

應用基因演算法於大專院校教師派課問題之研究

游峰碩

崑山科技大學資訊管理系 助理教授
e-mail: fsyu@mail.ksu.edu.tw

王子夏

崑山科技大學資訊管理系 研究生
e-mail: u9011023@mail.ndhu.edu.tw

摘要

本研究運用基因演算法探討大專院校教師派課問題(Faculty-Course Assignment Problem, FCAP)。教師派課問題不同於一般文獻上所探討的排課問題(Timetabling Problem, TP)。所謂的排課問題是指如何將下學期所欲開設之課程根據不同的資源限制指派至不同的時段，重點在於各項資源與時段之排程。而本研究探討的派課問題是指一個科系如何將下學期所欲開設之課程分派給系上之授課老師；研究的重點著重在派課時如何兼顧到「教」與「學」兩方面：在「教」的方面，研究重點在於派課的公平性、教師對於派課結果的滿意度；在「學」的方面，研究重點在於如何將適才適所的教師指派至相關課程以提升學生對於教學的滿意度。

關鍵詞：基因演算法、派課問題、排課問題。

Abstract

In this research, we study the faculty course assignment problem (FCAP). The FCAP is different from the timetabling problem (TP) discussed in the literature. The timetabling problems deal with how to assign courses in a semester to certain time-slots satisfying different resources constraints, and the faculty-course assignment problems deal with how to assign course to teachers regarding to teachers' preference, the fair distribution of teaching load and teachers' qualifications.

From literature, FCAP is always been considered as constraints in TP study or not at all, only a few papers study it independently. In this research, we apply genetic algorithm to search the optimal solution for FCAP. Using the study of timetabling problem as a basis, and focusing on the robustness and global search mechanism provided by GA as the research focus. The goal is to maximize the course assignment and teaching

assignment satisfactory. Traditionally, genetic coding constantly involves repair mechanism after certain genetic operation which greatly degrades the performance of genetic algorithm. In this study, we propose a coding schema to prevent it so as to enhance the performance of the process.

Keywords: genetic algorithms, timetabling problem, course assignment.

1. 前言

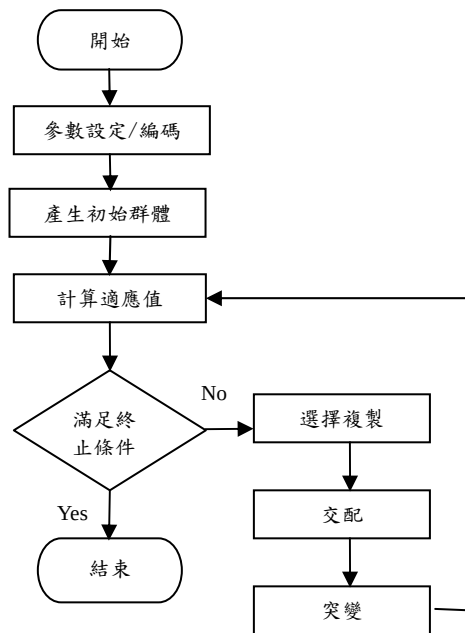
教師派課問題 (Faculty-Course Assignment Problem) 可以視為一般大專院校排課問題 (Timetabling Problem) 的特殊例子，而排課問題則可視為排程問題 (Scheduling Problem) 的一個特殊例子。排課問題是指如何將下學期所欲開設之課程根據教師、教室等不同的資源限制指派至不同的時段，其重點在於各項資源與時段之排程。而派課問題是指一個科系如何將下學期所欲開設之課程分派給系上之授課老師。派課問題的研究重點在於派課的公平性以及教師對於派課結果的滿意度。

教師派課問題的特性在於一般的大專院校擁有許多不同的學制，這些學制包含有二技、四技、進修部、在職專班、假日班等等，而各學制所開設之課程會因為學制的不同而種類繁多，其所涵蓋的課程科目種類也不盡相同。除此之外，每位教師所欲教授的課程也會因為教師的專長而有所不同。再者，在大專院校，甚至是不同科系在課程指派上亦有其相關規則必須符合，例如排課時需考慮到教師興趣、行政兼職、教師級別、授課鐘點數、兼課節數上限等等，所以在派課問題上會有不同的限制要求，這使得教師派課問題變得相當複雜。傳統的派課流程均是以人工作業的方式來完成，處理各種不同的派課限制條件。傳統的方式不僅費時費力，所分派之結果也可能與教師之滿意度相去甚遠，所造成之不公平性也極易

影響到組織的合諧。

另外，在排課的過程中，對於已派定的課表萬一有任何教師因為某些因素無法履行授課責任而需調整課表時，常會有牽一髮而動全局的困擾。上述種種理由皆導致傳統排課方式相當缺乏效率。在所發表之排課問題相關的文獻中，大多數研究專注於如何排出一個可行的課程時程表而忽略了授課教師的個人授課時段以及被指派的課程偏好，更少有研究探討如何在派課問題上給出滿意的答案。

本研究應用基因演算法於派課問題的研究上，其目的是協助教學單位在執行派課作業的過程中產生最佳的資源分配以提高派課的品質。基因演算法(Genetic Algorithm, GA)是由美國密西根大學教授 John H. Holland 及其同事、學生於 70 年代左右通過對生物進化過程進行模擬所得出的一種全新的機率最佳化方法，其概念與達爾文進化論的「物競天擇」觀念類似。生物的進化過程本身就是一個關於環境適應性的搜尋過程。因此，基因演算法是一種模擬生物界自然選擇與進化機制所發展出之高度平行、隨機、自適應的隨機搜尋法。基因演算法透過演化的機制隨機地逐步淘汰較差的個體，保留較好的個體，而被保留的個體使用交配(crossover)，突變(mutation)等遺傳運算子來產生優良的下一代，經過世代演化機制達到最佳化的目標。使用基因演算法，首先必須對問題的解進行編碼成為所謂的「染色體」；常見的編碼方式為採用二位元編碼或是文字編碼二種方式。一個染色體稱為個體(individual)，一個個體代表所探討問題的一個解；而一個染色體由許多個基因(gene)所組成，不同的基因排列代表著此一染色體的不同特徵；每一個染色體會經過評估過程來計算該染色體對所處環境的適應度，適應度越高的染色體其相對映問題解的品質越好。在進化的過程中，整體族群(population)一般會經歷「選擇」、「交配」、以及「突變」等過程。選擇運算子的主要目的在於決定哪些染色體可以存活至下一代；交配運算子的目的則在將兩個父代的染色體做部份資訊的交換以產生新的子代，其作用就好比在解空間中作大幅度的跳躍搜尋；突變運算子的主要目的在防止演化過程落入局部區域解。基因演算法基本的運作流程如下圖所示：



以下各章節內容摘要分述如下。第二節對於排課問題以及派課問題做相關的文獻探討。第三節描述本研究之派課問題並且建立所需要之數學模型。第四節描述本研究所採用之基因編碼方式以及相關的遺傳運算子的執行步驟。第五節對本研究之實驗結果做一歸納與結論。

2. 文獻探討

2.1 排課問題

排課問題的研究起步於 60 年代 [14]。每一個排課問題會因為所討論之學校型態 (例如大學排課，或是高中排課)，以及問題本身所考量之限制因素的不同而有所不同。因此，對於許多學者所提出之排課問題的研究，每一個都有其獨特的考量。A. Schaerf 將排課問題依據應用之領域分成以下三大類[22]：

- School timetabling：高中排課問題
- Course timetabling：大專院校排課問題
- Examination timetabling：考試排程問題

基本上，大專院校的排課問題遠比高中排課問題來的複雜許多。這兩者之間的最大差異在於學生選課的自由度、教師授課課程的指派、上課教室的固定性與否、以及課程的性質(必修課、選修課)。大專院校的排課問題除了牽涉到如何將教師指派到其所開的課程外，由於學生上課的教室也不固定，與高中制度中學生並非自由選課，且都在同一間教室上課有很

大的不同。除此之外，大專院校一個科系可能同時包含有好幾種不同的學制，再加上日間部，進修部以及假日班所涵蓋之課程不盡相同，因此，排課問題相形之下較高中排課問題更加複雜許多。

Bardadym 將大專院校的排課問題依功能性細分為下列五大類[3]：

- faculty timetabling：排定課程予合格教師。
- class-teacher timetabling：以班級為最小的排課單位排定課程。
- course scheduling：以單一學生為最小的排課單位排定課程。
- examination scheduling：排定學生考試時段使得同一個時段不會同時舉行兩個考試。
- classroom assignment：在排定教師課程之後，排定教師-課程予教室。

利用此分類法，本研究题目的範圍可以歸類於教師派課的排程問題 (faculty course timetabling problem)。

除了依據應用領域來分類之外，如果以排課問題所要追求的目標來分類，排課問題可以分為搜尋問題 (search problem)或是最佳化問題 (optimization problem)。對於排課，如果目的在於尋求滿足所有限制式之最佳解，此類排課問題可以被歸類為搜尋問題。另一方面，如果所要求的是一個滿足所有硬性限制的課表，並且目的是在尋找滿足目標函數(由軟性限制組成)之極大化或極小化的話，則此類問題被歸類為最佳化問題。然而即使一個排課問題歸類於最佳化的問題，其研究過程也是應用最佳化的技術於搜尋問題上。因此，文獻上所探討的排課問題均可以看成是搜尋問題的一種。不過，不論是以搜尋的觀點或是最佳化的觀點，幾乎所有不同形勢之排課問題，即使在最簡單的模式下，在計算上都是 NP-complete[5]。因此，當問題的範圍變大時，幾乎不可能找得出最佳解或是證明所得出之可行解為問題的最佳解。在排課問題的研究上，只有少數規模較小(課程數目少於 10)的排課模型可以求出最佳解。Pearl[21]指出對於 NP-complete 的問題只有啟發式方法(heuristic method) 是可行的，但即使如此，也無法保證能夠搜尋到問題的最佳解。除了計算上的困難之外，排課問題的困難還包括[12]：

- 對於大多數現實生活中的排課問題其解空間可能大到無法有效地應用 brute force 方式或是間接的搜尋方法。
- 如果嘗試著去找出一致性的演算法可能是不可行的，因為對於特定的排課問題上通常存在著某些特殊的限制式要求。
- 真實生活中的排課問題常常牽涉到無法精確表達或描述的限制式。

綜合以上所述理由，可見排課問題的複雜。

早期的研究，將排課問題歸屬於作業研究的領域，而研究方法也著重在應用數學規劃的技術來尋找最佳解。另外，由於排課問題是 NP-Complete 的問題，所以研究方面也多在比較排程的時間效率上，這是因為早期的研究受限於電腦硬體的能力限制。近來年，由於電腦在硬體以及軟體方面的進步，傳統的整數規劃，混合整數規劃(Mixed IP)方法再次變成研究組合問題的有效工具[18]，並且也被應用在求解某些特定條件下之排課的最佳化問題上[11]。近年來，人工智慧領域上所發展出來的技術，例如類神經網路(neural network)、基因演算法 (Genetic Algorithms)、Tabu 搜尋法 (Tabu search)、限制式滿足 (constraint satisfaction)、限制式邏輯規劃 (constraint logic programming) 等，漸漸被應用在解決排課問題的研究上，並且取得不錯的成果。

Akkoyunlu [2]，Csiman 和 Gotlieb [7] 對於排課問題提出了一個簡單的線性規劃模型 (TTP1)，此模型基本上將排課問題視為課程與時段的指派問題 (assignment problem) 而不考慮任何其他的排課限制。Dimopoulou 等學者 [11] 以及 Birbas[8] [9] [10] 對於希臘高中以及大學的排課問題提出一個數學規劃模型，此模型為一個較有效且緊緻的排課問題模型並且考量了許多重要的規則與限制。Ferland 和 Roy [13] 也將排課問題考慮成指派問題，從指派問題的觀點切入問題的領域，然後再將問題退化成 quadratic 指派問題來加以求解。Tripathy [25] 考慮成大型的 0-1 整數規劃問題並且使用 Lagrangean relaxation 來解決排課問題。

圖形著色問題 (Graph Coloring problem) 是古典圖形理論中 NP-Complete 的問題之一。Werra [26]，Selim [24] 提出利用圖形著色的方法來解決排課的問題，以有色的頂點代表課程，不能有時間衝突的兩課程間以邊連接，

如此只要確定有邊相連的兩頂點不會塗上相同顏色即可保證這兩門課不會衝堂。Werra [26] 利用圖形著色的理論求解簡單的高中班級-教師排課問題。

Tabu 搜尋是一種用來解決最佳化問題的演算法，它是一種區域搜尋 (local search) 的技術，該技術是一種植基於鄰域 (neighbor) 概念的區域搜尋法。搜尋開始於問題的初始解，初始解可以利用其他技術獲得或是隨機產生。然後，該演算法會進入搜尋空間的迴圈中，並且開始從一個問題解跳至解的鄰域範圍做更進一步的搜尋以至終止條件被滿足。Costa [6]，Hertz [15][16]，Schaerf [23] 應用 Tabu 搜尋於排課問題的研究上。Hertz 考量了課程群組的選項以及不同長度之課程的排定。

排課問題本身牽涉到教師，教室，課程以及時段等相當多的相關變數，因此問題之搜尋空間龐大且複雜，傳統的數學規劃方式有其應用上的限制，不足以幫助求得問題的滿意解答。近年來許多研究學者紛紛採用人工智慧的技術希望能夠對此問題提供一個解決的方法，而基因演算法也就自然而然地變成了首選的應用技術。

使用基因演算法於排課問題之研究首先由 Colorni 等學者[4]於 1990 年提出。Colorni 使用基因演算法嘗試建構義大利一所高中的課程表並且獲得不錯的結果。他們使用一個二維矩陣表示一個課程表。每一列代表一位教師，每一行代表一個時段，而矩陣中之元素值表示某課程。利用基因演算法，他們發現對於其所討論之排課問題中之硬性限制均可被滿足，但在演化過程中需要使用某些修補運算子來進行修補染色體的工作。Abramson 等學者 [1] 則考慮先將教師、教室、班級、課程這四項因素預先排好，然後再使用基因演算法以求得授課時段之排程結果；研究結果顯示當面對龐大的搜尋空間時，採用基因演算法的技術在搜尋問題之極小值的表現上相當良好。Wilke 等學者[27] 採用一種混合式的基因演算法求解德國某高中之排課問題；此混合式基因演算法會於演化過程中引進一種所謂的重構 (reconfiguration) 運算子。在一般情形下，該運算子不會被執行，只有當基因演算法在幾個世代後仍然無法產生高於目前最佳適應值之染色體時才會被啟動。研究數據結果顯示採用此方式可以得出比一般之基因演算法更優良之

排課結果。

2.2 派課問題

派課問題在本質上異於排課問題。本研究所探討之派課問題是指一個科系如何將下學期所欲開設之課程分派給系上之授課老師。在過去的文獻中，派課問題鮮少被獨立出來研究。Hultberg [17] 從古典的運輸問題 (transportation problem) 觀點來研究教師的派課問題，並且將問題表示成混合整數規劃問題 (mixed integer program)。對於派課問題，Hultberg 只考量了一個單純的條件：如何將指派給教師之不同科目的數量最小化。Hultberg 的研究將派課問題轉化成基數分割 (cardinality partitioning) 的極大化問題，並且藉此模式來證明派課問題是一個強 NP-困難 (NP-hard in the strong sense) 的問題。文中亦給出了一個使用分支限界 (branch-and-bound) 的演算法來求解其派課模型。

另一方面，有些學校的派課問題牽涉到教師與管理者，因此，從教師以及管理者兩個不同使用者的角度來考量，Ozdemir 等學者[19] 使用 0-1 非線性多目標規劃 (0-1 nonlinear multiobjective programming) 求解派課問題的最佳化。在派課問題中，Ozdemir 的作法是把每位教師的需求當成是一項需要滿足的目標，並以此作為問題的限制式；在問題上，不僅需要考量到各位教師們的需求，也必須考量到管理者的需求。Ozdemir 採用層級分析法 (Analytic Hierarchy Process, AHP) 來綜合簡化問題中之多目標函數成為單一函數，並且有效地求得其問題的最佳解。Parthiban 等學者[20] 也採用層級分析法並且提出一個以喜好為基礎之教師派課決策模型。

3. 問題描述

本研究所探討的派課問題是指一個科系如何將下學期所欲開設之課程分派給系上之授課老師。派課問題必須考慮到諸如教師喜好，教師專長，行政職務等因素，各因素有其輕重，因此，如同排課問題一般，我們將限制式區分為兩大類來考慮，分別為：硬性限制 (hard constraint) 以及軟性限制 (soft constraint)。所謂的硬性限制是指因為資源限制，學校規定或班級課程不能衝堂之性質，所必須要滿足的派課條件，是不可避免的條件。

而所謂的軟性限制則是問題中必須儘量去滿足的條件。大部份的軟性限制都是人為因素的考量。

3.1 軟性與硬性限制

本研究所規範之硬性限制與軟性限制列舉如下：

硬性限制 (Hard Constraints, HC)

- HC1. 同一門課程只能有一名教師被指派。
- HC2. 專任教師最少開授三門課，最多開授五門課。
- HC3. 兼任教師最多開授 3 門課。
- HC4. 兼負行政工作之教師最多開授兩門課。
- HC5. 所有班級的必修課程均有指派教師。

軟性限制 (Soft Constraints, SC):

- SC1. 教師的授課偏好應該儘量被滿足
- SC2. 被指派之課程的領域必須盡量與教師之專長領域相符合。
- SC1. 教師的授課時段偏好應該儘量被滿足。
- SC2. 同一教師被指派之相異課程數目越少越好 (此限制之目的在於期望相同課目最好由同一教師擔任)。
- SC3. 被指派之課程的領域必須盡量與教師之專長領域相符合

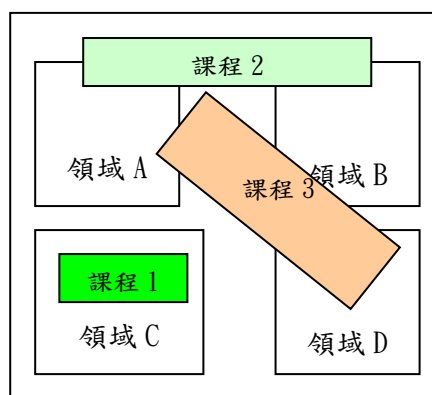
其中 SC2 軟性限制之設定主要是因為多數的教師不喜歡所被指派的課目太多，需要花費較多的時間去準備教材，此軟性限制可以降低教師的負擔，進而給學生較好的學習品質；而 SC3 軟性限制主要目的是希望課程的指派必須儘量與教師之專長相符合。

對於所探討之派課問題所牽涉之軟性限制，我們必須事先收集所需之資料以利將來演算法之運算。這些資料包含下列三項：「教師授課喜好表」、「課程領域分析表」、「教師專長分析表」，其中「教師授課喜好表」是用於 SC1 之計算；而「課程領域分析表」、「教師專長分析表」則是用於 SC2 以及 SC3 之計算。

「教師授課喜好表」：為一份包含該學期所有課程之課程表，對於一個學期所要開設之課程，每位教師對於課程提供其喜好度，喜好度等級區分為 10 等，其中 10 代表最滿意，1 代表最不滿意。每位教師需提供 10 個科目的喜好度於此表格上。

喜好 度 科目名稱	教師 一	教師 二	教師 三	...	...
課程一	5		10		
課程二		1			
課程三	2		6		

「課程領域分析表」：對於全系之課程結構做一綜合性分析並且建構課程之領域結構圖。一個課程可能只屬於某特定領域，但也有可能隸屬於不同的領域，下圖顯示課程 1 屬於領域 C，課程 2 屬於領域 A 與 B，課程 3 則屬於領域 A, B 以及 C。



上圖可以用表格之方式來表達如下：

領域/課程	課程 1	課程 2	課程 3
A		v	v
B		v	
C	v		
D			v

「教師專長分析表」：這是一份由教師所填寫之專長分析表，用以顯示其所屬之專長領域，其目的在於執行派課演算法時做為 SC2 計算之用。

	領域 A	領域 B	領域 C	領域 D
教師 1	v		v	
教師 1		v	v	v
教師 3	v		v	
教師 4		v		v

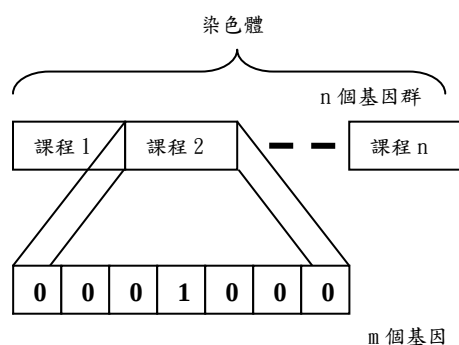
4. 設計方法

本研究採用基因演算法於派課問題的研究。因此，依據演算法之相關流程步驟做細部的說明，內容包括染色體的編碼方式、適應度函數的定義，以及選擇、交配、突變等相關之遺傳運算子的執行方式。

4.1 染色體編碼

根據基因演算法之執行步驟，首先定義本研究所要採用之染色體的編碼方式。一個染色體代表一個對於問題的解，因此，對於派課問題所定義出的染色體則代表一個派課的結果。染色體模式定義的好壞攸關接續之遺傳運算子執行的效能，從相關的排課問題研究中我們看到許多的演算法均需要有繁雜的基因修補過程，這大大降低了演算法的執行效率以及錯誤的可行解的產生。對於派課結果之染色體結構，我們定義如下：

令染色體 X 代表一派課結果， X 由 n 個基因群所組成，其中 n 為這學期所決定開設的課程數量，每個基因群包含 m 個基因，其中 m 為教師數量，所以每個基因群代表各個課程所排定的派課結果。染色體之圖示如下：



令 $C = \{C_1, C_2, \dots, C_n\}$ 表示所有課程所成之集合， $T = \{T_1, T_2, \dots, T_m\}$ 表示所有教師所成之集

合。定義決策變數 x_{ij} 如下：

$x_{ij}=1$ 假如教師 T_i 被指派到課程 C_j ；

$=0$ 如果教師 T_i 沒有被指派到課程 C_j ；

$1 \leq j \leq m$, $1 \leq i \leq n$ 。那麼對於派課之結果可以用一個矩陣 $X_{n \times m}$ 來表示。也就是說本研究以矩陣的形式來表示一個基因，這有別於傳統上只以數字或是字母為主的染色體編碼形式。另外，我們將前置作業所收集之「教師授課喜好表」轉換成喜好度矩陣 $F_{n \times m}$ 。喜好度矩陣中之每一列(row)代表所有教師對於該課程之喜好度，每一行(column)代表一名教師對於所有課程之喜好度。

4.2 適應度函數(fitness function)

適應度函數在基因演算法扮演相當重要的角色，因為適應度函數的責任在於決定一個染色體於演算過程中之生存機率。透過適應函數的計算，當染色體的適應值越佳，表示其所代表的解越能滿足問題的要求。一個染色體之適應值與其成本息息相關，對於成本較大的染色體，我們希望它的適應值越低。本研究之適應度函數由「派課總體喜好度(Total Average Preference, TAP)」以及「成本函數(cost)」兩大部分所組合而成，令 X 表示一個染色體，則其適應度定義如下：

$$fitness(X) = \frac{TAP(X)}{1 + cost(X)}$$

由上之公式可以看出，一個相當符合絕大多數教師喜好之派課結果所計算出之適應值會趨近於 1。這是因為對於該染色體 X ，我們會有 $TAP(X) \rightarrow 1$ 以及 $cost(X) \rightarrow 0$ 。下列小節分述各組成部分的定義。

4.2.1 派課總體喜好度(TAP)

「派課總體喜好度」代表全體教師對某一派課結果的平均喜好值。TAP 之計算由派課結果與「教師授課喜好表」矩陣 $F_{n \times m}$ 之轉置相乘而得。令 $P = \{P_1, \dots, P_n\}$ 代表喜好值所成的集合， n 代表課程總數，則 TAP 定義如下：

$$TAP(X) = \frac{\sum_{i \in I} \sum_{j \in J} x_{ij} p_{ji}}{\max(P) \cdot n}$$

$$= \frac{tr(X_{n \times m} F_{n \times m}^T)}{\max(P) \cdot n}$$

其中 $F_{n \times m}^T$ 表示喜好度矩陣 $F_{n \times m}$ 的轉置， $tr(A)$ 表示矩陣 A 的 *trace*， $\max(P)$ 代表集合 P 中之最大值。根據此定義，一個滿意的派課結果（指派給教師之課程皆滿足其最大喜好度）其派課整體喜好度之值將會趨近於 1。

4.2.2 成本函數(cost function)

成本函數是基因演算法中相當重要的一部份，成本函數的責任在於決定一個染色體的好與壞，讓演算法可以根據此資訊接續著相關的遺傳操作；另外，成本函數與下面要討論之適應值的計算也息息相關。對於一個染色體 X ，定義其 Cost 函數如下：

$$\text{Cost}(X) = d_1(X) + d_2(X)$$

$d_1(X)$ 代表違反 SC_2 之 cost，其計算方式定義如下：

$$d_1(X) = \sum_j \# \left(\sum_i G(x_{ij}) \right)$$

其中 $G(x_{ij})$ 代表課程 i 所在之課組的集合， $\#(S)$ 代表集合 S 中相異元素的個數。這裡要說明的是所謂的課組是指由課程名稱相同之課程所成的集合。例如：數學系負責全校微積分課程的開授，因為所需修課之人數為數眾多，因此，該系開列七門微積分課程以供全校學生選修。雖然是同一門課程，然而因為實質上有七個課程，為了區分其相異性，對於課程賦予不同之課程編號。因此，將這七門微積分課程歸類於同一課組，此一課組即為集合 $\{\text{Calculus_001}, \dots, \text{Calculus_007}\}$ 。 $G(x_{ij})$ 之計算方式以下表為例子做說明。

		教師一	教師二	
1	Calculus_001	v		
2	Calculus_002	v		
3	Calculus_003		v	
4	Calculus_004	v		
5	Calculus_005			
6	Calculus_006	v		
7	Calculus_007		v	

對於教師一，雖然說他/她被指派到四門微積分課程，但是實際上這四門課是同一課程。

此， $\sum_i G(x_{i1}) = \{\text{Calculus}\}$ ，所以

$$\# \left(\sum_i G(x_{i1}) \right) = 1。$$

對於其他的教師，依此類推計算並且加總起來作為 $d_1(X)$ 的值。利用此種計算方式，很顯然地可以看出 $d_1(X)$ 的值代表派課結果中教師被指派之相異課程數目，此計算值我們期望它越小越好，才能符合同一教師被指派之相異課程數目越少越好的目標。

$d_2(X)$ 代表違反 SC_3 之 cost。為了讓派課結果儘量符合適才適所的目的，達到最大的教學滿意度（也就是說不是因為選了就派，而是要考量到是否該教師有該領域專長），對於任意一個染色體 X ，我們定義教學滿意度函數 d_2 函數如下：

$$d_2(X) = \sum_{i \in I} \sum_{j \in J} x_{ij} (a_j - T_{ia_j})$$

其中 a_j 代表課程 j 所需要的領域數目， T_{ia_j} 代表教師 i 在領域 a_j 所擁有的專長數。 a_j 與 T_{ia_j} 這兩部分的資料來自於前置作業一以及前置作業二這個階段。 $d_2(X)$ 之設計考量主要是對於派課之結果計算出是否有將適任之教師指派到相關課程。如同 $d_1(X)$ 一樣，我們期望此數目越小越好。

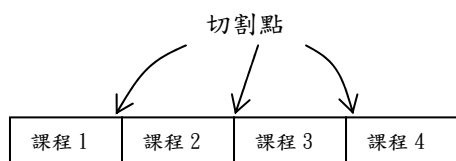
4.3 選擇

本研究之選擇運算子採用輪盤法則 (roulette wheel selection) 來決定哪些染色體將會存活至下一代；並且配合菁英保留法 (elitist selection) 的原則來確保一個世代中最好的一個染色體會存活至下一代。

4.4 交配

本研究採用單點交配法，但是作法上異於傳統基因演算法所採用的方式。傳統的單點交配法是於基因中隨機選取一個位置做為切割點。然而如果採用此法將需要額外的基因修復操作來校訂不合條件的基因，這將造成演算效率的降低。因此，對於切割點的選取，我們採

用隨機的方式來決定，而切割點的位置在課程與課程的交界處，如下圖所示：



下圖顯示交配前與交配後染色體的情況：

父代

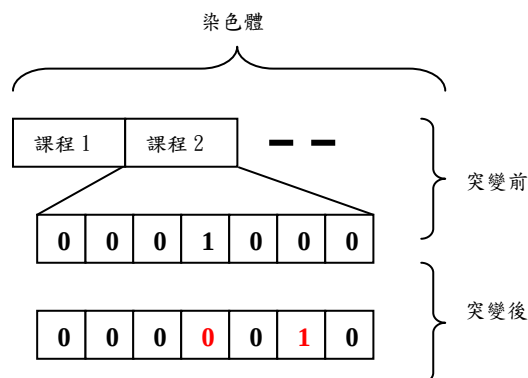


子代



4.5 突變

突變運算子的目的在於避免演算法太早收斂造成早熟的現象。這也是避免演算法落入局部最佳解所必須採行的步驟。本研究設定突變率為 0.01 並採用隨機方式選定於染色體中之任意基因位置做為突變的位置。對於選定之位置再隨機決定是否應該突變。如果是則將該基因位置之值做 0, 1 的變換，如下圖所示：



4.6 基因演算法

```
Produce initial population();
Find best chromosome in population();
while(not terminated())
{
    Generation number + 1 ;
    selection();
    crossover(){
        for all chromosomes in population{
            randomly select two different
            Chromosome();
            randomly select crossover
            position();
            exchange genes to produce
            offspring();
        }
    }
}
```

```
};
mutation(){
    x = randomly select a chromosome from
    population();
    t = randomly select a teacher from x();
    p = randomly select a gene from t();
    change the gene's value();
};
update the best fitness value(){
    For all Chromosome{
        F = Get Fitness value(){
            Compute TAP value;
            Compute cost value;
            Return fitness value;
        }
        If(F is better than current best
        fitness value){
            Update best fitness value;
        }
    }
};
}
```

5. 實驗結果

本節中，我們模擬一個六名教師、十五門課程的大學科系來做派課問題之研究，演化世代數訂為 3000 代，染色體 60 組，突變率為 0.01。表 1 及表 2 分別為教師們所填寫的對於十五門課程之教師授課喜好表及教師專長分析表，而表 3 則為課程領域分析表。

表 1 教師授課喜好表

教師 課目	教師 01	教師 02	教師 03	教師 04	教師 05	教師 06
課目 01	5			10		
課目 02	10	5		5		
課目 03				9		8
課目 04		8	7			
課目 05		6	9		10	
課目 06	9			7		
課目 07	6					10
課目 08			6			9
課目 09		10			5	
課目 10	8			8		5
課目 11		7				7
課目 12				6	8	
課目 13			10			6
課目 14	7	9				
課目 15			8		9	

表 2 教師專長分析表

領域 教師	領域 01	領域 02	領域 03	領域 04	領域 05	領域 06
教師 01	v	v			v	v
教師 02	v		v	v		
教師 03			v		v	v
教師 04	v	v				v
教師 05	v		v		v	
教師 06		v		v		

表 3 課程領域分析表

領域 課目	領域 01	領域 02	領域 03	領域 04	領域 05	領域 06
課目 01		v				
課目 02	v	v				
課目 03		v				
課目 04				v		
課目 05			v		v	
課目 06	v					
課目 07		v				
課目 08		v		v		
課目 09	v			v	v	
課目 10	v					
課目 11	v		v			
課目 12	v			v		
課目 13						v
課目 14	v				v	
課目 15				v	v	

經過基因演算法操作之後可得到以下結果(表 4~表 6)，深色部份為基因演算法所排定之派課結果，進行比較之後可觀察到，適應值越高之派課結果明顯較符合教師所期望之課程又可以平均分配課程，以維持公平性。

表 4 派課結果，適應值=0.128

教師 課目	教師 01	教師 02	教師 03	教師 04	教師 05	教師 06
課目 01	5			10		
課目 02	10	5		5		
課目 03				9		8
課目 04		8	7			
課目 05		6	9		10	
課目 06	9			7		
課目 07	6					10
課目 08			6			9
課目 09		10			5	
課目 10	8			8		5
課目 11		7				7
課目 12				6	8	
課目 13			10			6
課目 14	7	9				

課目 15			8		9	
-------	--	--	---	--	---	--

表 5 派課結果，適應值=0.108

教師 課目	教師 01	教師 02	教師 03	教師 04	教師 05	教師 06
課目 01	5			10		
課目 02	10	5		5		
課目 03				9		8
課目 04		8	7			
課目 05		6	9		10	
課目 06	9			7		
課目 07	6					10
課目 08			6			9
課目 09		10			5	
課目 10	8			8		5
課目 11		7				7
課目 12				6	8	
課目 13			10			6
課目 14	7	9				
課目 15			8		9	

表 6 派課結果，適應值=0.067

教師 課目	教師 01	教師 02	教師 03	教師 04	教師 05	教師 06
課目 01	5			10		
課目 02	10	5		5		
課目 03				9		8
課目 04		8	7			
課目 05		6	9		10	
課目 06	9			7		
課目 07	6					10
課目 08			6			9
課目 09		10			5	
課目 10	8			8		5
課目 11		7				7
課目 12				6	8	
課目 13			10			6
課目 14	7	9				
課目 15			8		9	

以下為使用本校資管系之派課問題所進行的實例驗證。全系共有 18 名教師、45 門課程，演化世代訂為 30000 代、族群數目訂為 60 組、突變率為 0.01。所使用之電腦配備:CPU 為 Intel P4 2.80G、RAM 1.5G。

根據上述設定，以本研究之模型執行 100 次運算，運算結果如下：

運算結果世代最大適應值為 0.057460
 運算結果世代平均適應值為 0.051488
 運算結果世代最小適應值為 0.045185
 平均每世代執行所需運算時間:8 分鐘

下圖為本研究模型執行100次運算的世代平均適應值之演化曲線，可由圖中觀察得知演化曲線約在5000代後開始收斂，於15000代後趨近穩定，證實此派課模型符合基因演算法之特性，並可使用於求解派課問題。

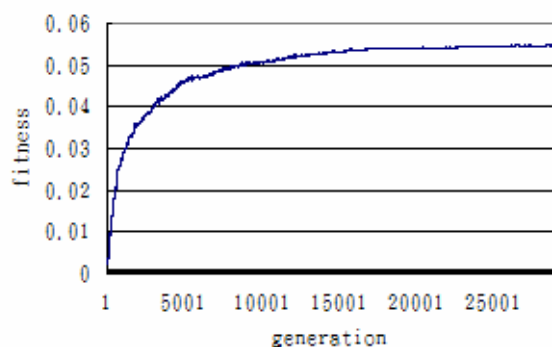


圖 1 平均適應值演化曲線

6. 結論

本研究初步的証實基因演算法對於派課問題有很大的幫助，並可以透過實驗結果的圖表證明此模型符合基因演算法的架構，適應值會隨著演化世代數不斷上升並趨於穩定。本研究最主要的貢獻在於不僅維持了派課的公平性，以及教師對於派課結果的滿意度，更能針對教師的專長作為派課上的一大要素，讓有長才的教師能被分配到適任的課目，而不是單純以教師授課喜好表作為派課的唯一依據，如此也能大幅提高學生的學習品質。

未來的研究方向可以朝向分析各變量對於結果的影響，亦可結合排課系統實作大專院校的派課與排課系統，以期讓更多的作業人員不再為每學期的課程問題勞心費神。

參考文獻

- [1] Abramson, D. and Abela, J., A parallel genetic algorithm for solving the school timetabling problem. *Technical report, Division of Information Technology, C.S.I.R.O.*, April, 1991.
- [2] Akkoyunlu, EA, A Linear Algorithm for Computing the Optimum University Timetable, *Computer Journal*, 16, pp. 347-350, 1973.
- [3] Bardadym, V. A., Computer-Aided School and University Timetabling: The New Wave, *Selected and Revised Papers of the 1st International Conference on Practice and Theory of Automated Timetabling*, (PATA 1995), Edinburgh, Springer LNCS 1153, pp. 22-45, 1996.
- [4] Coloni, A., Dorigo, M., and Maniezzo V., Genetic algorithms and highly constrained problems: The time-table case. In G. Goos and J. Hartmanis, editors, *Parallel Problem Solving from Nature*, pp. 55-59, Springer-Verlag, 1990.
- [5] Cooper, T. B., and Kingston, J. H., The Complexity of Timetable Construction Problem. *Proceeding of the First International Conference on the Practice and Theory of Automated Timetabling* (ICPTAT), 1995.
- [6] Costa, D., A Tabu Search Algorithm for Computing an Operational Time Table, *European Journal of Operational Research*, 76, pp. 99-110, 1994.
- [7] Csima, J. and Gotlieb, C., Tests on a Computer Method for Construction of School Timetables, *Comm. ACM* 7, pp. 160-163, 1961.
- [8] Daskalaki, S., Birbas, T., and Housos, E., Timetabling for Greek High Schools, *Journal of the Operational Research Society*, 48, pp. 1191-1200, 1997.
- [9] Daskalaki, S., Birbas, T., and Housos, E., An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research*, vol.153, pp. 117-135, 2004.
- [10] Daskalaki, S., Birbas, T., Efficient solutions for a university timetabling problem through integer programming, *European Journal of Operational Research*, vol.160, pp. 106-120, 2005.
- [11] Dimopoulou, M. and Miliotis, P., Implementation of a university course and examination timetabling system, *European Journal of Operational Research* 130, pp. 202-213, 2001.
- [12] Fang, Hsiao-Lan, Genetic Algorithms in Timetabling and Scheduling, *Ph.D. dissertation, Dept. Artificial Intelligence*, University of Edinburgh, 1994.
- [13] Ferland, J. A. and Roy, S., Timetabling problem for university as assignment of activity to resources, *Computers and Operational Research*, 12(2), pp. 207-218, 1985.
- [14] Gotlieb, C. C., The construction of class-teacher time-tables, *Information Processing, Proceedings of IFIP Congress* 62, pp. 73-77, 1963.
- [15] Hertz, A., Tabu Search for large scale timetabling problems, *European Journal of Operational Research*, 54, pp. 39-47, 1991.

- [16] Hertz, A., Finding a feasible course schedule using tabu search, ***Discrete Applied Mathematics***, 35(3), pp. 255-270, 1992.
- [17] Hultberg, T. and Cardoso, D., The teacher assignment problem: A special case of the fixed charge transportation problem, ***European journal of operational research***, vol.101, pp. 463-473, 1997.
- [18] Johnson, E. L., Nemhauser, G. L., Savelsbergh M.W.P., Progress in linear programming based branch-and-bound algorithms: An exposition, ***INFORMS Journal of Computing*** 12, 2000.
- [19] Ozdemir, M. S., and Gasimov, R. N., The analytic hierarchy process and multiobjective 0-1 faculty course assignment, ***European Journal of Operational Research***, 157, pp. 398-408, 2004.
- [20] Parthiban, P., Ganesh, K., Narayanan, S., Dhanalakshmi, R., Preferences Based Decision-making Model (PDM) for Faculty Course Assignment Problem, In ***Proceedings of Engineering Management Conference, IEEE International***, pp. 1338-1341, 2004.
- [21] Pearl, J., Heuristics: ***Intelligent search strategies for computer problem solving***, Addison-Wesley, Reading, MA, 1984.
- [22] Schaerf, A., A survey of automated timetabling. ***Technical Report CS-R9567, CWI***, Amsterdam, NL, 1995.
- [23] Schaerf, A., Tabu Search Techniques for Large High School timetabling Problems. ***Centrum voor Wiskunde en Informatica Report Rapport***, CS-R9611, 1996.
- [24] Selim, S. M., Split vertices in vertex colouring and their application in developing a solution to the faculty timetable problem, ***The Computer Journal***, 31(1), pp. 76-82, 1988.
- [25] Tripathy, A., School timetabling – A case in large binary integer linear programming, ***Management Science***, 30(12), pp. 1473-1489, 1984.
- [26] Werra, D. de, An introduction to timetabling, ***European Journal of Operational Research***, 19, pp. 151-162, 1985.
- [27] Wilke, P., Grobner, M., and Oster, N., A Hybrid Genetic Algorithm for School Timetabling, ***Advances in Artificial Intelligence: 15th Australian Joint Conference on Artificial Intelligence***, Canberra, Australia, LNCS 2557, 455-464, 2002.