

基因演算法在國小排課問題之應用

邱炤樟
屏東科技大學資訊管理學系研究生
Onioni814@hotmail.com

陳灯能
屏東科技大學資訊管理學系副教授
dnchen@mail.npust.edu.tw

摘要

在一般學校的行政事務中，排課問題一直都是每學期開學前最重要且複雜的問題之一，為了符合教師及學生的需求，也必須滿足課程設計及教室空間等限制，如何安排出可以符合所有限制條件的課表一直都是令人困擾的問題。然而排課問題已經被證明是 NP-Complete 問題，並不存在最佳的解決辦法，也因此多數的排課工作都是由人工的方式來進行，然而人工排課卻是一件非常浪費時間的事情。本研究希望透過基因演算法，透過軟性及硬性條件限制的設計，藉由基因演算法的自我演化能力，達到自動安排出可接受課表之目的。本研究採用 Web 為系統的基礎架構，便於使用者操作系統，並以兩個國小的課程安排為例，實際驗證本系統之效能。

關鍵字：基因演算法、排課

1. 前言

1.1 研究背景與動機

在排課問題之中，教師、班級、節次、科目為四個主要因素，排課時必須先使每班在每個節次只排入一個科目，再讓每位老師的上課節次不衝堂，其次為上課教室安排、排課法規的考量、教師不排課時段的設定、學校特殊節次安排等，讓課表可以更符合法規、學校和老師的期望。

在排課過程中，最大的困難在於，若是要排出符合期望的課表，就必須滿足多重條件的限制，如教室資源的限制、教師不排課時段的限制、主要課程與次要課程安排時段的限制等，當限制越多，排課工

作的困難度也相對提升。排課問題也可以說是一種排程問題，Even et al. (1975)曾透過實驗證明排課問題是一個 NP-Complete 的問題。因此，如何找出一種合適的演算法，找尋出一個我們所需要的可接受解，是一個值得研究的問題。

在眾多解決排程問題的演算法中，基因演算法(Genetic Algorithm, GA)是表現相當優異的一種，適合用來解決複雜度高的排程問題。本研究希望能將基因演算法應用於國小排課問題上，設計一套以 Web 介面為操作方式的平臺，讓排課人員只需透過瀏覽器做設定，就能自動的完成排課工作。

1.2 研究問題探討

在各級學校的排課問題中，其差異可說是非常大，排課時要考慮的變數也有所不同，本研究討論的範圍是國民小學的排課問題。

在九年一貫課程綱要規定中，對於學生一週的上課節數有相關之限制，且每個年段的上課節數是不同的，如表 1。

表 1 國小 1~6 年級各領域學習節數分配表

年級 科目	一	二	三	四	五	六
領域學習節數	20	20	25	25	27	27
彈性學習節數	2-4	2-4	3-6	3-6	3-6	3-6
學習總節數	22-24	22-24	28-31	28-31	30-33	30-33

由於國小「彈性學習節數」並非一個固定的常數，而是一個有範圍的變數，每間學校可以透過校務會議，決定該校學生一

週上課節數為幾節，因此每間國小的一週上課節數是不盡相同的，本研究希望能製作一個適用於所有國小的排課系統，首先就必須要解決這個問題。

設定好一週上課節數後，接下來的問題便是，如何將教師 t 的任教科目 l 排入班級 c 的 p 節次，然而此一過程必須考量到許多條件，列舉如下：

一、基本原則：

1. 根據教師專長，安排任教科目。
2. 兼任行政工作的教師，應給予減課。
3. 各科目的週總節次必須符合課程綱要規定。
4. 每個班級的每一節次均只能安排一個科目。
5. 教師不能有衝堂的現象。

二、學校特殊需求：

九年一貫課程有七大學習領域及七大議題，在指導學生認識七大議題時，除融入各科教學以外，另外會在每週安排一節彈性宣導時間，此節次為全校統一，在此時間進行全校性的宣導活動。

三、各科目的獨特排課原則：

1. 國語的作文課、自然的實驗課、生活的勞作課，因教學時間較長，必須連排兩節。
2. 國語、數學等主科不得安排於下午第一節。
3. 體育課要避開用餐前與用餐後的時間，故要避免排在第一節、第四節、第五節。
4. 每天至少要安排一節以上的國語或數學，以利主科教學的連貫性。
5. 英語、社會等科目，每天不得安排一節以上。

四、教師特殊需求：

1. 兼任主任的教師依據教育處規定，必須設定一天不排課時間。
2. 在職進修的教師，必須有兩個半天的不排課時間。

3. 巡迴教師因有部分時間在他校上課，因此在排課上必須設定某時段不能在此校排課。

4. 兼課教師因可能只在學校上 1~2 天課，故其他天必須都設定為不排課時間。

五、符合現有教學場地：

某些特殊科目，如體育、電腦、音樂、美術、自然等，依各校規模之不同，會有各科專用的教室，在排課過程中，必須考量到教學場地的限制，基本上每個班級使用某個教學場地的時間必須錯開。

1.3 研究範圍

本論文的研究範圍為國小一~六年級六個年段的課表。在排課問題當中，大致上可以分為兩階段，第一階段為確定學校內的每位老師授課的班級及科目，安排的原則以教師專長及法定授課節數為主，確認了每位老師的授課班級和授課科目後，再排定教師到各班級上課的日期及節次。

第一階段的排課問題，需要考量到相當多的人為因素，每間學校的狀況都不大相同，通常是在校務會議當中提出討論，並做決策，少數則是由校長或教務主任決策之。本論文並不討論第一階段的排課問題，僅進行第二階段排課的研究。

本論文將第二階段排課問題的限制條件分為兩類，一為硬性限制（Hard constraints），一為軟性限制（Soft constraints）。硬性限制為程式中產生課表的規範，較為嚴格，程式生成的課表，必須確保它絕對不能違反硬性限制；軟性限制則主要在驗證一個課表是否滿足我們的其他需求，當程式自動生成課表，雖不會違反硬性限制，但違反軟性限制的次數可能會非常多，本研究嘗試以基因演算法，讓程式在交配及突變的運算中，慢慢課表減少違反軟性限制的次數，以期能獲得違反軟性限制最少的最佳解。

2. 文獻探討

2.1 基因演算法

基因演算法的理論源自於達爾文的進化論，生物利用不斷的演化而產生最適合生存的物種。基因演算法的具體運作方式，是以電腦程式模擬自然界中生物的演化過程。

要使用基因演算法求解，必須先將問題中的變數編碼，編碼後的數字或字串，就如同生物體內的基因，而由這些數字或字串所組成的解，則是染色體。在進行演算之前，必須先決定整個演算過程中，要運用多少組解來進行運算，這些解所形成的集合，則代表族群。族群之大小依據問題的複雜度而定。每個染色體內的基因，由於是由程式隨機產生，故具有多樣性，可使演算過程中具有隨機及多點的特性。

當決定好以上的條件後，接下來則是要定義適應函數，也就是要定義出一種方法，來判斷一個染色體是好還是壞，當染色體較優則留下，較劣則淘汰，依據自然界中「物競天擇，適者生存」的規律，讓族群中總是可以留下較好的染色體。

演算的初期，先產生一個初始族群，再利用交配、突變等運算，繁衍初始族群內的染色體，當族群內染色體的數量達到預設的最大值時停止運算，並檢視族群內各個染色體的適應值，判斷染色體的好壞，將較差的染色體淘汰。

經過不斷的搭配運算及突變運算，整個族群在經過競爭選擇、汰弱留強後，品質不良的染色體將會消失，而較好的染色體則存活下來並繼續產生下一代。不斷的重複此過程，染色體的品質將會越來越好，越來越能適應我們給予的條件，到最後即可演化出整個問題的最佳解。

此演算法最早是由 John Holland 所提出，並於 1975 年出版「Adaptation in natural and artificial systems」一書，建立基因演算法的原型，基因演算法的運作架構圖如圖 1。

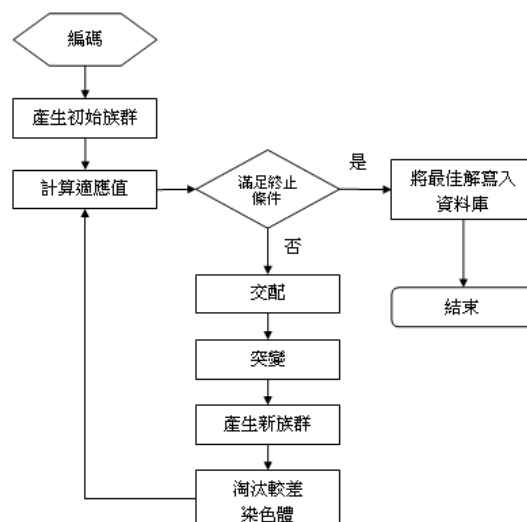


圖 1 基因演算法的運作架構

2.2 排課問題之研究發展

最早提出以電腦解決排課問題的學者是 Appleby (1961) 等人，在研究中使用簡單的經驗法則來解決小規模的排課問題。1980 年代，Selim (1982) 採用固定時段課程優先排入，再依序填入其他課程的模式，解決大學排課的問題。Dowland and Lim (1982) 則是將排課問題模組化，設計了四個模組來解決排課問題。Loo 等人 (1986) 引入二元矩陣的概念，先排指定時段的科目，然後指定科目優先權，依照優先權順序排入課表。

Burke et al. (1994) 以基因演算法，並加入專家知識庫針對排課結果提供意見，增加排課系統最佳化及使用滿意度。

王富民 (2001) 則是提出一種改良式的基因演算法，結合代理者計算 (Agent Computing) 的概念，並將其應用於大學排課問題中。

邱元泰 (2002) 利用基因演算法應用於國中排課問題，並依據國中排課時需考量的各項規則，設定四個硬性限制及七個軟性限制，利用 Matlab 撰寫演算程式證明切實可行，且在三個半小時左右的時間內排出五十八班、一百二十四位教師之學校課表。

謝正瑜(2002)利用遺傳基因演算法縮短排課時間，並以教師的觀點來定義對排課的滿意度，在多重資源的限制下，提高排課結果的滿意度。

俞慧蓮(2008)運用禁忌搜尋演算法自動關聯動態排課策略因素模組，做為自動排課系統之模型並以三所不同規模大學排課資料實際驗證模型，達到有效解決自動排課系統之實用性和區域解問題，並提昇其品質與效率。

2.3 現有排課系統之探討

以台東縣為例，目前全縣國中小統一之學務管理系統為 SFS3.1 系統(2010.09 更新版本)，SFS 學務系統是採用開放原始程式碼所架構的校務行政電腦化系統，由多人共同合作開發，而在此管理系統內就有自動排課的功能。



圖 2 SFS3.1 自動排課模組

實際使用此系統，會發現此系統並不實用，因為此系統依據使用者輸入之教師授課節數及任教科目等訊息，僅遵循幾個簡單的排課原則來進行排課，如教師不衝堂、滿足教室資源、安排預設科目等。然而實際上進行排課問題時，並非考慮這些條件即可，因此使用此排課系統所排出來的課表，會出現相當多的問題，經整理後陳述如下：

1. 某些教師在固定時間不能排課，如各處室主任、巡迴教師、兼課教師等，在此系統無法做此類設定。
2. 因隨機排課的關係，有機率造成一天內上同樣的課程兩節以上，甚至將某

些科目一個星期的課程全部排在同一天。

3. 同樣課程安排在同一天時未連排，分散於上下午或間隔 1~2 節，造成授課的連貫性差。
4. 體育課因隨機排課的關係，可能被排在剛吃完早餐或午餐之後的節次上課，或排在接近用餐的時間，對學生的健康會造成一定的影響。
5. 部分課程所需時間較長，需連排處理，如國語的作文課、自然的實驗課、生活的美勞課等，但在此系統無法自動做連排處理。

此系統雖然可以根據使用者所輸入的訊息，排出符合條件的課表，但此課表並非立即可以使用，還需要經過人工的調整，才能夠成為正式的課表。且人工調整的繁雜度，會隨著全校班級數的增加，而跟著遞增，故此系統並未真正滿足實際上國中小排課的需求，無法達到完全自動排課的效果。

2.4 綜合評論

由以往文獻可得知，目前探討排課問題的研究中，多數是以高中職以上的學校為討論的對象。然而國小的排課所要解決的問題，與高中職或大學完全不同。大學的排課，主要要解決的問題是教師可排課的時段較少，要如何將各課程安排至教師可排課的時段。國小的排課問題，主要在於要解決各類的課程，是否有安排至適合該課程上課的時段。本研究設定七個在國小排課中，需要考量到的規則，希望藉由基因演算法，盡量得出令人滿意的課表。

傳統的基因演算法，選取染色體或基因進行交配或突變，常以隨機的方式自族群或染色體中選取，然而在演算過程的後期，有缺陷的基因往往只是染色體中特定幾個基因，若是以隨機的方式決定要交配或是突變的基因，則可能會選取到已經完全沒有問題的基因，而造成時間上的浪費，降低程式執行的效率。

故本研究提出一種選取基因的方法，由上一代族群中選取最好的染色體並加以分析，判斷在此染色體中哪幾個基因已完全沒有缺陷，並設定為「禁忌名單」，下一代進行交配及突變，選取基因時則自動跳過禁忌名單的內基因，選取有缺陷的基因進行交配或突變，以期能夠以更快的效率得到最佳解。

3. 系統架構

3.1 設定限制條件

在本研究中，設定五個硬性限制及七個軟性限制，硬性限制為染色體內基因生成之依據，軟性限制則作為定義適應度函數的依據。

本研究設定的硬性限制及軟性限制詳述如下：

一、硬性限制 (Hard constraints, 在本研究中以縮寫 HC 表示)

HC1: 彈性宣導需全校或全年段統一時間

HC2: 特定科目每週連排一次(不跨中午)

HC3: 教師不得有衝堂的現象

HC4: 每個班級的每一節次均只能安排一個科目

HC5: 需滿足既有教室資源

二、軟性限制 (Soft constraints, 在本研究中以縮寫 SC 表示)

SC1: 一天之內相同科目不得排兩節以上，若安排兩節則需連排

SC2: 某些老師部分時段不可排課

SC3: 國語、數學避開下午第一節

SC4: 避免教師滿堂(一天七節課)

SC5: 體育避開第一節、第四節、第五節

SC6: 每天至少安排一節以上的國語或數學

SC7: 社會、英語每天不得安排超過一節

程式一開始的初始解為一個空的矩陣，矩陣的大小為「全校班級數(C) × 每週最多課程數(S)」。此矩陣也代表程式中的染色體，染色體內基因的生成是以班級為單位，在不違反硬性限制的條件下，填

入染色體內。然而基因的生成，若是採用完全隨機的方式，則可能會產生過多違反軟性限制的基因，而造成程式求解的效率變差，因此本研究設定六個排課時的經驗法則，透過突變運算改良程式所生成的基因，在課程尚未填滿矩陣之前，矩陣內仍有大量的空值，此時使用交配運算並無太大的意義，故僅使用突變運算生成基因，當課程已完全填滿矩陣，再加入交配運算，當以下三個條件任一條達成時，即認為已得到最佳解，程式停止運算：

- (1) 已產生完全不違反軟性限制之解。
- (2) 在 100 個世代中皆無法產生更好的解。
- (3) 演算世代達到 500 代。

3.2 定義編碼方式

教師、班級、節次、科目為排課問題中四個主要因素，在本研究中設定六個資料表來儲存相關資訊，資料表間存在互相對應的關係，各資料表功能分述如下：

1. 資料表 A: 紀錄每個年級的每個課程要安排的科目。
2. 資料表 B: 紀錄全校教師姓名及可排課的節次。
3. 資料表 C: 紀錄全校各班的每個課程由哪位老師上課。
4. 資料表 D: 紀錄每個年級一週內可排課的節次。
5. 資料表 E: 紀錄全校各班的每個課程在哪間教室上課。
6. 資料表 F: 紀錄程式所產生的可行解。

本研究在研究問題探討中曾提及，因為九年一貫課程綱要的規定，在不同的國小，學生一週的上課節數會有不同差異，因此本研究先設定一個資料表 A，用來紀錄每個年級一週上課的總節數及上課的科目，如圖 3 所示。資料表中的每一列代表一個年級，國小階段一共六個年級，故六列。每一列的每一欄中填入的數字，為課程代號，若一週國語課有 6 節，則前六欄皆填入 1。九年一貫課程綱要中對於一週節

數的規定，國小階段一週最少 23 節，最多 32 節，故資料表設定為 32 欄，若某個年級上課節數不足 32 節，未上課的欄位則填入-1，將矩陣補齊。

id	name	1	2	3	4	5	6	7	...	32	33	34	35
1	國語	1	1	1	1	1	1	1	...	-1	-1	-1	-1
2	數學	2	2	2	2	2	2	2	...	0	0	0	0
3	生活	2	2	2	2	2	2	2	...	0	0	0	0
4	自然	3	3	3	3	3	3	3	...	0	0	0	0
5	社會	3	3	3	3	3	3	3	...	0	0	0	0
6	健康	4	4	4	4	4	4	4	...	0	0	0	0
7	體育	4	4	4	4	4	4	4	...	0	0	0	0
8	綜合	4	4	4	4	4	4	4	...	0	0	0	0
9	音樂	5	5	5	5	5	5	5	...	0	0	0	0
10	英語	5	5	5	5	5	5	5	...	0	0	0	0
11	國語	6	6	6	6	6	6	6	...	0	0	0	0
12	電腦	6	6	6	6	6	6	6	...	0	0	0	0
13	美術	7	7	7	7	7	7	7	...	0	0	0	0
14	彈性	7	7	7	7	7	7	7	...	0	0	0	0

圖 3 資料表 A 及科目代號資料表之對應

之後設定教師資料表 B，如圖 4 上方所示，資料表中第一欄紀錄教師姓名，其餘欄位紀錄教師是否可以排課。國民小學課表一週安排節次最多為 7 節，每週最多上課天數為 5 天，故其餘欄數設定為 35 欄，t 則為全校教師數。在資料表中，若是某教師的某節次可以排課，則在對應的欄位填入 0，反之則填入-1。

再來建立資料表 C，在第一階段的排課中，哪位老師要教哪個年級的哪個科目，都已經確認，此資料表的主要用途，是用來紀錄第一階段的排課結果，在做第二階段的排課之前，必須在此資料表手動輸入相關資訊。

以一個六班的小校為例，資料表 C 如圖 4 下方所示，以全校的班級數決定資料表有幾列，id 欄位為班級代號，name 欄位代表班級名稱，欄位 1~32 代表課程代號，搭配資料表 A、B 做對應，如一年甲班的課程 1 填入數字 11，代表一年甲班的課程 1(國語)，授課教師為編號 11 的教師(蔡師)。

id	姓名	1	2	3	4	5	6	7	...	32	33	34	35
1	劉師	0	0	0	0	0	0	0	...	-1	-1	-1	-1
2	許師	0	0	0	0	0	0	0	...	0	0	0	0
3	王師	0	0	0	0	0	0	0	...	0	0	0	0
4	吳師	0	0	0	0	0	0	0	...	0	0	0	0
5	趙師	0	0	0	0	0	0	0	...	0	0	0	0
6	錢師	0	0	0	0	0	0	0	...	0	0	0	0
7	孫師	0	0	0	0	0	0	0	...	0	0	0	0
8	李師	0	0	0	0	0	0	0	...	0	0	0	0
9	郭師	-1	-1	-1	-1	-1	-1	-1	...	0	0	0	0
10	陳師	0	0	0	0	0	0	0	...	0	0	0	0
11	蔡師	0	0	0	0	0	0	0	...	0	0	0	0

id	班級	1	2	3	4	5	...	27	28	29	30	31	32
1	一甲	11	11	11	11	11	...	-1	-1	-1	-1	-1	-1
2	二甲	10	10	10	10	10	...	-1	-1	-1	-1	-1	-1
3	三甲	5	5	5	5	5	...	9	5	1	-1	-1	-1
4	四甲	7	7	7	7	7	...	9	7	1	-1	-1	-1
5	五甲	6	6	6	6	6	...	9	6	1	3	3	6
6	六甲	2	2	2	2	2	...	9	2	1	8	8	2

圖 4 資料表 B(教師可排課節次)及資料表 C(各班各個課程的任課老師)之對應

接下來設定資料表 D，國小一週課程分為五日，每日七節，本研究將所有節次從 1~35 做編號，如圖 5。然而並非每節課都是上課時間，例如星期一下午低年級不上課，星期五下午中低年級皆不上課等。資料表 D 可以設定哪些年級在哪些節次是不上課的，在資料表中，欄位 1~35 代表上課節次，填入-1 代表此節次該年級是不上課，因版面關係，部分課程以...省略。此資料表內數字 0 的出現次數總合，代表全校一週的總授課節數。

id	班級	1	2	3	4	5	6	7	...	31	32	33	34	35
1	一甲	0	0	0	0	-1	-1	-1	...	0	0	-1	-1	-1
2	二甲	0	0	0	0	-1	-1	-1	...	0	0	-1	-1	-1
3	三甲	0	0	0	0	0	0	0	...	0	0	-1	-1	-1
4	四甲	0	0	0	0	0	0	0	...	0	0	-1	-1	-1
5	五甲	0	0	0	0	0	0	0	...	0	0	0	0	0
6	六甲	0	0	0	0	0	0	0	...	0	0	0	0	0

	Mon	Tues	Wed	Thu	Fri
一	1	8	15	22	29
二	2	9	16	23	30
三	3	10	17	24	31
四	4	11	18	25	32
五	5	12	19	26	33
六	6	13	20	27	34
七	7	14	21	28	35

圖 5 資料表 D(各班級一週可排課節次)及一週各天節次代號表之對應

為了滿足教室資源，接下來設定資料表 E，由於國小各班級均有自己的班級教室，很少有甲班的課程跑到乙班教室去上課的狀況，故某課程若在自己班級教室上課，本研究在資料表 E 中將其設定為 0，若在

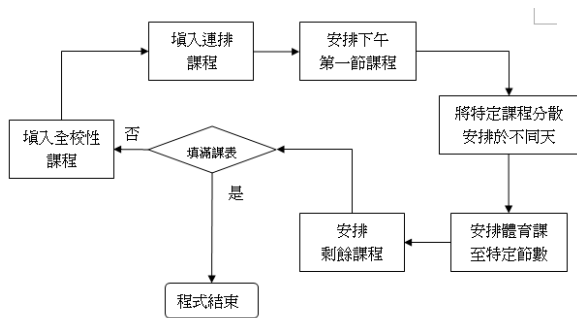


圖 12 突變運算流程圖

3.5 定義交配運算(Crossover)

一、取樣方法

1. 從一開始 50 個可行解中，隨機選取 2 個不同的可行解
2. 以亂數方法決定交配列（同年級的課表才可以交配）

二、交配步驟

1. 將連排科目、全校性課程直接複製到子代
2. 將父代 B 的節次，在不違反子代已有此節次及造成教師衝堂的情況下，填入至子代。
3. 找出目前仍未填入的節次及目前仍未安排到的科目代號，並找出這些科目的任教老師是誰。算出這些任教老師，在這些節次的空堂數，之後遵循空堂數少的老師先排課，多的後排課的原則，在不衝堂的條件下將課表排滿。

3.6 定義適應值

本研究以可行解違反軟性限制之次數定義其適應值，並以加權分數判斷可行解的優劣，因 SC1~SC7 各限制式在排課考量的重要性不同，故給予 SC1、SC4 較高的權重。此外，由於本研究之初始解為空的矩陣，矩陣內的元素皆為 0，之後才倚賴突變運算將課表填滿，因此在程式尚未將課表填滿之前，矩陣內若有元素為 0，則給予最高的權重 10，以便在基因演算法運作的過程中，最先將此類型的問題解決。

在本研究的設定中，可行解的加權分數越高者表示此可行解越差，越低者越好，當程式生成加權分數為零的可行解時，表示已產生最佳解。

	限制式敘述	權重
SC1	一天之內相同科目不得排兩節以上，若安排兩節則需連排	5
SC2	某些老師部分時段不可排課	1
SC3	國語、數學避開下午第一節	1
SC4	避免教師滿堂(一天七節課)	2
SC5	體育避開第一節、第四節、第五節	1
SC6	每天至少安排一節以上的國語或數學	1
SC7	社會、英語每天不得安排超過一節	1
Z	課表矩陣內有元素為 0	10

圖 14 本研究之限制式敘述及權重

3.7 定義淘汰規則

本研究在產生初始解的步驟，產生個數設定為 50 個，程式自動停止，且進入交配及突變運算，產生新的子代。當個數達到 100 個時，將較差的 50 個刪除，回到交配及突變運算繼續產生新的子代，如此週而復始，直到程式產生最佳解。

在基因演算法的運作過程中，最大的問題是要避免程式落入局部最佳解，由於在交配運算及突變運算中，子代會承襲父代大部分的基因，故在數代之後，交配池內的染色體會變得非常相似，而導致無法

原始父代交配列 (上=父代 A 下=父代 B)

3	4	28	22	32	31	8	23	1	35	29	9	17	18	6	2	26	25	11	24	30	33	10	5	34	7	27	13	15	14	16	12
31	32	4	23	3	9	24	8	22	11	17	35	27	28	18	13	29	14	34	2	6	5	25	7	30	16	15	33	1	26	10	12

交配後子代交配列演變情形

步驟 1	3	4											17	18																		12
步驟 2	3	4	23	9	24	8	22	11	35	17	18				14	34	2	6	5	25	7	30			33	1	26			12		
步驟 3	3	4	15	23	29	9	24	8	22	11	31	35	17	18	16	27	28	14	34	2	6	5	25	7	30	10	13	33	1	26	19	12

圖 13 交配運算三步驟示意圖

經過以上三個步驟，仍有可能沒辦法排滿課表（因教師衝堂之故），所以在查詢資料表 D 後，檢查排入的課程是否跟全校一週上課總節數相同，若是相同代表已排滿課表，此時才將資料寫進資料庫 F。

產生更好的子代。為了要維持交配池中基因的多樣性，在以往文獻中是以投入大量突變數來解決這個問題[2]，然而本研究提出一種特殊的淘汰規則，來維持基因的多樣性。

程式執行步驟如下：

1. 當交配池中可行解達到 100 個時，開始計算各個可行解違反限制式的加權分數，並作統計及分組，如圖 13。
2. 找出個數超過 10 個的組別，刪除至剩下 10 個。在此階段被刪除的可行解，總數設定為 N。
3. 將剩餘的可行解依照加權分數作排序，刪除最差的(50-N)個。

加權分數	0	1	2	3	4	5	6	7	...
個數	0	0	3	1	(12)	6	(13)	5	...

圖 15 本研究可行解淘汰規則示意圖

3.8 定義禁忌名單

傳統的基因演算法，選取染色體或基因進行交配或突變，常以隨機的方式自族群或染色體中選取，然而在演算過程的後期，有缺陷的基因往往只是染色體中特定幾個基因，若是以隨機的方式決定要交配或是突變的基因，則可能會選取到已經完全沒有問題的基因，而造成時間上的浪費，降低程式執行的效率。

故本研究設定，當每一代的染色體生成後，將這一代最好的染色體抽取出來分析其違反軟性限制的基因究竟位於何處，並將完全不違反軟性限制的基因位置標記出來，設定為禁忌名單，在下一代染色體選取基因進行交配及突變運算時，自動略過這些位於禁忌名單中的基因。

4. 系統驗證

4.1 六班學校之排課實驗

本小節首先以全校只有六個班級之小學校進行排課問題實例驗證。該校每一個年級只有一個班級，共六個學校，全校教師人數為 11 人，排課問題的班級、科目、

教師代碼如下：

課程代號	國語										數學					生活					體育		綜合		彈性		無此課程							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
班級	教師	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
一年甲班		11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	8	8	8	8	8	11	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
二年甲班		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	4	3	3	3	10	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	

課程代號	國語										數學					自然		社會		體育		綜合		音樂		英語		電腦		美術		彈性	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
班級	教師	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
三年甲班		5	5	5	5	5	5	5	5	5	5	5	5	5	3	3	3	8	8	8	8	8	5	5	5	5	9	9	5	1	4	4	5
四年甲班		7	7	7	7	7	7	7	7	7	7	7	7	1	1	1	4	4	4	4	4	4	7	7	7	7	9	9	7	1	9	9	7
五年甲班		6	6	6	6	6	6	6	6	6	6	6	6	4	4	4	4	4	4	4	4	4	6	6	6	6	9	9	6	1	3	3	6
六年甲班		2	2	2	2	2	2	2	2	2	2	2	2	8	8	8	8	8	8	8	8	2	2	2	1	9	9	2	1	8	8	2	

1	邱○幃	7	歐○玲
2	陳○宇	8	豐○洲
3	鍾○惠	9	郭○蓉
4	蔡○德	10	谷○萍
5	陳○婷	11	李○珠
6	張○卿		

此排課問題之限制條件為本研究設定之四個硬性限制及八個軟性限制，各限制詳述如下：

一、硬性限制 (Hard constraints, 以縮寫 HC 表示)

HC1：彈性宣導需全校或全年段統一時間，該校彈性宣導時間訂於星期二下午第一節。

HC2：特定科目每週連排一次(不跨中午)，國語、中高年級自然、低年級生活每週連排一次。

HC3：教師不得有衝堂的現象

HC4：每個班級的每一節次均只能安排一個科目

HC5：需滿足既有教室資源，該校電腦教室及自然教室各一間，各班上課時間不得衝堂。

二、軟性限制 (Soft constraints, 以縮寫 SC 表示)

SC1：一天之內相同科目不得排兩節以上，若安排兩節則需連排

SC2：某些老師部分時段不可排課

1. 英語巡迴教師郭○蓉星期二、星期五及星期四早上在其他國小上課，這些時段不能排課。
2. 教師邱○幃、谷○萍、陳○婷於研究所進修中，星期五下午不排課。
3. 教師鍾○惠，因私人因素，下午第一

節及星期五下午不排課。

SC3：國語、數學避開下午第一節

SC4：避免教師滿堂(一天七節課)

SC5：體育避開第一節、第四節、第五節

SC6：每天至少安排一節以上的國語或數學

SC7：社會、英語每天不得安排超過一節

本研究測試「交配率」、「突變率」、「禁忌名單」對於求解效益之影響，在參數設定方面，族群最大值設定為 100 條染色體，每一組參數皆執行 100 次，在以下三個條件任一條成立時，停止運算：

1. 已產生完全不違反軟性限制之解。
2. 在 100 個世代中皆無法產生更好的解。
3. 演算世代達到 500 代。

在分析各組數據時，除了完成任務之世代數要盡量少、執行速度盡量快之外，也需計算當程式停止時，產生解之懲罰函數，以判斷各組產生之解是否優良；另外各組世代數的變異數，亦是觀察之重點，變異數越小，代表每次求得解的世代數差異越小，使用該組數據求解時，求解所花費的時間越穩定。

所有演算程式均以 ASP 撰寫，並於 AMD Athlon II K125 (1.7G)、記憶體 2G 之 Notebook 上執行運算。

表 2 交配次數 N，突變次數 50-N，不使用禁忌名單

N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
40	18	262	82.5	2531.87	0.92	76.25	0.44	85/100	0
30	22	193	76.1	1734.41	1.06	80.61	0.33	87/100	0
20	22	209	64.2	1337.17	1.18	75.77	3.49	91/100	1
10	9	165	55.8	866.01	1.35	75.05	0.13	95/100	0
0	20	151	59.8	1102.81	1.37	82.03	0.21	95/100	0

以上 500 次實驗，皆不使用禁忌名單，比較「交配率」及「突變率」對於求解效益之影響。經由實驗結果(表 2)可得知以下幾點結論：

1. 由於突變運算加入了許多經驗法則，因此突變運算越多次，每代執行的時間會增加。

2. 交配運算由於未加入任何經驗法則，故交配運算越多次，求解的效益越差；且產生的解平均懲罰函數越高，代表使用交配運算越多次，越有機率產生素質較差的解。
3. N=20 這組由於有一次運算落入局部最佳解，導致第六個班級的課表排不出來，全部為空值，違反了硬性限制，成為最差的一組。
4. 以實驗數據來看，交配次數 0 和交配次數 10 這兩組為較佳的參數設定。

接下來測試使用禁忌名單是否可使程式求解的效益增加，由各組突變運算中，選擇其中 10 次，根據禁忌名單來選取染色體及基因。測試結果如表三所示。

表 3 交配次數 N，突變次數 50-N，其中 10 次使用禁忌名單

N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
40	16	171	48.0	784.79	0.98	47.1	0.03	97/100	0
30	10	91	38.0	288.12	1.14	43.3	0	100/100	0
20	9	108	37.9	322.67	1.25	47.4	0	100/100	0
10	7	138	38.6	462.73	1.36	52.3	3.31	97/100	1
0	11	159	33.9	417.29	1.51	51.0	0	100/100	0

以上 500 次實驗，加入了 10 次使用禁忌名單的策略，測試加入了禁忌名單後，對於求解效益之影響。分析實驗結果並與表 2 做比較，可得知以下幾點結論：

1. 加入了禁忌名單的使用，雖然每代所需的時間有略為上升，但由於世代數降低，故求得最佳解的平均時間大幅下降。
2. 加入了禁忌名單的使用，得到最佳解的世代數、世代變異數、平均懲罰函數皆明顯降低，表示禁忌名單的使用，對於求解的效率、穩定性及最佳解的品質皆有明顯的幫助。
3. 以實驗數據來看，交配次數 0、20、30 這三組參數設定都有不錯的表現。

確認使用禁忌名單有助於求解的效率、穩定性及品質後，再來測試使用禁忌名單次數多寡，對於求解效益的影響，本研究以 20 次、30 次、40 次禁忌名單的使

用，對各組數據進行實驗。測試結果如表 4、5、6 所示。

表 4 交配次數 N，突變次數 50-N，其中 20 次使用禁忌名單

N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
40	9	146	42.8	858.60	1.00	42.9	3.33	95/100	1
30	9	131	32.4	328.65	1.15	37.3	0.01	99/100	0
20	10	119	29.5	232.99	1.30	38.4	0	100/100	0
10	4	116	25.8	247.13	1.56	40.2	0	100/100	0
0	5	70	26.3	166.81	1.56	41.2	0	100/100	0

表 5 交配次數 N，突變次數 50-N，其中 30 次使用禁忌名單

N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
40	10	125	40.4	401.90	0.97	39.4	0.01	99/100	0
30	5	154	28.8	331.25	1.14	32.9	0.01	99/100	0
20	8	112	25.3	220.71	1.33	33.6	0	100/100	0
10	5	120	25.2	308.57	1.49	37.5	6.5	98/100	2
0	5	82	24.2	207.66	1.62	39.1	0	100/100	0

表 6 交配次數 N，突變次數 50-N，其中 40 次使用禁忌名單

N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
40	11	117	41.8	667.49	0.99	41.6	3.25	99/100	1
30	5	165	38.8	898.18	1.20	46.4	0.16	96/100	0
20	4	120	35.1	542.15	1.38	48.4	3.31	97/100	1
10	4	123	43.5	581.20	1.54	43.5	0.11	97/100	0
0	3	120	22.9	369.91	1.56	35.8	6.5	98/100	2

綜合以上 2500 次實驗，可得知禁忌名單的使用確實有助於求解的效益，但使用次數以 30 次左右為限，使用次數 40 次反而會令程式容易落入局部最佳解，無論是求解速度或是最佳解的品質皆大幅下降。在本次實驗中有八組數據完全沒有產生違反軟性限制的解，茲整理如表 7，並以程式跑出最佳解的平均時間由小至大排序：

表 7 交配次數 N，突變次數 50-N，使用禁忌名單 T 次

T	N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
30	20	8	112	25.3	220.71	1.33	33.6	0	100/100	0
20	20	10	119	29.5	232.99	1.3	38.4	0	100/100	0
30	0	5	82	24.2	207.66	1.62	39.1	0	100/100	0
20	10	4	116	25.8	247.13	1.56	40.2	0	100/100	0
20	0	5	70	26.3	166.81	1.56	41.2	0	100/100	0
10	30	10	91	38	288.12	1.14	43.3	0	100/100	0
10	20	9	108	37.9	322.67	1.25	47.4	0	100/100	0
10	0	11	139	33.9	417.29	1.51	51	0	100/100	0

以求解速度來看，交配次數 20 次，使用禁忌名單 30 次這組是最快的，且超過第二名甚多。就穩定性而言，不使用交配運算，使用禁忌名單 20 次這組最穩定。

此八組數據在 100 次的運算之中皆無違反軟性限制，表示在 25 組數據中是較佳的選擇，故本實驗使用此八組數據進行接下來更多班級的排課問題，測試各類參數在不同班級數的排課問題中，是否會有差異。

4.2 十六班學校之排課實驗

本小節模擬一個全校 16 班，31 位教師之排課問題。排課問題的班級、科目、教師代碼如下：

課程代號	國語					數學					生活					健康	綜合	關	彈性	無此課程														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
一年智班	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	3	1	1	2	1	1	1	1	1	1	1	1	1	1	1		
一年仁班	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	7	7	5	5	6	5	1	1	1	1	1	1	1	1	1	1	1	
二年智班	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	9	9	8	8	2	8	1	1	1	1	1	1	1	1	1	1	1	
二年仁班	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9	10	10	6	10	1	1	1	1	1	1	1	1	1	1	1	1

課程代號	國語					數學					自然	社會	健康	體育	綜合	音樂	英語	電腦	美術	彈性	無此課程															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32				
三年智班	11	11	11	11	11	11	11	11	11	11	11	7	7	7	7	11	11	11	11	13	13	11	11	11	4	9	2	7	12	12	11	1	1	1		
三年仁班	14	14	14	14	14	14	14	14	14	14	14	7	7	7	14	14	14	14	16	16	14	14	14	4	9	6	7	15	15	14	1	1	1	1		
四年智班	17	17	17	17	17	17	17	17	17	17	17	16	16	16	16	17	17	17	19	19	17	17	17	4	9	18	17	17	17	1	1	1	1	1		
四年仁班	20	20	20	20	20	20	20	20	20	20	20	16	16	16	16	20	20	20	22	22	23	23	23	20	20	4	9	21	7	20	20	1	1	1	1	
四年智班	22	22	22	22	22	22	22	22	22	22	22	16	16	16	16	22	22	22	24	24	22	22	22	4	9	2	7	22	22	1	1	1	1	1	1	
四年仁班	24	24	24	24	24	24	24	24	24	24	24	16	16	16	16	25	25	25	25	24	24	24	24	4	9	6	24	26	26	24	1	1	1	1	1	1

課程代號	國語					數學					自然	社會	健康	體育	綜合	音樂	英語	電腦	美術	彈性	無此課程																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32						
五年智班	27	27	27	27	27	27	27	27	27	27	27	28	28	28	4	4	4	27	16	16	27	27	27	4	9	9	18	27	27	1	1	1	1	1	1	1		
五年仁班	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	28	28	4	4	4	12	13	13	12	12	12	4	9	9	21	7	12	12	1	1	1	
五年智班	29	29	29	29	29	29	29	29	29	29	29	30	30	30	30	30	13	13	29	24	24	29	29	4	9	9	2	7	29	29	1	1	1	1	1	1	1	
五年仁班	33	33	33	33	33	33	33	33	33	33	33	23	23	23	23	30	30	13	13	35	23	23	23	23	4	9	9	6	23	29	29	1	1	1	1	1	1	
六年智班	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	31	31	25	25	25	26	23	23	26	26	4	9	9	2	7	26	26	1	1	1	1	1	
六年仁班	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	31	31	31	31	31	31	31	19	19	15	15	15	4	9	9	6	7	15	15	1	1	1

這間學校低、中、高三個年段一週內的上課節數皆不同，低年級 23 節、中年級 29 節、高年級 32 節。在教室資源方面，有兩間自然科教室及一間電腦教室。教師不排課時段設定教務主任週二不排課，學務主任週三不排課，總務主任週四不排課。全校統一宣導時間訂於星期五上午第一節。其餘設定與 4.1 節之實驗參數相同。

本次實驗所設定的參數，參考 4.1 節中運算結果較佳的八組數據，測試此八組數據在 16 班，31 位老師的排課問題中，求解

的效率與品質是否有所改變。測試結果如表 8 所示。

表 8 交配次數 N，突變次數 50-N，使用禁忌名單 T 次(16 班)

T	N	最少世代	最多世代	平均世代	世代變異數	每代時間(秒)	平均時間(秒)	平均懲罰函數	完全不違反軟性限制	違反硬性限制
30	20	24	100	50.9	266.49	1.84	93.8	0	100/100	0
20	20	21	101	52.4	231.53	1.83	95.9	0	100/100	0
30	0	19	138	45.4	424.72	1.89	85.6	0.02	98/100	0
20	10	25	148	56.0	397.30	1.85	103.8	0.01	99/100	0
20	0	30	156	53.8	324.02	1.86	100.2	0	100/100	0
10	30	41	147	77.8	472.59	1.78	138.3	0	100/100	0
10	20	35	136	74.7	400.57	1.81	135.5	0	100/100	0
10	0	36	191	73.9	605.47	1.84	136.2	0	100/100	0

使用此八組數據解決 16 班 31 位教師的排課問題，在 800 次的運算之中，僅有 3 次違反了軟性限制，機率僅有 0.375%，表示這八組數據幾乎都可以完美的完成 16 班的排課問題。其中交配次數 20 次，使用 30 次禁忌名單這組仍為求解速度最快的一組，且實驗至今仍未出現任何一次違反軟性限制的解。

與 6 班的排課問題相比，16 班的排課問題求得最佳解的平均世代數大約增加 2 倍，平均時間增加 2.67 倍。

5. 結論及未來研究方向

排課問題之困難在於其限制條件非常複雜，以往許多文獻中，看中了基因演算法強大的搜尋能力，使用它來求解。然而若是毫無策略的隨機搜尋，程式求解的效率往往相當差，故本研究在突變運算中，加入了六個排課時常用到的經驗法則，讓程式生成的基因違反軟性限制的次數能夠盡量降低，以增加求解的效率。

此外，為了避免在演算的後期，因為隨機選取而選取到已經沒問題的基因，造成程式進行過多不必要的運算，本研究提出禁忌名單的使用，讓程式自動搜尋在染色體中，哪些基因是有問題，哪些是沒問題的，選取時跳過已經沒問題的基因，選取有問題的基因進行交配或突變，以期能夠迅速解決基因內部的問題，增加求解的效率。經由實驗驗證，此方法對於程式的求解，確實有顯著的幫助。

一個排課系統的開發，最困難的地方就是很難滿足每一間學校的需求，每間學校由於地理、人文環境的不同，對於課程安排會有相當大的差異，針對本研究所設計的排課系統，未能完備之處，提出三項未來研究方向：

1. 特殊班級的編排

本研究之排課系統，以每個年級只有一班，全校只有六個班級之小校為系統評估對象，該校無特殊班級之需求。然而在其他學校，有美術班、音樂班、體育班等特殊班級之編制，其上課之課程與一般班級有些許的差異，在本研究中並未討論到這些問題，未來可針對這方面的問題進行研究。

2. 一個課程可由兩位以上的教師授課

本研究在設定資料表時，一個課程僅能填入一位教師的姓名，然而這樣的設定不能滿足所有學校的需求。以台東的某些國小為例，鄉土語言課程可能一個班級會分成兩組，一組選擇閩南語，一組選擇原住民語，分別由不同的兩位老師授課。

3. 演算法運作的程式改以 Java Script 等語言撰寫較佳

本研究所開發的系統，不管是在資料庫的輸出輸入，或是演算法的運作，都是以 ASP 為主。然而 ASP 是一種 Server 端的語言，因此在程式演算的過程中，Server 端 Cpu 的使用率會上升至 100%，資源幾乎全部會被佔滿，這會導致同一時間內，幾乎只能讓一個使用者使用排課系統，其他使用者若要使用系統，資源分散的結果，可能導致程式運作非常慢，甚至導致 Server 當機。

因此較佳的模式應該是使用 ASP 管理資料庫，由資料庫中輸入及取出排課資料，接下來交由 Client 端的 Js 程式進行排課，等排課完成後再將結果傳回 Server 端。

參考文獻

中文部分

- [1] 王富民，『基因演算法於排課問題上之研究』，國立台灣師範大學資訊教

- 育研究所碩士論文，民90年。
- [2] 邱元泰，『遺傳演算法在排課問題之應用』，國立中正大學數學研究所碩士論文，民91年。
- [3] 林師檀，『禁忌搜尋法與遺傳演算法混合模式在地下水復育優選問題之應用』，國立中興大學環境工程學系碩士論文，民91年。
- [4] 張士宇，『基因演算法應用於大學排課最佳化系統建置之研究』，國立台北科技大學土木與防災研究所碩士論文，民98年。
- [5] 許武義，『網頁式排課管理系統』，暨南國際大學資訊管理研究所碩士論文，民89年。
- [6] 俞慧蓮，『以動態排課策略因素及禁忌搜尋法求解自動排課問題』，民97年。
- [7] 謝正瑜，『利用遺傳基因演算法進行排班最佳化之研究-以大學排課為例』，華梵大學資訊管理學系碩士班碩士論文，民92年。
- [6] S.M. Selim, "An algorithm for constructing a University faculty timetable," *Computer & Education*, 6(4), 1982, pp. 323-332.
- [7] S.M. Selim, "An algorithm for producing course and lecture timetables," *Computer & Education*, 7(2), 1983, pp. 101-108.

英文部分

- [1] E. Burke, D. Elliman and R. Weare, "A Genetic Algorithm Based University Timetabling System," *Proceedings of the 2nd East-West Int. Conf. on Computer Technologies in Education*, 1994, pp. 35-40.
- [2] W.B. Dowsland and S. Lim, "Computer aided school timetabling – part 1: the history of computerised timetabling," *Compute Education*, 1982, pp. 22-23.
- [3] W.B. Dowsland and S. Lim, "Computer aided school timetabling – part 2: the micro-computer for school timetabling," *Computer & Education*, 1982, pp. 2-4.
- [4] S. Even, A. Itai and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems," 16th IEEE Annual Symposium on Foundations of Computer Science, 1975, pp. 184-193.
- [5] E.H. Loo, T.N. Goh and H.L. Ong, "A heuristic approach to scheduling university timetables," *Computer &*