A_b)_{iw}, \quad (2.5)$$

$$(A_b)_{iw} = \begin{cases} 1, & (A_b)_i \geq \theta, \\ 0, & \text{其他,} \end{cases} \quad (2.6)$$

其中， N 為抗體總數， θ 為常數，一般取 $0.8 \leq \theta \leq 1$ 。

(4) 抗體刺激與抑制：

依據抗體濃度調整抗體，如(2.5)式。高親合度的抗體受到刺激，當抗體濃度高到一定的值時就要受到抑制，反之相對是提高濃度低的抗體產生選擇機率。

(5) 群體更新：

對未知的抗原響應產生抗體更新。

2.3 共生式人工免疫系統

共生式人工免疫演算法為 Halavati[10]於 2009 年基於原始人工免疫演算法架構所提出，跟原始人工免疫演算法整體上並無太大差異，僅在共生式人工免疫演算法提出了抗體基元的概念，類似於基因演算法中合作式的想法[12][13]。

原始人工免疫演算法過程中是將一個抗體視為一個完整的解，在演化尋找最佳解的過程中，每個抗體與抗體之間是為競爭的關係。例如，一個抗體同時代表解的兩種特徵，而利用這種抗體表示法，當抗體親和力好的部分被親和力差的部分環繞時，那整個抗體會呈現低親和力，也代表會有較小的機率被選擇、複製，結果使得對演化有利的部分慢慢消失。而共生式人工免疫演算法提出的概念就是將一個抗體視為多個抗體基元組合而成，將每種特徵分出來，然後再個別去進行演化。圖 2.5、圖 2.6 為完整的共生式人工免疫系統流程圖。

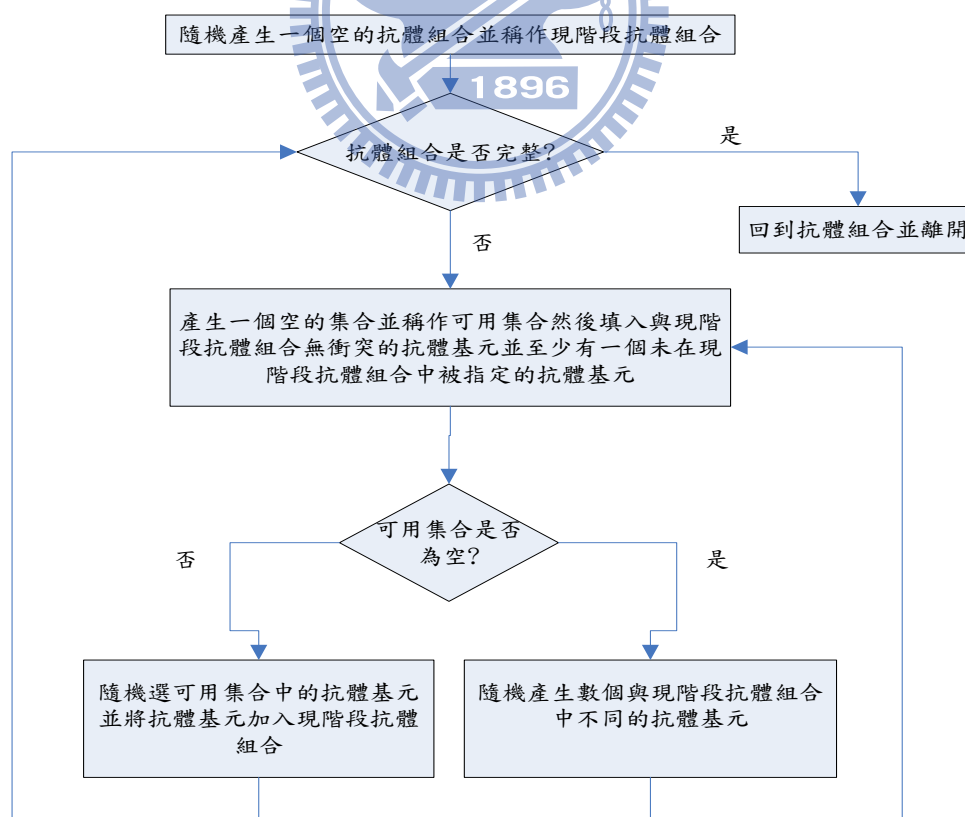


圖 2.5 產生抗體集合演算法。

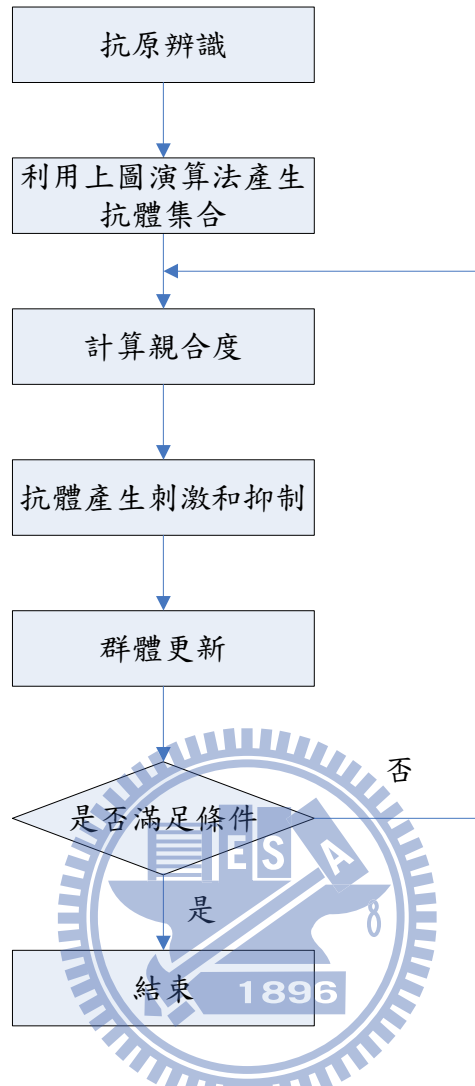


圖 2.6 共生式人工免疫系統流程圖。

2.4 資料探勘

資料探勘(data mining)是用來擷取所需資訊一種有效的方法[14],這些資訊藏於大量的資料庫中,透過資料探勘可找出使用者所需要且有價值的潛藏規則。資料探勘可視為資料庫知識發掘(knowledge discovery in database)的一部分,其中知識發掘的過程包括資料選取、前置處理、轉換、資料探勘及解釋與評估。若資料庫本身具有足夠的完整性,則所得到的結果將會是重要的知識。資料探勘運用分類、群集分析、關連性、序列分析、機器自我學習及其它統計方法,從資料庫中找出隱藏、未知且有用的資訊。

資料探勘可分成五種模式，分類(classification)、趨勢分析(trend analysis)、分群(clustering)、關聯(association)及循序特徵(sequence pattern)。

1. 分類：根據一些變數的數值做計算，再依照結果作分類。例如可將全部垃圾資料分為可燃不可燃，以這些完成分類的資料，研究可燃垃圾與不可燃垃圾的性質和特徵，這就是從歷史資料中所得到的知識，再利用這些擷取到的知識，去對未來的資料做預測。
2. 趨勢分析：以現今所得到的資訊，來預測未來的數值。例如工廠利用去年的訂單和員工請假頻率，來衡量今年需要雇請多少員工，或是將去年與今年員工數做比較，找出其中的差異性。
3. 分群：是將資料分組，其目的是找出各組之間的差異及同組中成員的相似性，使群組內差異小，群組外差異大。例如百貨公司將商品以使用客戶年齡層、性別、消費能力，分不同樓層區塊擺放，這種分群可使百貨公司更易於統計客戶的需求，也增加客戶的便利性。
4. 關聯：是要在同一個交易中找出隱含其它項目之存在的項目，找出某一事件或是資料中會同時出現的部分。例如客人買了餅乾和糖果，則在這情況下，多數客人喜歡也買汽水，透過這種關聯性，商家將糖果、餅乾及汽水，放置在同一個區塊，且只針對餅乾和糖果做特價，則可達到將三種商品一起賣出的效果，達到銷售目的。
5. 循序特徵：與關聯關係密切，最大的不同是循序特徵中相關項目是以時間加以區分。例如客人買了餅乾和糖果，在這種情況下，兩天後客人會再來買汽水的機率。

資料探勘流程圖如圖 2.7 所式，各步驟解釋如下：

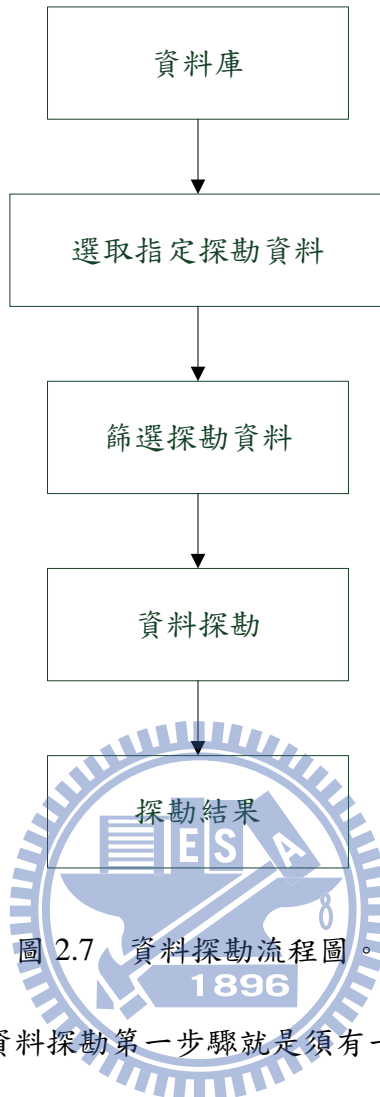


圖 2.7 資料探勘流程圖。

1. 建立資料庫：要執行資料探勘第一步驟就是須有一個完整且大量的資料庫，這個資料庫中未必所有項目都是使用者所需要，也未必每一筆資料都無瑕疵，因此在使用前應先做前處理。
2. 選取指定探勘資料：資料庫並未必僅依照使用者當下需求所建立，因此許多種類的資訊是不必要的，探勘的資料可能不會是整個資料庫的全部，需要的僅是其中一個資料庫的一部份，或是數個資料庫裡的一部份資料。
3. 篩選探勘資料：篩選出較好的資料，刪除一些不必要的內容，例如資料重複、資料錯誤、或資料無效…等，目的使探勘過程減少計算量以及確保資料的完整性。
4. 資料探勘：種類如上述資料探勘五種模式，分類、趨勢分析、分群、關聯及循序特徵等。

5. 探勘結果分析：藉由資料探勘，找出隱藏的特徵或關聯性。

資料探勘憑著可預測未來的趨勢，可對使用者提出有用的建議，但唯有好的資料庫及前置處理，才是資料探勘最重要的一步。

2.4.1 發掘關聯式規則

發掘關聯式規則 (mining association rule) 最早是 1993 年由 Agrawal[9][10] 等人所提出，主要用來找出項目間的關聯性，例如用於賣場商品販售上，可找出各商品間販賣的關係，像是買牙刷的人有多少機率會買牙膏、買洗髮精的人有多少機率會買潤髮乳等。文獻[15]-[18]提及關聯法則於市場應用的重要性。

支持度(support)和信任度(confidence)是發掘關聯式規則最重要的兩個指標，若以賣場為例，兩者表示如(2.7)式、(2.8)式，支持度($X \rightarrow Y$)表示同時購買商品 X 跟 Y 的機率，而信任度($X \rightarrow Y$)代表當顧客購買商品 X 時，同時會購買商品 Y 的機率，由這兩種指標可以了解顧客的購買習慣，可成為賣場商品做促銷時所需重要資訊。

使用上，首先須設定最小支持度(minimum support)以及最小信任度的門檻(minimum confidence)，利用這兩個指標為使用者做篩選。發掘關聯式規則有兩個步驟：步驟一，找出所有頻繁項目。步驟二，由步驟一所得到的頻繁項目，產生高關聯性的規則。

$$\text{支持度} = s_{X \rightarrow Y} = (\text{同時包含 } X \text{ 跟 } Y \text{ 的銷售紀錄}) / (\text{總交易紀錄}). \quad (2.7)$$

$$\text{信任度} = c_{X \rightarrow Y} = (\text{同時包含 } X \text{ 跟 } Y \text{ 的銷售紀錄}) / (\text{包含 } X \text{ 的銷售紀錄}) \quad (2.8)$$

如圖 2.8 所示，為一簡單範例，交易項目包含{A, B, C, D, E, F}六項，而超過最小支持度的項目總共有{A}、{D}、{E}、{F}、{A, D}、{A, E}、{D, E}及{A, D,

E}等7項。

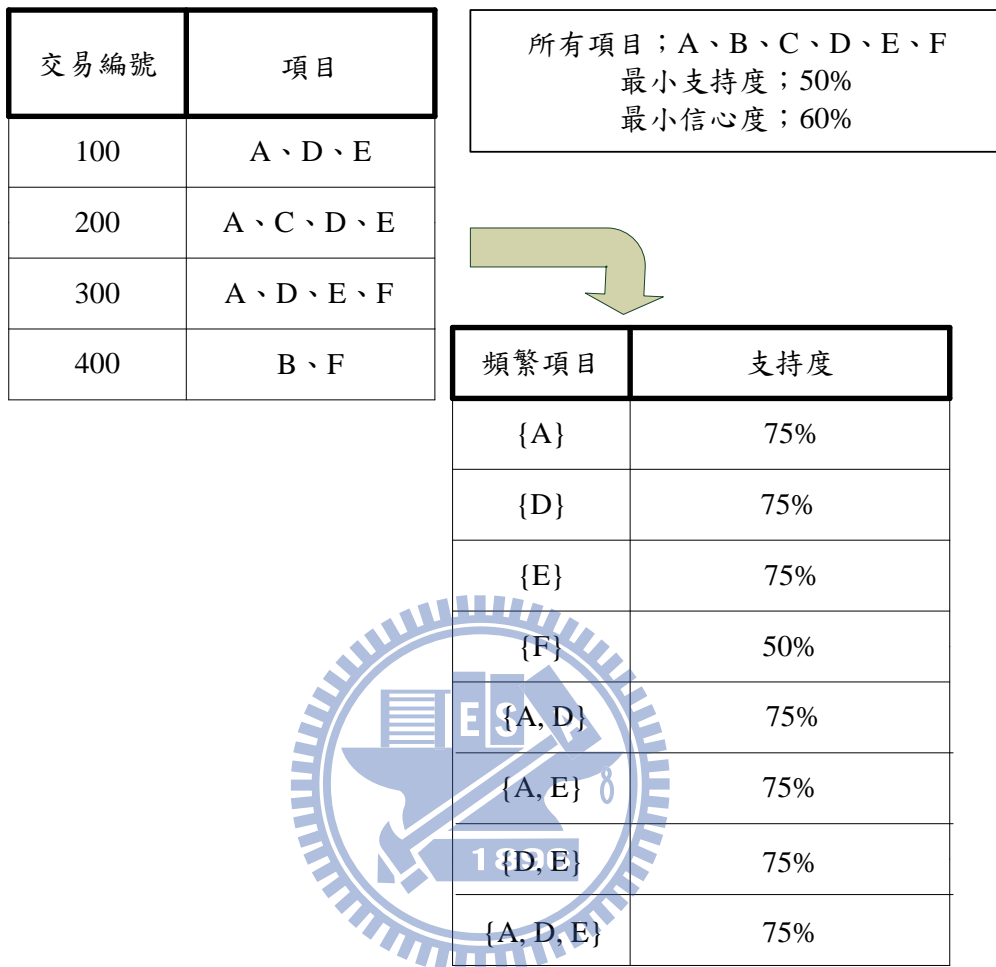


圖 2.8 發掘關聯式規則簡例。

經由(2.8)式，可算出各種規則組合的信任度，信任度可視為關聯的強度，如信任度 $(A \rightarrow D) = \text{支持度}(\{A\} \cup \{D\}) / \text{支持度}(\{A\}) = 100\%$ ，以商品為例，表示購買商品A的情況下同時會購買商品D的機率是100%，符合使用者一開始所設定的60%；又如信任度 $(A \rightarrow F) = \text{支持度}(\{A\} \cup \{F\}) / \text{支持度}(\{A\}) = 33\%$ ，信任度小於最小信任度，所以這個關聯式規則並不是使用者所需要的。

2.4.2 頻繁樣式成長演算法

有別於 Apriori 演算法，以逐層搜尋的方式找出關聯規則，頻繁樣式成長法 (FP-growth) 並無延用此一方式，整體核心為頻繁項樹 (FP-tree) 的建構，這也是效能可以優於 Apriori 演算法的主要原因。頻繁樣式成長法在一開始做一次全面的搜尋，先去除小於最小支持度的項目，再將各筆資料內的項目依照支持度的高低進行排序，依此資料建立頻繁項樹。

最小支持度：60%

交易編號	交易項目	交易編號	排序後指定項目
100	{A, C, D, F, G, I, M, P}	100	{A, F, C, M, P}
200	{A, B, C, F, I, M, O}	200	{A, F, C, B, M}
300	{B, F, H, J, O}	300	{F, B}
400	{A, B, C, K, S, P}	400	{A, C, B, P}
500	{A, C, E, F, L, M, N, P}	500	{A, F, C, M, P}

圖 2.9 資料指定並排序。

如圖 2.10，資料庫中有五筆交易資料，最小支持度為 60%，所以當資料內項目出現少於三次則會在第一次挑選中被刪除，挑選後的項目再依照出現的次數作排序，即為圖 2.10 中左邊的表格，上步驟結束後，建立標頭索引表 (header table)，索引表的建立僅依存在項目出現的次數加以排序，目的是在找出頻繁項樹中每個項目的第一個節點，由圖 2.10 左表及標頭索引表，建立頻繁項樹，且每個相同項目以虛線連接，如圖 2.10 所示。

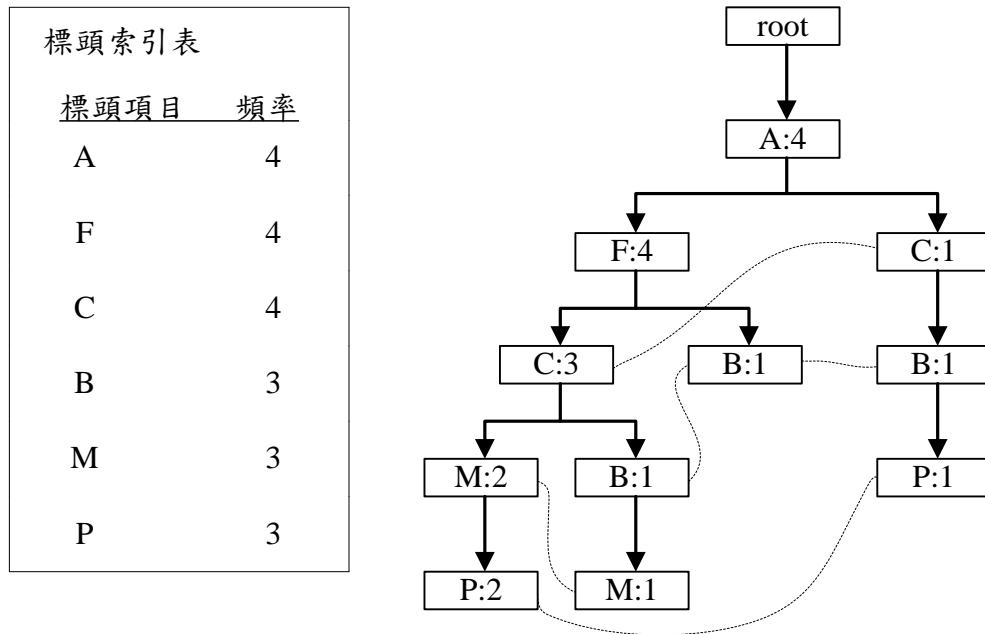


圖 2.10 建構頻繁項樹。

建立出頻繁項樹後，可藉由此得到想要的關聯資訊，例如想知道與項目 P 相關的關聯規則，可得到兩條路徑，路徑一與出現次數為 AFCM : 2，另一條為 ACB : 1，由此兩條路徑，得知與 P 關聯的各項目出現次數分別為 {A:2, F:2, C:3, M:2, B:1}，最小支持度為 60%，及最少須出現 3 次，可得出與 P 相關的關聯規則為 CP:3。

頻繁樣式成長法與資料探勘中另一方法 Apriori 演算法相比較，頻繁樣式成長法有幾項優點：

1. 不須產生候選項目，因此不需做候選項目測試。
2. 根據頻繁項目的提供來分解探勘目標以及資料庫。
3. 可縮小探勘時的資料庫容量。
4. 不須重複搜尋整個資料庫。

故本論文選擇頻率樣式成長法來分群抗體基元。

2.5 模糊類神經網路

模糊系統目前已經廣泛的應用在自動控制系統以及信號處理等方面，尤其近年來對於控制器的設計有著十分重要的貢獻，模糊系統已經慢慢有成為設計控制器的理想工具的趨勢。由於模糊系統可應用的範圍非常的廣泛，因此有時亦稱之為模糊專家系統(fuzzy expert system)。其模仿人類對於事物描述方式具有模糊性的觀念，因此模糊系統的優點即是不需要完整精確的數學模型。

另一方面其結合人類的知識於系統的設計上。而模糊化的好處是可以提供較好的推廣性、錯誤容忍度、以及更適合應用於真實世界中各種非線性系統。以下章節 2.5.1 將介紹模糊系統的架構；2.5.2 則說明模糊系統結合類神經網路來建構模糊化類神經網路。

2.5.1 模糊系統

首先我們討論模糊模型對於信息的陳述方法依照模糊規則的不同概略可以分為兩種的類型；第一類模型是由學者 Mamdani[20]所發展出來，一般稱之為 linguistic 模型，其模糊規則是屬於語意式的模糊規則(linguistic fuzzy rule)，其數學模型可以表示為：

$$r^\ell : \text{If } x_1 \text{ is } A_1^\ell \text{ and } x_2 \text{ is } A_2^\ell, \dots, \text{ and } x_m \text{ is } A_m^\ell, \text{ then } y^\ell = c^\ell, \quad (2.9)$$

此處 $r^\ell (\ell = 1, 2, \dots, R)$ 代表第 ℓ 個模糊規則， $A_1^\ell, A_2^\ell, \dots, A_m^\ell (\ell = 1, 2, \dots, R)$ 為歸屬函數(membership functions)，歸屬函數屬於模糊化機構方塊內，模糊化機構的功能就是將明確的資料轉換為適當的模糊範圍來表示，而此機構會因為模糊系統功用的不同，而使用各種不同的模糊化歸屬函數，通常以高斯式歸屬函數、梯形歸屬函數或三角型歸屬函數表示，當然還有其它各種可以妥善表達分布的歸屬函數。而 $X = [x_1, x_2, \dots, x_n]$ 為輸入變數， y^ℓ 表示第 ℓ 條模糊規則的輸出變數。

第二類模型是由學者 Takagi 以及 Sugeno[21]所提出，一般稱之為 TSK 模型。它可以描述一般動或靜態的非線性系統，利用模糊化分割(fuzzy partition)將輸入空間視為線性分割的拓展，其模糊規則屬於函數式模糊規則，可描述為：

$$r^\ell : \text{If } x_1 \text{ is } A_1^\ell \text{ and } x_2 \text{ is } A_2^\ell, \dots, \text{ and } x_n \text{ is } A_n^\ell, \text{ then } y^\ell = a_0^\ell + a_1^\ell x_1 + \dots + a_n^\ell x_n, \quad (2.10)$$

其中 y^ℓ 表示第 ℓ 條模糊規則的輸出變數， $a_i^\ell (i=0, 1, \dots, n)$ 為實數參數。此種模型可處理非線性系統，其做法是將整個輸入空間切割為數個可用線性系統描述的模糊空間，而輸出空間再以一個線性方程式表示。本篇論文我們將會使用 TSK 模型來做類神經網路。

2.5.2 模糊化類神經網路

圖 2.12 簡單的描述了一個以多層的模糊化類神經網路來建構模糊化類神經模型，此模型有 n 個輸入、一個輸出，並擁有五層的架構，每一層的模糊化類神經元都做相同的運算。第一層為輸入層，直接將輸入的信號傳送至下一層，這一層中每一個相連接的權值皆為一，並且不會被調整。第五層為輸出層。網路第三層稱之為規則層(rule layer)，其為 n 個輸入值，構成模糊規則的基底。整個模型中每一層的運算元的功能及詳細運算我們將會說明於下。

第一層：第一層的類神經元並不作任何的運算，只是直接將輸入信號傳送至下一層。連接輸入信號與類神經元的權值在運算中也不會被調整，將一直保持為 1；其計算公式如下：

$$u_i^{(1)} = x_i. \quad (2.11)$$

第二層：第二層的類神經元，執行輸入值與模型的相關模糊集合的相容程度運算，其計算如下：

$$u_{ij}^{(2)} = s_i^j(u_i^{(1)}), \text{ for } i = 1, 2, \dots, n; j = 1, 2, \dots, R, \quad (2.12)$$

其中 $u_i^{(1)}$ 表示第 i 個輸入信號、 $u_{ij}^{(2)}$ 表示第 i 個輸入信號在模糊規則為 j 時類神經元的輸出信號、而 $s_i^j(u_i^{(1)})$ 表示模糊化機構，它的功能是将明確的資料模糊化成為模糊的資訊，我們可將它視為一種映射關係，由明確的輸入空間映射至模糊集合空間。模糊化機構會視各種不同的應用而使用不同的歸屬函數表示式，在本篇論文中我們將選擇以高斯式歸屬函數作為模糊化機構，其數學式如下：

$$s_i^j(u_i^{(1)}) = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right), \quad (2.13)$$

其中 m 表示高斯函數的中心點 (mean)、 σ 為寬度 (deviation) 並以 σ 控制歸屬函數遞減的速率。這種模糊化方式較其他方法(三角形函數、梯形函數)複雜，所需的計算量也比較大，但是此種模糊化方式的優點是，當輸入資料易被雜訊干擾時，還能有效的消除由雜訊引起的錯誤，即有比較好的容錯能力。

第三層：第三層的類神經元執行模糊規則啟動強度(firing strength)的運算，每一個類神經元有 m 個由第二層輸出而來的輸入值，其運算數學式如下表示：

$$u_j^{(3)} = \prod_i u_{ij}^{(2)}, \quad (2.14)$$

這邊第三層類神經元的運算就相當於模糊理論中所提到的模糊交集運算 (fuzzy AND operation)。

第四層：這一層又稱為推論層，在這一層中主要是將第三層的輸出值乘上一

個由輸入參數所組成的線性組合而得到推論結果，這一層的輸出說明如下：

$$u_j^{(4)} = u_j^{(3)} (w_{0j} + \sum_{i=1}^n w_{ij} x_i), \quad (2.15)$$

其中 w_{ij} 代表輸入參數所組成的線性組合中的權重。

第五層：在這一層中主要是做一個去模糊化的動作，透過推論的結果將模糊值轉換成一個明確的結果，去模糊化的動作說明如下：

$$y = u^{(5)} = \frac{\sum_{j=1}^R u_j^{(4)}}{\sum_{j=1}^R u_j^{(3)}} = \frac{\sum_{j=1}^R u_j^{(3)} (w_{0j} + \sum_{i=1}^n w_{ij} x_i)}{\sum_{j=1}^R u_j^{(3)}} \quad (2.16)$$

其中 R 是規則數， n 是輸入參數的維度。

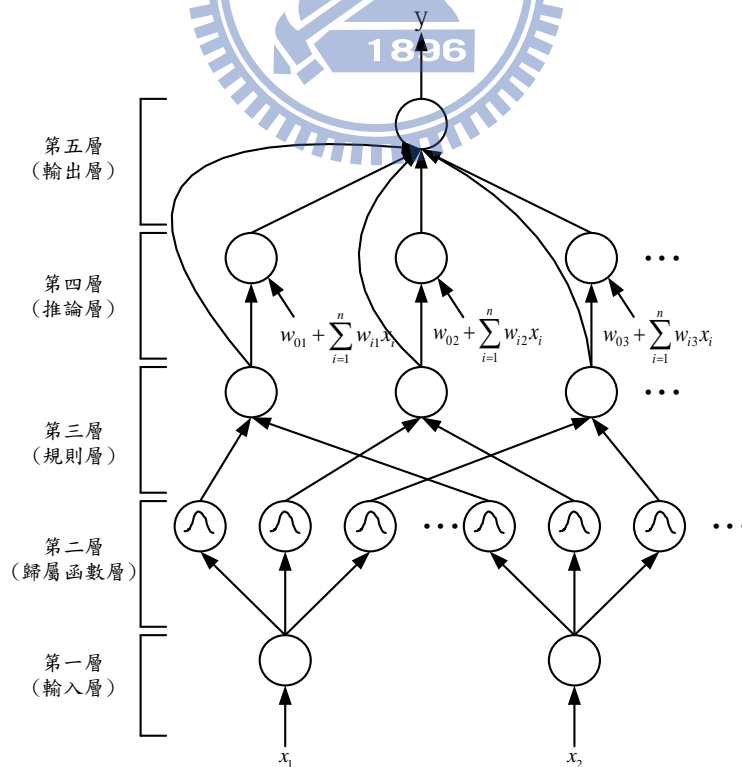


圖 2.11 TSK 形式模糊類神經網路的架構。

第三章 基於頻繁樣式成長法的共生式人工免疫系統

本論文主要為函數模擬，在尋找函數最佳解上，主要為改進共生式人工免疫演算法，以資料探勘尋找所面對問題各抗體基元之間的關聯性，做為解分類的依據；在函數模擬上，以 TSK 模糊類神經系統做為主要核心，再以改良過的基於頻繁樣式成長的共生式人工免疫系統找到最佳化的系統參數，使模擬出的資料能夠更接近真實函數的圖形。

使用基於頻繁樣式成長共生式人工免疫系統的原因，是因為原始人工免疫系統與共生式人工免疫系統皆有其適合解決的問題，但也有各自在最佳化上的盲點，且仔細觀察這兩種方法，原始人工免疫系統將所有抗體放在同一個族群裡，共生式人工免疫系統則讓所有抗體基元分到不同的族群，而頻繁樣式成長法為基礎的共生式人工免疫系統則是基於面對問題的不同來調整抗體基元所應分到的族群。

章節 3.1 為本文演算法概述；章節 3.2 為基於頻繁樣式成長法成長的共生式人工免疫演算法，此處為本文訂立抗體基元的架構。

3.1 演算法概述

現今用於最佳化處理的演算法多為基因演算法(GA)，而人工免疫演算法為近幾年所發表的最佳化演算法，在表現上並不遜色於基因演算法，使用上也逐漸嶄露頭角，許多改良型人工免疫最佳化的演算法也陸續出現，其中值得注意的方法為共生式人工免疫系統，共生式人工免疫系統是一套應用廣泛的方法，但此演算法最大的缺點為當碰到抗體基元間有關聯性時，由於共生式人工免疫系統分群的方法會使演算法性能下降，意即使用者必須嘗試多種抗體基元分群方式，然後

單純比較之間的效果，才能找出判斷哪一種分群效果是好的，這樣的使用方式，導致在面對抗體基元有關聯性或者為中、高維度的問題時遇到困難。

本文在最佳化的處理上，同樣使用共生式人工免疫系統做為演算方法，對於無族群分類標準的問題，透過資料探勘來解決並告知使用者哪些抗體基元維度彼此間具有關聯性，圖 3.1 為最佳化處理時的基本流程，面對複雜函數時，先以資料探勘尋找出各個抗體基元關聯性，決定哪些抗體基元該放在同一族群，哪些該分開，再依照這個族群分類結果提供共生式人工免疫系統使用，以達到最好的表現。

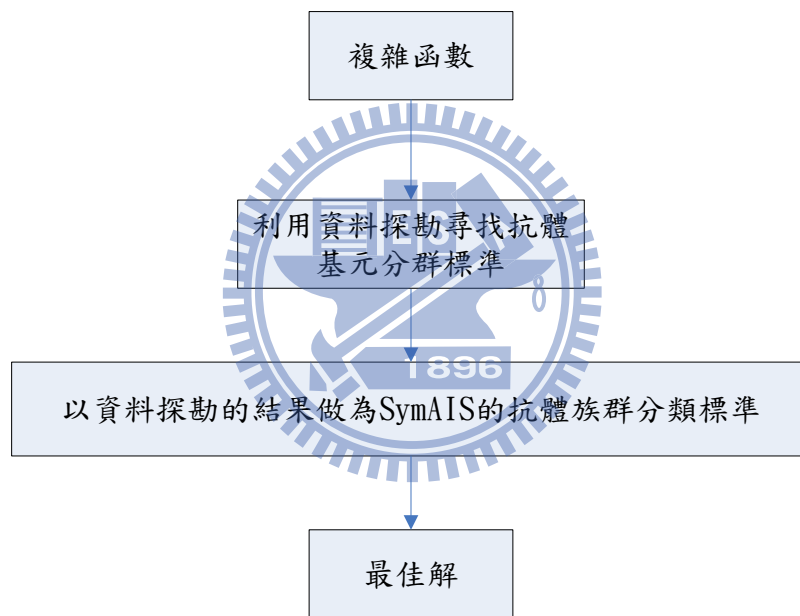


圖 3.1 基本流程。

3.2 基於頻繁樣式成長的共生式人工免疫系統

此章節目的在於訂立共生式人工免疫系統之抗體基元分類規則，以抗體基元間的關聯性做為是否分在同一類的標準，在族群分類上，以頻繁樣式成長法做為搜尋相關性項目的演算法，藉由此方法找出抗體基元之間的關聯程度，以確立分類準則。

3.2.1 抗體基元分類規則確立

本論文與共生式人工免疫演算法最大的差別，就在於加入了資料探勘的技巧，使選取抗體基元時在相互之間有關聯性時可以被分在同一群再進行演化，以達到增進演算法效率的目的。在資料探勘的部分，以頻繁樣式成長法做為抗體基元找尋關聯性的工具，在使用上首先設定數筆測試資料，再以測試資料的表現為依據定立規則，可分為四個步驟，隨機分組、資料測試、資料選取以及規則確立，說明如下：

1. 隨機分組：當面對高維度抗體族群的分類方式，這也是使用上的第一個步驟，但可惜的是，大多數問題並無法察覺該如何將維度做區分，也無法知道分成幾個族群是最佳的方式，也就是說，如果一開始便決定每一筆測試資料的抗體基元個數以及族群內維度個數都是相同的，當面對抗體基元間有關聯性時，這樣的設定並不妥當。如圖 3.2 所示，此為一符合實際情況的分類方式，假設問題具有三十個維度，若此時處於最好的分類情況，會發現每個族群內的維度個數並不會相同，例如 *swarm_1* 包含維度編號{1, 9, 11, 13}共四個維度，*swarm_2* 只有維度編號{10}一個維度，而 *swarm_4* 卻含有維度編號{6, 17, 18, 20, 25, 29, 30}共 7 個維度，除此之外，使用者也無法得知族群個數應如何設定才是最佳解，因此本文在族群個數以及各族群內所包含的維度數目和編號皆以隨機產生，且當一筆測試資料完成後，重新設定族群個數以及各族群內維度數目和編號，增加彼此間配對組合的多樣性。

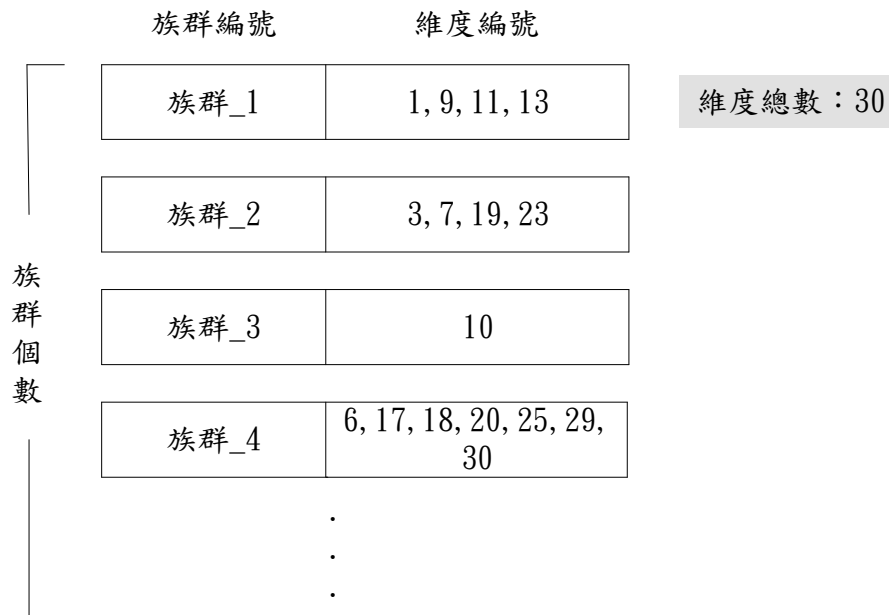


圖 3.2 實際分類情況。

本文中隨機分組程序將會得到如同圖 3.3 的結果，圖中僅列出三筆測試資料，最適當的測試資料數目需視維度的大小而定，數目取太少，造成配對組合方式不足，這樣的結果意義並不大；反之數目取太多，雖然可更精確得到維度之間彼此的關聯性，但卻會造成資料量龐大，演算時間過長的窘境。此種方式得到的各種配對組合，比較圖 3.2 可發現，是較符合實際情況的分類。

2. 資料測試：將隨機分組後所得到的資料，做為初步族群分組的依據，以這幾種分組情況進行測試，每一筆測試資料經過運算會對應到一筆適切值(fitness value)，這些適切值將構成初步資料庫。

維度總數：30

		第一筆		第二筆		第三筆	
		族群編號	維度編號	族群編號	維度編號	族群編號	維度編號
族群 個 數	族群_1	2, 5, 8	族群_1	3	族群_1	21, 25, 30	
	族群_2	3, 17	族群_2	9, 19, 26, 30	族群_2	3, 19, 27	
	族群_3	14, 26	族群_3	8, 10, 23	族群_3	5	
		⋮		⋮		⋮	
		⋮		⋮		⋮	
		⋮		⋮		⋮	
	族群_13	9, 16, 25, 30	族群_21	1, 17, 18, 29	族群_9	6, 28, 29	

圖 3.3 本文測試資料分類情況。

3. 資料選取：初步資料庫建立後，此步驟主要用來判別資料庫裡各筆資料的好壞，適切值較好的資料，代表分類情況較佳，反之適切值較差的資料，表示當時的分類情況，對最佳化的過程來說是不良的。判別好壞的方式本文以各筆資料適切值的百分之四十最為標準，意即分布於 40% 的適切值視為優良，匯入優良的資料成為優良資料庫，且假設一筆資料視為優良，則這筆資料內所有的配對組合都將視為優良，成為未確定規則，其餘資料將視為表現較差的分類情況，刪除這些不必要的資訊，初始資料庫中這些表現較差的分類情況，將不在優良資料庫中出現。資料庫中每一筆資料的型態，將會告知哪些組合是優良的，例如{維度_2, 維度_5, 維度_8}一組為優良、{維度_3, 維度_17}一組為優良、{維度_9, 維度_16, 維度_25, 維度_30}一組為優良...，這些組合變是未確立規則，需要再透過規則確立的步驟檢驗。

4. 規則確立：得到優良資料庫後，將以這個資料庫裡的各筆資料產生一組或多組規則，以頻繁樣式成長法做為尋找規則的方法，首先將優良資料庫裡的資料建立頻率項樹，由頻繁項樹可找到維度間的關聯性，其中關聯法則兩個重要的指標支持度和信任度，在本文中所設定最小支持度為整體資料個數的50%，最小信任度亦為50%，當完成頻繁項樹後，透過關聯法則可以知道哪些維度分於同一族群裡，可滿足最小支持度及最小信任度，這些組合方式，將成為後續運算上的分群組規則，意即共生式人工免疫系統族群分類規則因此確立。

圖 3.4 為族群分類規則確立的詳細流程圖，族群分類規則訂立後，有可能會產生失敗的情況，若訂立規則後演算過程中，所得到的適切值不滿足優良資料庫的優良標準，則視為規則失敗，將重新訂立族群分類規則，造成規則失敗的原因有兩個：

1. 測試資料錯誤：測試資料在進行演化時，可能陷入局部最佳解，以至於這樣的資料庫並不正確，影響族群分類情況。
2. 測試資料數目過少：測試資料數目太少導致資料庫裡的各筆資料並不完整，此種情況下容易造成族群分類情況產生不客觀的效果。

而如何判斷規則數多少為適當，則是利用模糊系統中模糊集合歸屬函數重疊性(overlapping)是否過高來判斷，若重疊性過高則代表重複的規則數(redundancy)太多。

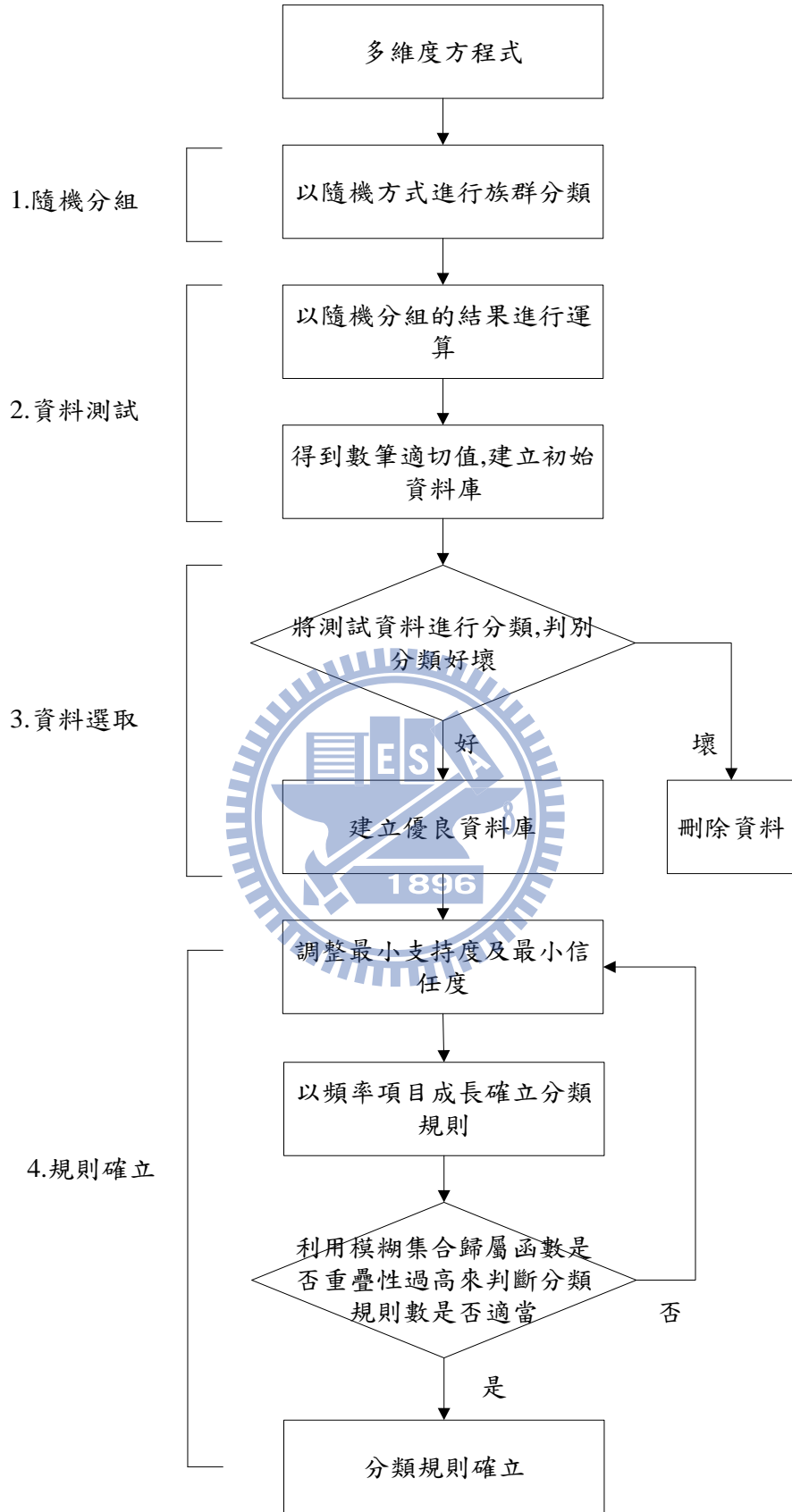


圖 3.4 族群分類詳細流程。

圖 3.5 和圖 3.6 為抗體基元分組簡例，其中包含維度_1 到維度_17 共 17 個維度，在優良資料庫中存放五筆交易資料，圖 3.5 為這五筆交易資料，經過最小支持度刪除頻率低項目後，再依照頻率的高低由高至低做排序所得的結果，此處最小支持度為 50%，因為資料個數為五筆，所以各項目必須出現至少 3 次才會被選取，其餘出現少於 3 次的項目將被刪除。

優良資料庫

編號	維度項目
1	{維度_1, 維度_3, 維度_4, 維度_6, 維度_7, 維度_9, 維度_13, 維度_16}
2	{維度_1, 維度_2, 維度_3, 維度_6, 維度_9, 維度_15}
3	{維度_2, 維度_6, 維度_8, 維度_10, 維度_15}
4	{維度_2, 維度_3, 維度_11, 維度_16, 維度_17}
5	{維度_1, 維度_3, 維度_5, 維度_6, 維度_12, 維度_13, 維度_14, 維度_16}

↓ 最小支持度：50%

編號	維度項目
1	{維度_6, 維度_3, 維度_1, 維度_13, 維度_16}
2	{維度_6, 維度_3, 維度_1, 維度_2}
3	{維度_6, 維度_2}
4	{維度_3, 維度_2, 維度_16}
5	{維度_6, 維度_3, 維度_1, 維度_13, 維度_16}

圖 3.5 族群分類簡例。

圖 3.6 為依據選取後的交易資料所作的頻率項樹，經過最小支持度的篩選後，被選取的維度只剩下{維度_1, 維度_2, 維度_3, 維度_6, 維度_13, 維度_16}共六個維度，由頻率項樹，可找出個項目的關聯規則，此數最小信任度為 50%，各項目規則如下：

1. 維度_16 相關項目：{維度_6, 維度_3, 維度_1, 維度_13}出現次數 2，{維度_3, 維度_2}出現次數 1，出現次數超過 3 次的項目只有維度_3，因此與維度_16 相關的規則只有{維度_3, 維度_16}這一組，此組規則確立。

確立規則：{維度_3, 維度_16,}

2. 維度_13 相關項目：{維度_6, 維度_3, 維度_1}出現次數 2，無出現次數超過 3 次的項目，無規則產生。

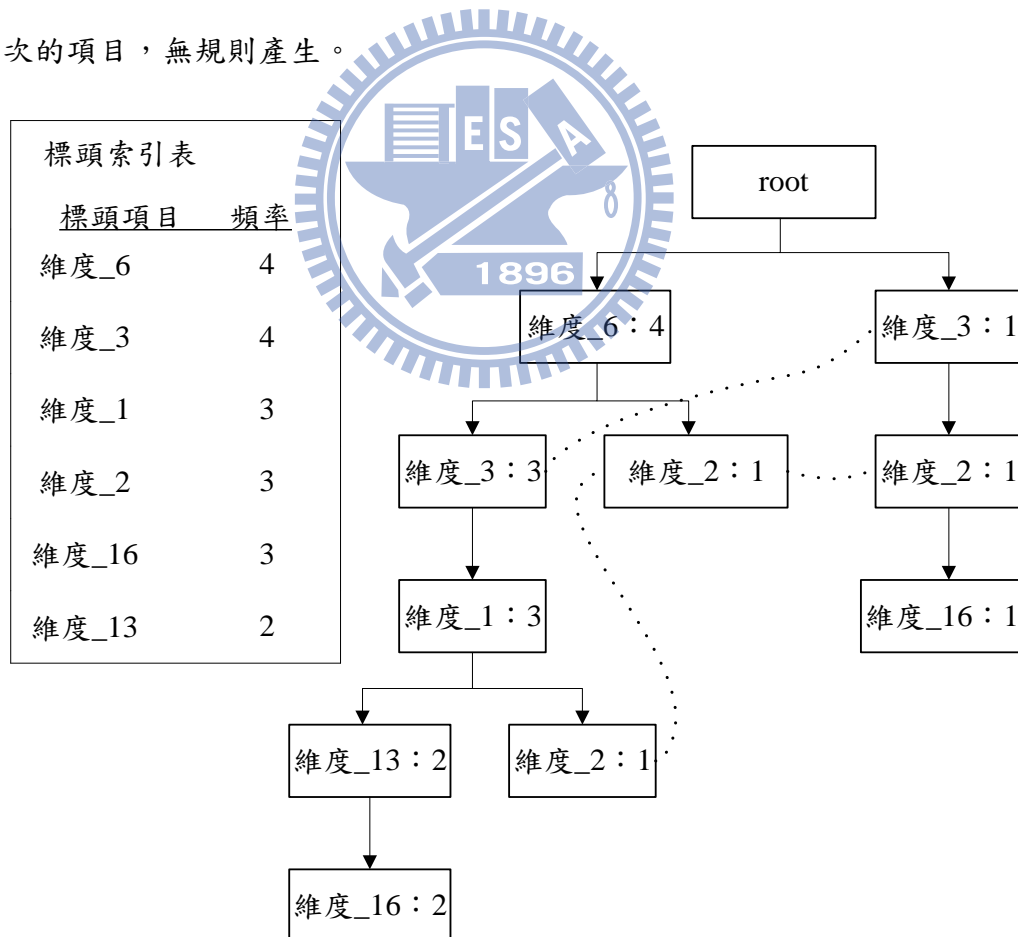


圖 3.6 族群分類簡例頻率項樹。

3. 維度_2 相關項目：{維度_6, 維度_3, 維度_1}出現次數 1，{維度_6}出現次數 1 次，{維度_3}出現次數 1，並無出現次數超過 3 次的項目，規則產生失敗，此處無規則產生。

此範例得到一組規則，規則一為{維度_3, 維度_16}，最後確立的規則含有兩個維度，除了這兩個維度外，其餘維度皆視為獨立的個體，不與其他維度具有關聯性，這些獨立的維度，將各自劃分到不同的族群裡，每個族群中只包含本身一個項目，因此最後可得到的族群分類規則為：族群一{維度_3, 維度_16}、族群二~族群十五分別為{維度_1}、{維度_2}、{維度_4}、{維度_5}、{維度_6}、{維度_7}、{維度_8}、{維度_9}、{維度_10}、{維度_11}、{維度_12}、{維度_13}、{維度_14}及{維度_15}，亦即此處經由關聯性的探勘後，16 個維度將分為 15 個族群，以這 15 個族群做為共生式人工免疫演算法抗體族群分類規則，依照此族群分類規則完成後續最佳化的演算。

3.2.2 族群分類規則確立後最佳化演算法

本文所改良最佳化演算法的重心落於抗體基元的分群上，訂立出一套合理的分類規則，規則確立後，再以共生式人工免疫系統完成整體最佳化過程，原始人工免疫系統與共生式人工免疫系統在抗體基元分類規則上相比較為單純，原始人工免疫系統將所有維度項目分於同一族群裡，每一次的演算若得到較好的最佳解，會一次更新所有維度項目的值。共生式人工免疫系統在族群分類上剛好與原始人工免疫系統相反，將所有維度全部獨立，每一個維度會放到一個族群裡，一次只會演化一個維度，其餘的維度皆以記憶細胞所得的上一代最佳解代替。

假設一個維度 6 的例子，各維度以 A~F 表示，族群個數為三，若分類規則為：族群一{A, C, F}、族群二{E}、族群三{B, D}。如圖 3.7 所示，A(n)代表 A 維度裡第 n 個抗體基元，基於頻繁樣式成長的共生式人工免疫系統會先最佳化第

一個族群，其中包含三個維度項目{A, C, F}，一次的演算法將一起更新這三個維度的最佳解，直到第一個族群最佳化過程完成後，在進行第二個族群的最佳化，此處族群二僅包含一個維度項目，因此族群二的最佳化過程一次只會更新一個維度的最佳解，以此類推，相較於原始人工免疫系統與共生式人工免疫系統，基於頻繁樣式成長的共生式人工免疫系統會因族群內所包含的維度數目不同，在最佳化過程一次更新的維度數目也會不同，這也是考慮到維度之間的關聯性所做的改進。

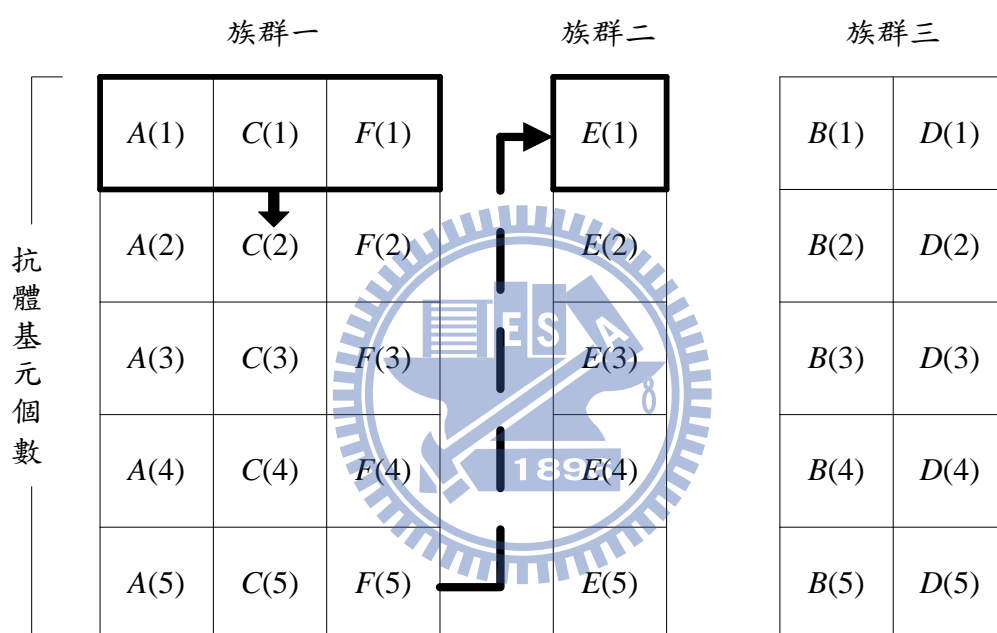


圖 3.7 基於頻繁樣式成長的共生式人工免疫系統演化示意圖。

3.3 函數模擬架構

本論文以模糊類神經網路做為函數模擬主要架構，如此的優點在於透過類神經網路，使用上不需建構一組模擬函數的基底函數，可更容易使用此架構達到模擬函數圖型的目標。

3.3.1 架構模糊類神經網路

本文的模糊類神經網路採用 TSK 模糊類神經系統，在架構上分為五層，分別為輸入層、歸屬函數層、規則層、推論層以及輸出層，圖 3.8 為輸入數目 2，規則數為 5 的 TSK 模糊類神經架構，以下將有各層詳細的介紹：

1. 第一層：輸入層，這一層提供輸入的管道，輸入在第一層中並不做任何運算，或可說輸入訊號與神經元間的權值為 1，僅將輸入傳送至第二層，此層公式如公式 3.1， $u_i^{(n)}$ 為第 n 層中第 i 個輸入的輸出，架構如圖 3.8 所示，輸入個數為 2。

$$u_i^{(1)} = x_i, \text{ for } i=1,2 \quad (3.1)$$

2. 第二層：歸屬函數層，模糊化第一層的輸出訊號，訊號會由在第一層的單一數值轉換成數個模糊集合，僅程度上不同，本文中使用的歸屬函數為高斯歸屬函數，計算公式如公式 3.2，此層中有兩個重要指標為平均值和標準差，分別以 m_{ij} 和 σ_{ij} 來表示， m_{ij} 對應到第 i 個輸入第 j 個歸屬函數的平均值，同理， σ_{ij} 對應到第 i 個輸入第 j 個歸屬函數的標準差。本文在此層中每個輸入對應到 5 個歸屬函數，因為輸入數目為 2，因此第二層中總共 10 個歸屬函數，每個歸屬函數有各自的平均值與標準差，換句話說，此層中所需最佳化的參數數目將有 20 個。

$$u_{ij}^{(2)} = \exp\left(-\frac{[u_i^{(1)} - m_{ij}]^2}{\sigma_{ij}^2}\right), \text{ for } i=1, 2; j=1, 2, \dots, 5 \quad (3.2)$$

3. 第三層：規則層，執行模糊規則啟動強度的運算，以模糊交及運算得到一個法則的激發量，運算公式如公式 3.3，此層中總共會有 5 個神經元，但此層中並無需要最佳化的參數產生。

$$u_j^{(3)} = \prod_i u_{ij}^{(2)}, \text{ for } i=1, 2; j=1, 2, \dots, 5. \quad (3.3)$$

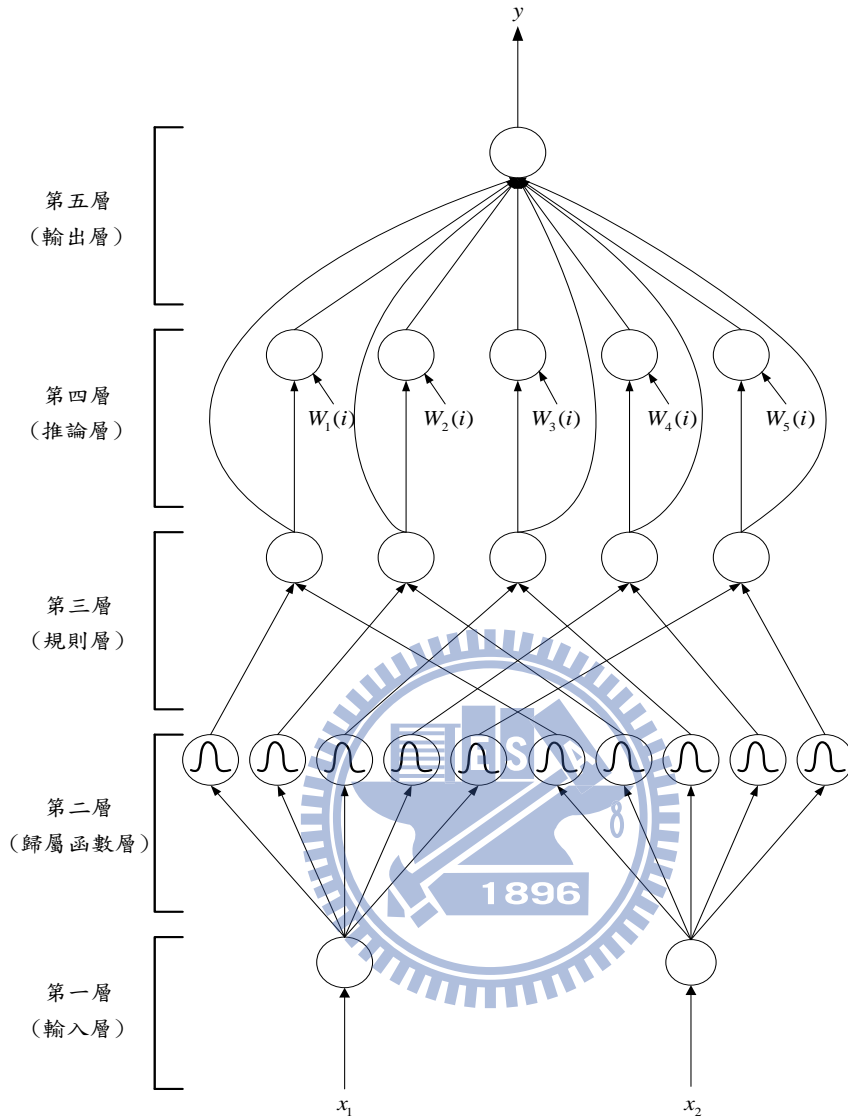


圖 3.8 本論文所使用 TSK 模糊類神經網路架構。

4. 第四層：推論層，此層中會將第三層的輸出，乘上一組線性組合式，以 $W_j(i)$ 表示，推論層用於推論規則層所得到的資訊，如公式 3.4 所示，其中 N 為輸入個數，每一組 $W_j(i)$ 會產生三個可調變數，例如 $W_1(i) = w_{01} + (w_{11}x_1 + w_{21}x_2)$ ，其中 w_{01} 、 w_{11} 以及 w_{21} 為使用者須調整參數，因此在此層中，總共會有 15 個參數須做最佳化的處理。

$$u_j^{(4)} = u_j^{(3)} W_j(i) \quad (3.4)$$

其中 $W_j(i) = w_{0j} + \sum_{i=1}^N w_{ij}x_i$, for $i=1, 2; j=1, 2, \dots, 5$.

5. 第五層：輸出層，將系統做解模糊化的動作，成為使用者所需要的資訊，如公式 3.5 所示， R 為規則數， N 為輸入數目。

$$y = u^{(5)} = \frac{\sum_{j=1}^R u_j^{(4)}}{\sum_{j=1}^R u_j^{(3)}} = \frac{\sum_{j=1}^R u_j^{(3)} (w_{0j} + \sum_{i=1}^N w_{ij}x_i)}{\sum_{j=1}^R u_j^{(3)}}, \quad (3.5)$$

其中 $i=1, 2; j=1, 2, \dots, 5$.

以 TSK 進行函數模擬具有三項優點：

1. 以模糊類神經模擬函數，不需建構模擬函數時所需的基底函數，更容易達到函數模擬的目標。
2. 類神經網路具備由輸入及輸出推論非線性函數的能力。
3. 常見模糊類神經網路為 TSK 與 Mamdani 兩類架構，差別在於 TSK 增加第四層即為推論層，此層乘上一組線性組合式，對於解決非線性問題具有較佳的能力。

模糊類神經系統表現的好壞，將直接受到上述提及的幾個參數所影響，參數的個數即為最佳化過程中變數的數目，即為系統的維度，意即系統維度的大小，將取決於規則數和輸入數目，參數個數的算法如公式 3.6 所示，本文中規則數為 5，輸入數為 2，因此總共會產生 35 個參數須做最佳化的調整，分別為 $\{m_{11}, \sigma_{11}, m_{12}, \sigma_{12}, m_{13}, \sigma_{13}, m_{14}, \sigma_{14}, m_{15}, \sigma_{15}, m_{21}, \sigma_{21}, m_{22}, \sigma_{22}, m_{23}, \sigma_{23}, m_{24}, \sigma_{24}, m_{25}, \sigma_{25}, w_{01}, w_{11}, w_{21}, w_{02}, w_{12}, w_{22}, w_{03}, w_{13}, w_{23}, w_{04}, w_{14}, w_{24}, w_{05}, w_{15}, w_{25}\}$ 共 35 個。

$$\text{參數個數} = 2 \times \text{規則數} \times \text{輸入數} + (1 + \text{輸入數}) \times \text{規則數}. \quad (3.6)$$

輸入數目由題型所決定，而規則數由使用者所訂定，規則數太少會使演算法

無法學習到所要的目標，但規則數越多並不代表系統會表現越好，過多的規則數容易造成演算過程中計算量的沉重負擔，例如若設定 7 個規則數，2 個輸入數，則最佳化過程將會面對將近 50 個維度的問題，這樣的維度大小，計算量相當龐大，將會造成演算上太過於沉重的負擔。

3.3.2 最佳化過程

上述 TSK 架構所得需最佳化參數為 35 個，本文以基於頻繁樣式成長法的共生式人工免疫系統做為調整參數的最佳化工具，此處族群分類的依據，透過章節 3.2 來確立規則，整體流程如圖 3.9。

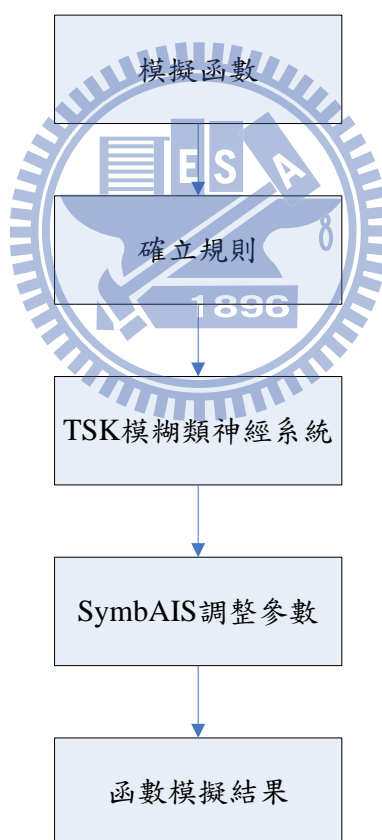


圖 3.9 函數模擬流程圖。

第四章 實驗結果與討論

本章節將運用於函數模擬，以改良後的共生式人工免疫系統結合 TSK 模糊類神經網路，將模糊類神經網路中的數個參數做最佳化的處理，使模擬後的函數圖形可接近於實際函數圖型。本研究實驗用設備為 AMD Phenom II X4 B55 3.6GHz 中央處理器，4GB 記憶體之桌上型電腦，使用 Matlab R2010a 做為撰寫程式及實驗模擬軟體。各節敘述如下，4.1 節為函數模擬圖型實驗，4.2 節為實驗模擬分析。

4.1 函數模擬

本節將以 TSK 模糊類神經網路進行函數模擬，利用第三章所提到確立後基於頻率項目成長的共生式人工免疫系統尋找 TSK 模糊類神經網路的各參數值，在演算過程中，藉由頻率項目成長找尋 TSK 各參數間的關聯性，以此改良後的共生式人工免疫系統使模糊類神經網路在函數模擬上更接近實際函數的圖形表現。

4.1.1 函數類型選取

本論文中提供三組函數進行函數圖型模擬，如以下方程式 4.1、方程式 4.2 及方程式 4.3，此處與函數最佳化不同，執行函數最佳化時，頻率項目成長用於尋找函數間各變數的關聯性，分類規則的訂立為方程式中 x_i 之間是否具有關連性，而函數模擬在訂立族群分類規則上對方程式裡的各變數並不感興趣，在此處頻率項目成長將尋找 TSK 模糊類神經系統中各變數的關聯性質，如各平均值及標準差，因此雖然函數模擬所提供的方程式看起來較函數最佳化方程式來的簡單且容易，但實際上函數模擬在演化上所處理的維度數目，大部分都會比函數最佳化來得多，執行上計算量的負擔也較重。

函數一：

$$y = \begin{cases} x, & 0 \leq x \leq 3, \\ 7.5 - 1.5x, & 3 < x \leq 5. \end{cases} \quad (4.1)$$

函數二：

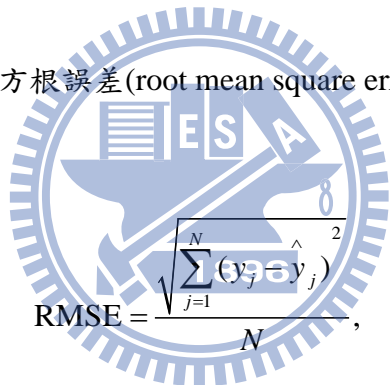
$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, x_2 \leq 5. \quad (4.2)$$

函數三：

$$y = \begin{cases} (1 + x_1^{-2} + x_2^{-1.5})^2, & 2 \leq x_2 \leq 3, \\ 5 + \sin(2x_1) + \sin(2x_2), & 3 < x_2 \leq 4. \end{cases} \quad (4.3)$$

4.1.2 函數模擬實驗結果

本文中效能比較以均方根誤差(root mean square error, RMSE)為標準，定義如公式 4.4 所示，



$$\text{RMSE} = \frac{\sqrt{\sum_{j=1}^N (y_j - \hat{y}_j)^2}}{N}, \quad (4.4)$$

其中 N 為取點個數； y_j 為 j 點的實際值； \hat{y}_j 為 j 點時本系統輸出值。

本文在 TSK 參數最佳化所使用演算法為改良後的共生式人工免疫系統，因此在比較對象上，以原始人工免疫系統和共生式人工免疫系統做為參數最佳化的兩種方法。本論文等三種方法在 function evaluations 數目皆為 1000 次，函數一與函數二設定如表 4.1 所示。

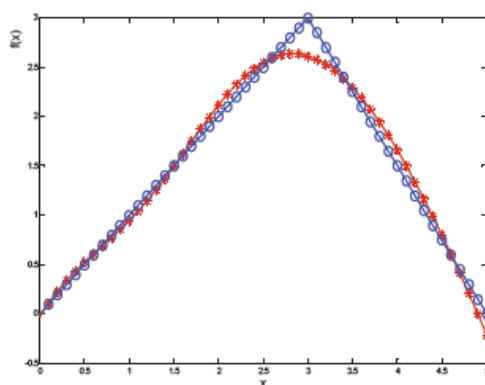
表 4.1 函數初始設定。

函數編號	TSK 規則數目	初始範圍	演化門檻值
函數一	5	5.12	100
函數二	5	10	100
函數三	5	10	100

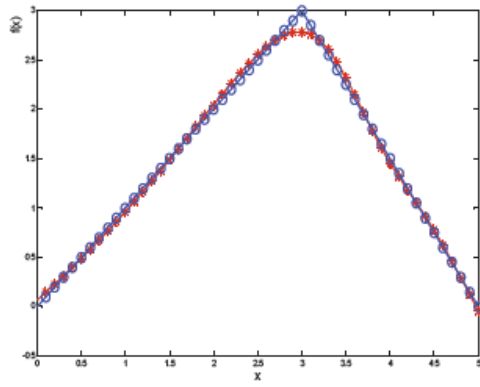
表 4.2 為函數一模擬結果，由表中可看出，以原始人工免疫系統為參數最佳化方法在表現上較不理想，而共生式人工免疫系統雖然表現較原始人工免疫系統好，但以均方根誤差來看，仍然是以本文所提出改良後方法較能最好，圖 4.1 為三種方法模擬圖型，本論文與共生式人工免疫系統較接近實際圖，且在變化較大的部分，本論文明顯優於其他兩種演算法。圖 4.2 為原始人工免疫系統、共生式人工免疫系統與本論文三種方法與實際情形的誤差值。第三章演算法流程圖為利用模糊集中歸屬函數重疊性是否過高來判斷分類規則數是否恰當與未利用此方法調整規則數所需運算時間比較，若原本函數分類規則數過多則函數之歸屬函數重疊性會過高，而這邊重疊性(overlapping)亦稱為相似性(similarity)，這時利用提高最小信任度及最小支持度來減少冗餘之規則數(redundancy)來達到增進演算法速度的目的。此處令 λ 為其相似性的值，當其值越大代表重疊率越高。表 4.3 為函數模擬中函數一 λ 不同時之數據。表 4.4 為與未利用歸屬函數相似刪減規則數之時間比較。

表 4.2 函數一模擬結果。

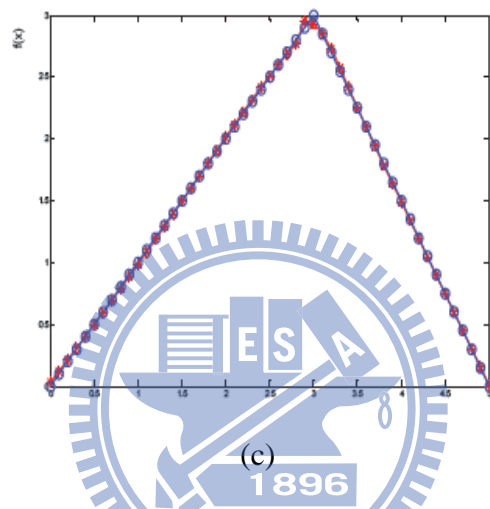
	均方根誤差
以原始人工免疫系統調整	0.0189
以共生式人工免疫系統調整	0.0071
以本論文調整 TSK 參數	0.0021



(a)



(b)



(c)

圖 4.1 函數一模擬結果：(a)以原始人工免疫系統調整 TSK 參數，(b)以共生式人工免疫系統調整 TSK 參數(c)以本論文調整 TSK 參數。

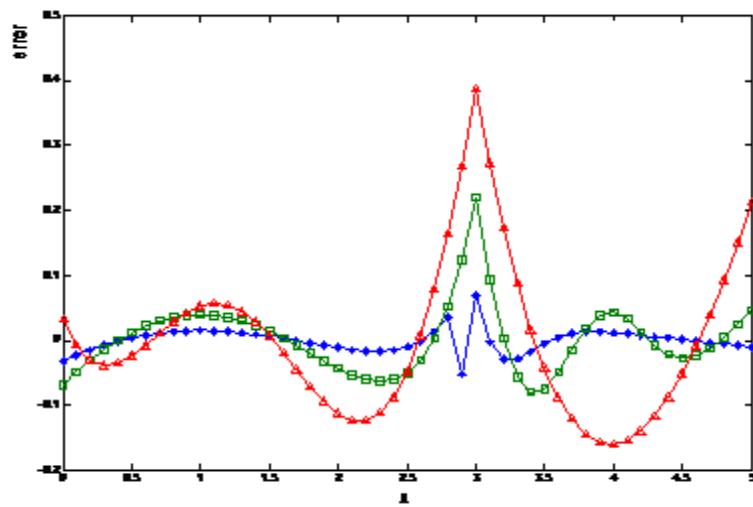


圖 4.2 三種方法誤差值。

表 4.3 函數模擬中函數一 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.55	25	0.0039
0.60	29	0.0026
0.65	30	0.0021
0.70	32	0.0022
0.75	33	0.0022
未調整	35	0.0023

表 4.4 函數一利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	2.948
未調整	3.039

表 4.5 為函數二模擬結果，函數二在困難度上較函數一來得高，因此效能表現上會比模擬函數一來的差。由表 4.5 可看出，模擬結果依舊以本文所使用方法最好，若分別觀察三種方法，原始人工免疫系統演化速度快，演化初期效果也不錯，但當需要做較準確的微調參數時，此時效果較差，並無法得到好的模擬結果；共生式人工免疫系統雖然速度慢，但其演化效果在各階段較為平均，也可以比較準確調整較細微的部分，在表現上優於原始人工免疫系統；本文提出之方法在三中方法中表現最好，在演化速度上介於兩者之間，在演化過程中，無論是在細微處的微調，或是效能表現上皆優於其餘兩種，圖 4.3 為三種方法用於模擬函數二圖型。表 4.6 為函數模擬中函數二 λ 不同時之數據。表 4.7 為與未利用歸屬函數相似刪減規則數之時間比較。

表 4.5 函數二模擬結果。

	均方根誤差
以原始人工免疫系統調整 TSK 參數	0.0114
以共生式人工免疫系統調整 TSK 參數	0.0079
以本論文調整 TSK 參數	0.00397

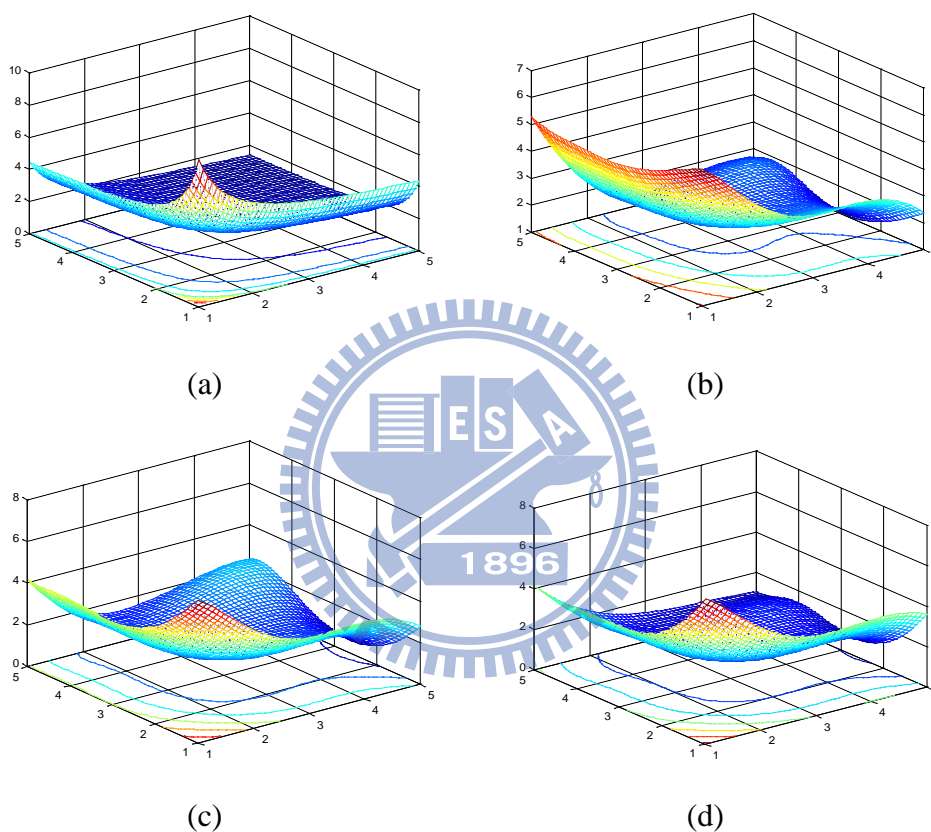


圖 4.3 函數二模擬結果：(a)實際圖型，(b)以原始人工免疫系統調整 TSK 參數，(c)以共生式人工免疫系統調整 TSK 參數，(d)以本論文調整 TSK 參數。

表 4.6 函數模擬中函數二 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.45	26	0.00442
0.50	27	0.00405
0.55	29	0.00397
0.60	33	0.00398
0.65	35	0.00413
未調整	35	0.00414

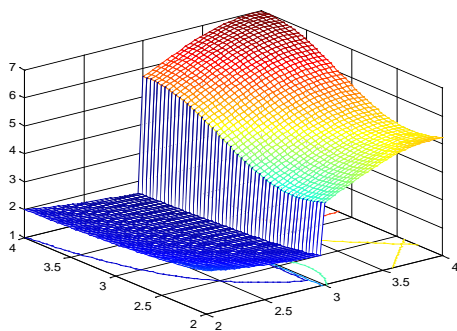
表 4.7 函數二利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	5.487
未調整	7.011

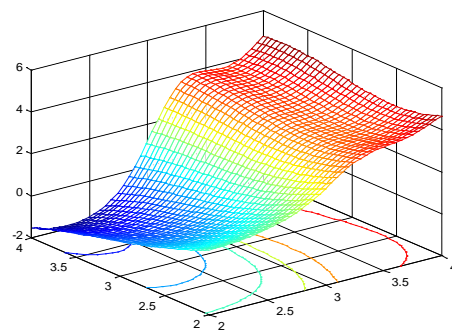
表 4.8 為函數三模擬結果，函數三為一不連續函數，在圖形中具有一段層面，圖 4.4 為三種方法模擬函數三圖型。表 4.9 為函數模擬中函數三 λ 不同時之數據。表 4.10 為函數三利用歸屬函數之重疊性調整規則數前後之時間比較。

表 4.8 函數三模擬結果。

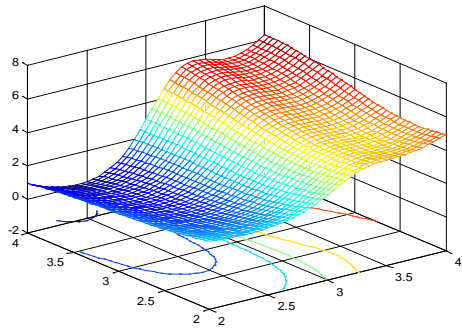
	均方根誤差
以原始人工免疫系統調整 TSK 參數	0.0412
以共生式人工免疫系統調整 TSK 參數	0.0265
以本論文調整 TSK 參數	0.0104



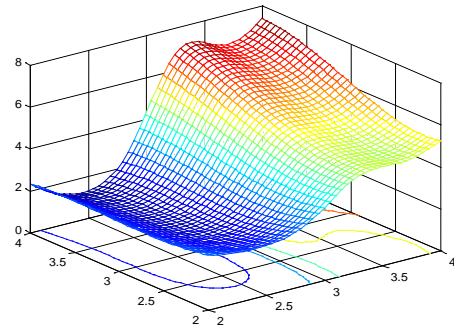
(a)



(b)



(c)



(d)

圖 4.4 函數三模擬結果：(a)實際圖型，(b)以原始人工免疫系統調整 TSK 參數，
(c)以共生式人工免疫系統調整 TSK 參數，(d)以本論文調整 TSK 參數。

表 4.9 函數模擬中函數三 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.70	30	0.0206
0.75	31	0.0178
0.80	34	0.0104
0.85	35	0.0120
0.90	35	0.0120
未調整	35	0.0122

表 4.10 函數三利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	13.918
未調整	14.317

函數三在難度上遠高於函數一和函數二，主要在斷層面上逼近困難，表 4.8 及圖 4.4 可看出，模擬函數三得到的結果並不理想，但本論文在表現上仍可優於其他兩種方法。

4.2 函數最佳化

本節將提供三個函數進行最佳化的模擬，函數的種類包含各維度間相關或各維度間獨立以及不全相關及不全獨立的不同類型，透過頻率樣式成長法找出相關聯的維度，確立抗體基元分類標準，使演算法可達到更好的效果。

4.2.1 最佳化方程式選取

本論文最佳化模擬的方程式選了以下三種：Rosenbrock function、Ackley's function 以及 generalized Rastrigin function，分別以 f_0 、 f_1 及 f_2 表示，如以下方程式所示， n 為各方程式中維度個數，本實驗中 n 值皆為 30，意即皆具有 30 個維度。

Rosenbrock function：

$$f_0(x) = \sum_{i=1}^{\frac{n}{2}} (100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2). \quad (4.5)$$

Rosenbrock function 方程式維度間關係維全相關，並在 $n > 3$ 時出現局部最佳解。

圖 4.5 為函數實際圖形。

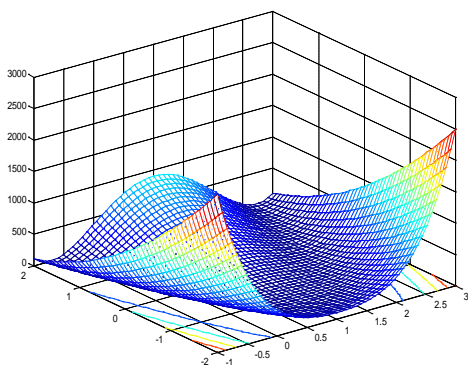


圖 4.5：Rosenbrock function 實際圖形。

Ackley's function：

$$f_1(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e. \quad (4.6)$$

Ackley's function 維度之間全獨立 4.並存在許多最佳局部解，圖 4.6 為 Ackley's

function 之實際圖形。

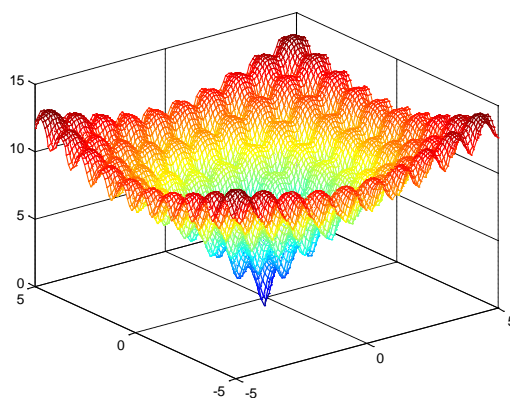


圖 4.6 Ackley's function 之實際圖形。

generalized Rastrigin function :

$$f_2 = \left(\sum_{i=1}^{m/2} (100(x_{2i} - x_{2i-1})^2 + (1 - x_{2i-1})^2) \right) + \left(\sum_{i=m+1}^n x_i^2 - 10 \cos(2\pi x_i) + 10 \right). \quad (4.7)$$

Generalized Rastrigin function 方程式維度間全獨立並存在許多局部最佳解。圖 4.7 為 generalized Rastrigin function 之實際圖形。

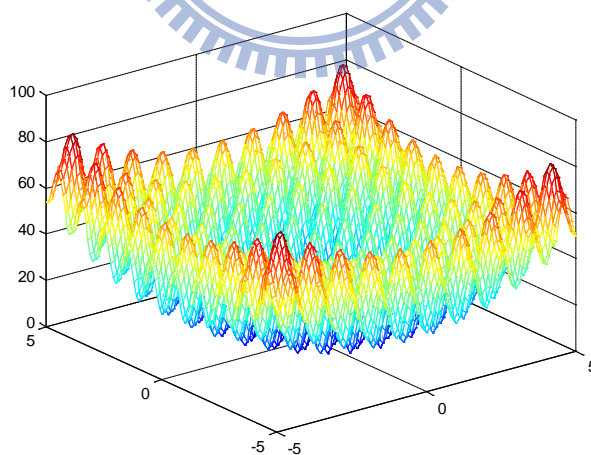


圖 4.7 generalized Rastrigin function 實際圖形。

f_0 各維度間完全相關每個維度之間都具有關聯性，意即關聯的情況並不只是其中某幾個維度，而是所有的維度都是關聯項目，不同的維度將彼此影響其它維度的演化，也就是說此種方程式若抗體基元分類結果較接近原始人工免疫系統，

如果能把所有維度分在同一類，在演化上的效果會較好； f_1 各維度間為完全獨立，每個維度並不受其它維度的影響，此種情況在抗體基元分類上，若能將所有維度各自分到不同的族群，意即類似於共生式人工免疫系統的分類準則在表現上會得到較好的效果； f_2 維度部分部分相關且部分獨立，可用於判斷本論文演算法是否具備提升效能的能力。本文中所提出基於頻繁樣式成長法的共生是人工免疫系統，便是為了達到這樣的結果，對於維度相關方程式或是維度非相關方程式甚至於維度間部分相關且部分獨立方程式，皆能做出最適當的處理。

4.2.2 函數最佳化實驗結果

本論文在函數最佳化部分比較的三種方法在演算上每個族群維度大小並不相同，因此若以程式執行速度來比較並不公平，另外這三種演算法的演算方式有所差異，每一個演化代中所做的份量亦不相同，因此也不可以演化代數做為相同的限制，因此本論文在停止條件的設定上，以 function evaluations(FEs)的數量做為準則，每運算一次適切值則 FEs 值增加 1。表 4.11 為各方程式初始設定參數。

表 4.11 實驗參數初始設定。

函數編號	函數維度	初始範圍	演化門檻值 (threshold)
f_0	30	2.048	100
f_1	30	30	5.00
f_2	30	30	5

本論文所使用的方法以及比較的兩種演算法在 function evaluations 的數目上皆設定為 2×10^5 ，所有實驗的數據皆測試 50 次，最後所得函數值為 50 次實驗的平均值。在抗體基元數的取法上，為使模擬可以得到較客觀的結果，本論文先做

抗體基元數目測試，表 4.12~表 4.14 為 $f_0 \sim f_2$ 粒子數由 1~50 的實驗結果。

表 4.12 f_0 抗體基元數目測試結果。

f_0	抗體 基元 個數	函數值	抗體 基元 個數	函數值	抗體 基元 個數	函數值
	1	34.6844	18	0.00733	35	0.00764
	2	22.9581	19	0.00791	36	0.00724
	3	4.27145	20	0.00699	37	0.00756
	4	1.54612	21	0.00692	38	0.00761
	5	0.67832	22	0.00698	39	0.00764
	6	0.36900	23	0.00713	40	0.00767
	7	0.30596	24	0.00713	41	0.00771
	8	0.23573	25	0.00729	42	0.00789
	9	0.19382	26	0.00735	43	0.00802
	10	0.15088	27	0.00744	44	0.00792
	11	0.10399	28	0.00762	45	0.00771
	12	0.05529	29	0.00765	46	0.00734
	13	0.01938	30	0.00762	47	0.00823
	14	0.00921	31	0.00768	48	0.00894
	15	0.00703	32	0.00763	49	0.00719
	16	0.00702	33	0.00722	50	0.00772
	17	0.00701	34	0.00742		

表 4.13 f_1 抗體基元數目測試結果。

f_1	抗體 基元 個數	函數值	抗體 基元 個數	函數值	抗體 基元 個數	函數值
	1	0.074218	18	0.000177	35	0.000169
	2	0.028183	19	0.000173	36	0.000179
	3	0.007915	20	0.000169	37	0.000159
	4	0.001627	21	0.000169	38	0.000166
	5	0.001061	22	0.000170	39	0.000162
	6	0.000734	23	0.000167	40	0.000172
	7	0.000601	24	0.000163	41	0.000179
	8	0.000355	25	0.000172	42	0.000171
	9	0.000241	26	0.000160	43	0.000173
	10	0.000197	27	0.000173	44	0.000172
	11	0.000183	28	0.000183	45	0.000162
	12	0.000186	29	0.000169	46	0.000181
	13	0.000174	30	0.000168	47	0.000153
	14	0.000176	31	0.000165	48	0.000178
	15	0.000173	32	0.000166	49	0.000162
	16	0.000179	33	0.000167	50	0.000168
17	0.000177	34	0.000170			

表 4.14 f_2 粒子數測試結果。

f_2	抗體 基元 個數	函數值	抗體 基元 個數	函數值	抗體 基元 個數	函數值
	1	3.5399017	18	0.0000153	35	0.0000146
	2	1.3345612	19	0.0000134	36	0.0000147
	3	0.6339643	20	0.0000142	37	0.0000142
	4	0.0235110	21	0.0000133	38	0.0000158
	5	0.0514271	22	0.0000138	39	0.0000132
	6	0.0036362	23	0.0000140	40	0.0000140
	7	0.0007339	24	0.0000143	41	0.0000149
	8	0.0000533	25	0.0000159	42	0.0000147
	9	0.0000493	26	0.0000151	43	0.0000152
	10	0.0000328	27	0.0000149	44	0.0000159
	11	0.0000287	28	0.0000152	45	0.0000169
	12	0.0000231	29	0.0000151	46	0.0000148
	13	0.0000210	30	0.0000142	47	0.0000149
	14	0.0000191	31	0.0000148	48	0.0000152
	15	0.0000193	32	0.0000157	49	0.0000144
	16	0.0000188	33	0.0000148	50	0.0000148
	17	0.0000162	34	0.0000157		

由表 4.12~表 4.14 可發現，各函數在抗體基元數小於 10 時有較明顯的變化，維度間抗體基元數過小會導致整體效能嚴重受到影響，當抗體基元數大於 25 時，表現上變化不大，因此本論文在抗體基元數的選取上，以基元 11~25 間隔 1 的情

況進行模擬。

演算法模擬 f_0 的實驗結果比較如表 4.15~表 4.17 所示，此方程式為一個維度完全相關的例子，每個維度間彼此都具有關聯性，因此透過本論文的族群分類機制，會將所有的維度皆放於同一族群裡，如此的分類結果與 AIS 的族群分類方式是相同的，所以在演化上會與 AIS 所得到的結果相近，由表 4.15 可以看出，此一維度完全相關的函數，若以 AIS 進行演化會比 SymbAIS 來得有效，雖然 SymbAIS 可避免 AIS 演算法的缺點，但 CPSO-S 將關聯維度劃分開來所造成的缺陷遠大於自身演算法所提供的優點，因此結果如同第二章所述，維度關聯函數仍以 AIS 可得到較好結果。表 4.18 為方程式 f_0 當 λ 不同時之數據。表 4.19 為利用歸屬函數之重疊性調整規則數前後之時間比較。

表 4.15 f_0 使用本論文方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_0	本論文	11	0.15088
		12	0.09845
		13	0.01427
		14	0.00846
		15	0.00703
		16	0.00701
		17	0.00697
		18	0.00694
		19	0.00693
		20	0.00699
		21	0.00690
		22	0.00687
		23	0.00688
		24	0.00691
		25	0.00692

表 4.16 f_0 使用 AIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_0	AIS	11	0.14876
		12	0.11279
		13	0.04117
		14	0.00915
		15	0.00653
		16	0.00664
		17	0.00670
		18	0.00688
		19	0.00702
		20	0.00757
		21	0.00759
		22	0.00755
		23	0.00750
		24	0.00744
		25	0.00741

表 4.17 f_0 使用 SymbAIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_0	SymbAIS	11	0.73659
		12	0.72144
		13	0.69584
		14	0.67116
		15	0.66840
		16	0.66421
		17	0.70120
		18	0.74563
		19	0.78119
		20	0.83584
		21	0.83667
		22	0.84579
		23	0.84892
		24	0.85002
		25	0.85302

表 4.18 方程式 f_0 當 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.60	15	0.0124
0.65	21	0.00693
0.70	25	0.00687
0.75	28	0.00688
0.80	29	0.00701
未調整	30	0.00716

表 4.19 f_0 利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	12.819
未調整	15.224

表 4.20~表 4.22 為方程式 f_1 於各種實驗方法所得到的結果， f_1 的各維度間彼此獨立，所有維度皆不相關，因此本論文在此實驗中，族群分類後的結果為所有維度皆放於不同的族群裡，總共產生族群個數為 30。此種分法與 SymbAIS 相同，意即本論文的方法在實驗數據將會於 SymbAIS 相近。由實驗數據可明顯看出，當函數的類型為維度間具有相關性，且維度數目較大時，SymbAIS 的表現會明顯優於 AIS，本論文藉由 FP-growth 尋找關聯項目，應用於此類型方程式也可達到與 SymbAIS 相近的結果。表 4.23 為方程式 f_1 當 λ 不同時之數據。表 4.24 為 f_1 利用歸屬函數之重疊性調整規則數前後之時間比較。

表 4.20 f_1 使用本論文方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_1	本論文	11	0.000196
		12	0.000190
		13	0.000187
		14	0.000182
		15	0.000173
		16	0.000172
		17	0.000172
		18	0.000170
		19	0.000169
		20	0.000169
		21	0.000169
		22	0.000168
		23	0.000168
		24	0.000165
		25	0.000165

表 4.21 f_1 使用 AIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_1	AIS	11	6.83013
		12	6.44207
		13	5.91790
		14	5.42632
		15	4.81892
		16	4.67119
		17	4.21585
		18	4.00148
		19	3.80914
		20	3.70377
		21	3.70006
		22	3.67158
		23	3.65912
		24	3.62007
		25	3.60661

表 4.22 f_1 使用 SymbAIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_1	SymbAIS	11	0.000255
		12	0.000249
		13	0.000232
		14	0.000217
		15	0.000197
		16	0.000194
		17	0.000193
		18	0.000190
		19	0.000187
		20	0.000186
		21	0.000183
		22	0.000182
		23	0.000179
		24	0.000174
25	0.000172		

表 4.23 方程式 f_1 當 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.55	16	0.0107
0.60	21	0.00457
0.65	26	0.000165
0.70	29	0.000168
0.75	30	0.000179
未調整	30	0.000181

表 4.24 f_1 利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	15.417
未調整	18.001

表 4.25~表 4.27 為 f_2 於各實驗方法所得結果， f_2 為一函數部分相關且部分獨立的方程式，原始人工免疫系統與共生式人工免疫系統因為在分類的設定

上，都無法與實際情況得到較接近的分類結果，因此在實驗結果上，因為本論文在族群分類規則上較自由，並沒有一開始便設定好的情況，可依照題型的不同做自主調整，可得到較佳的最佳解搜尋，尤其是如函數 f_2 這類原始人工免疫系統與共生式人工免疫系統皆表現不好的題型，本論文對於提升演算法的效能優點，可表現出來。表 4.28 為方程式 f_2 當 λ 不同時之數據。表 4.29 為 f_2 利用歸屬函數之重疊性調整規則數前後之時間比較。

表 4.25 f_2 使用本論文方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_2	本論文	11	0.000031
		12	0.000029
		13	0.000025
		14	0.000022
		15	0.000019
		16	0.000019
		17	0.000017
		18	0.000016
		19	0.000014
		20	0.000013
		21	0.000013
		22	0.000013
		23	0.000013
		24	0.000014
		25	0.000014

表 4.26 f_2 使用 AIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_2	AIS	11	2.3008
		12	2.2746
		13	2.2018
		14	2.1156
		15	2.0079
		16	1.9746
		17	1.8432
		18	1.8006
		19	1.6155
		20	1.5442
		21	1.5979
		22	1.5998
		23	1.6071
		24	1.6114
		25	1.6213

表 4.27 f_2 使用 SymbAIS 方法實驗結果。

函數	演算法	各維度粒子數	函數值
f_2	SymbAIS	11	0.00884
		12	0.00807
		13	0.00754
		14	0.00716
		15	0.00631
		16	0.00620
		17	0.00594
		18	0.00555
		19	0.00526
		20	0.00517
		21	0.00520
		22	0.00524
		23	0.00531
		24	0.00542
25	0.00553		

表 4.28 方程式 f_2 當 λ 不同時之數據。

λ	規則數目(個)	RMSE
0.30	17	0.000048
0.35	21	0.000026
0.40	25	0.000013
0.45	27	0.000014
0.50	28	0.000015
未調整	30	0.000022

表 4.29 f_2 利用歸屬函數之重疊性調整規則數前後之時間比較。

方法	所需執行時間(秒)
演算法利用歸屬函數之重疊性判斷規則數是否過多進而調整最小支持度及最小信任度	38.645
未調整	39.078

4.3 模擬結果分析

在函數模擬的實驗中，以原始人工免疫系統為參數最佳化演算法雖然速度較快，但卻只能做大略的調整，並無法做到較細微的圖形曲線逼近；而以共生式人工免疫系統做為最佳化演算法速度上較慢，但卻可以較準確逼近圖型；本論文在速度上介於原始人工免疫系統與共生式人工免疫系統之間，分類結果大多會將每一組輸入在歸屬函數層的平均值和標準差置於同一族群內，如公式 3.2 所示，平均值和標準差在 TSK 模糊類神經系統中為相關變數，因此這樣的分類結果有助於演算法效能的表現，此外本文以改良式共生式人工免疫系統做為參數最佳化工具，最重要的部分為基於頻繁樣式成長的共生式人工免疫系統在演化上彈性大，可解決共生式人工免疫系統對有相關聯項目演化困難的窘境，因此本文在模擬結果上，皆比其餘兩者來得好。此次實驗中，有個較值得注意的部分，表 4.9 中 AIS 演算法當各維度包含 15 個粒子以及 20 個粒子時，由實驗數據看出，維度中包含的粒子數越多，有時並不一定會使演化效果更好，主要原因在於本實驗不以演化代數做為限制，而是以 function evaluations 代替演化代數做為演算法終止條件，意即當每個維度中粒子數增加，每一次演化代中將消耗更多 function evaluations 的次數，若大略以演化代數來看，維度中粒子數多的實驗，最佳化過程中起始至演化結束，執行的演化代數會比維度中粒子數較少的實驗來得少，以致於會出現粒子數較多反而較差的情況。但粒子數的影響主要在於個數小於 10 的情況，因此當粒子個數足夠時，彼此間差異並不大。

在函數最佳化的實驗中，維度內抗體基元的個數會影響效能的表現，尤其當個數過於小時，會造成較大的影響，因此在使用的建議上，可將抗體基元個數設定在 15~25 之間，彼此之間差異不大。

函數 f_0 各維度間皆為相關，因此本論文在確立規則步驟時，當測試資料族群分類情況與原始人工免疫系統相近，意即把所有的維度置於一個族群內，以實驗

結果來看，因為函數 f_0 的維度全相關性，因此結果如預期的想法，本論文與原始人工免疫系統表現會較共生式人工免疫系統好。函數 f_1 各維度間都是彼此獨立且維度數目大，這時原始人工免疫系統將所有抗體基元分到同一族群的分類方式會使演算法效能降低，而本論文與共生式人工免疫系統將抗體基元分別分到各個族群演化，而得到較好的結果。函數 f_2 維度部分相關且部分獨立，因此原始人工免疫系統及共生式人工免疫系統在本實驗中表現並不好，本論文演算法在族群分類上彈性較佳，可使族群分類較接近實驗函數的實際情況，因此本論文在函數 f_2 的實驗上，可達到提升共生式人工免疫演算法效能。表 4.24 為整理三種演算法於各函數的結果對照表。

表 4.30 各方法應用於三種函數結果對照表。

		本論文	AIS	SymbAIS
f_0	族群分類規則	所有維度置於同一族群	所有維度置於同一族群	所有維度分開至不同族群
	表現優劣	良好	良好	差
f_1	族群分類規則	所有維度分開至不同族群	所有維度置於同一族群	所有維度分開至不同族群
	表現優劣	良好	差	良好
f_2	族群分類規則	部分分於各不同族群，部分各自獨立	所有維度置於同一族群	所有維度分開至不同族群
	表現優劣	良好	差	差

第五章 結論

本論文將共生式人工免疫系統結合資料探勘，以頻率項目成長尋找各維度間的關聯性，如此一來便可解決面對抗體基元之間有相關聯性而造成演算法效能不彰的窘境，使用者在一開始便可訂立出合理的抗體集合分類方法，解決共生式人工免疫系統所遇到的瓶頸，由第四章實驗結果可看出，透過關聯性的尋找，所得到的族群分類結果為合理且實用，省去了很多利用嘗試錯誤法所浪費的時間。

此外，當處理的問題具有過大的維度時，因為共生式人工免疫系統在維度過大的情況下效而表現不佳，每一次的演化中，僅考量整體的適切值，各維度是否較上一代來得好，共生式人工免疫系統並無法確保，對於最佳解的找尋造成相當大影響，因此多數使用者在沒有族群分類的規則下，會以共生式人工免疫系統做為最佳化過程中所使用的演算法，但如此一來，將所有維度分到不同的族群裡，完全不考慮維度間是否具有關聯性，會造成計算量龐大的問題，且對關聯維度來說並不理想，雖然仍可比原始人工免疫系統較有效達到找尋最佳解的目的，但卻會因為沉重的計算量而使計算時間相當冗長。透過本論文所提出的方法，將關聯維度找出並置於同一族群裡，可有效的使族群數目減少，在計算量上，可達到縮小的效果，使執行速度上可以更加快速，另外在最佳化過程中，也可避免關聯維度被分到不同族群所造成不好的影響，因此不僅在演化計算量上，以及最佳解的找尋，本論文都可達到較不錯的表現。

未來希望可透過關聯法則，進行設定參數的調整，使參數不需經由一連串的錯誤嘗試法來決定。

参考文献

- [1] J. H. Holland, *Adaptation in Natural and Artificial System*, Cambridge: MIT Press, 1992.
- [2] J. D. Bagley, “The behavior of adaptive systems which employ genetic and correlation algorithms,” *Ph. D. dissertation*, Michigan, USA, 1967.
- [3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.
- [4] J. E. Hunt and D. E. Cooke, “Learning using an artificial immune system,” *Journal of Network and Computer Applications*, pp. 189-212, 1996.
- [5] D. Dasgupta and N. Attouh-Okine, “Immunity-Based Systems : A Survey,” *Proceeding of the IEEE Transactions on Systems, Man and Cybernetics*, vol. 1, pp. 369 -374, 1997.
- [6] T. Fukuda, K. Mori, and M. Tsukiama, “Parallel search for multi-modal function optimization with diversity and learning of immune algorithm,” *Artificial Immune Systems and Their Application*, Springer-Verlag, pp. 210-220, 1999.
- [7] L. N. de Castro and F. J. Von Zuben, “Learning and optimization using the clonal selection principle, ” *IEEE Transactions on Evolutionary Computation*, vol. 6 no. 3, pp. 239–251, 2002
- [8] R. Halavati and S. B. Shouraki, “Symbiotic immune system, ” - *Soft Computing- A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 6, pp. 565-575, 2009.
- [9] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proc. ACM Special Interest Group on Management of Data*, pp. 207-216, 1993.

- [10] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," in *Proc. Very Large Data Bases*, pp. 487-499, 1994.
- [11] Y. Dote, "Soft Computing (Immune Networks) in Artificial Intelligence," *Proceeding of the IEEE Transactions on Systems, Man and Cybernetics*, vol. 2, pp.1382 -1387,1998.
- [12] H. G. Cobb, "Is the genetic algorithm a cooperative learner?" *Foundations of Genetic Algorithms 2.*, pp. 277-296, 1992.
- [13] S. H. Clearwater, T. Hogg, and B. A. Huberman, "Cooperative problem solving," in *Computation: The Micro and Macro View*, pp. 277-296, 1992.
- [14] P. Cabena and P. Hadjnian, *Discovering Data Mining*, Upper Saddle River: Prentice Hall, 1998.
- [15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, San Francisco: Elsevier, 2001.
- [16] J. Pei and J. Han, "Can we push more constraints into frequent pattern mining?" in *Proc. Int. Conf. Knowledge Discovery and Data Mining*, pp. 350-354, 2000.
- [17] S. Park, W. W. Chu, J. Yoon, and C. Hsu, "Efficient searches for different lengths of FP-growth algorithm," in *Proc. 4th Int. Conf. Information and Knowledge Management*, pp. 31-36, 2003.
- [18] A. Russell and R. S. Probablistic, "Reasoning about data and hash-based algorithm for mining association rules," in *Proc. ACM Special Interest Group on Management of Data*, pp. 245-263, 2002.
- [19] 蘇木春，張孝德，機器學習、類神經網路、模糊系統以及基因演算法則，全華科技圖書股份有限公司。
- [20] G. Castellano, A. M. Fanelli, and C. Mencar, "A neuro-fuzzy network to generate human-understandable knowledge from data," *Cognitive Systems Research* vol. 3, no. 2 pp. 125-144, 2002.