**一個智慧型的排課系統**

|  |  |  |
| --- | --- | --- |
| 吳佳霖 | 劉師睿 | 吳孟修 |
| 國立宜蘭大學 | 國立宜蘭大學 | 國立宜蘭大學 |
| worry74147@gmail.com | knight31241@gmail.com | speater306@gmail.com |

|  |  |  |
| --- | --- | --- |
| 梁家菁 | 溫偉成 | 蔡崇煒 |
| 國立宜蘭大學 | 國立宜蘭大學 | 國立宜蘭大學 |
| qwe13527452@gmail.com | wilber830915@gmail.com | cwtsai0807@gmail.com |

# 摘要

現今國內外大專院校在每學期所開設的課程數量非常多，除了各學系的專業課程，亦包含通識課程。因此開課之前，系上需要排定下一學期預計開課的課程時段及教師，需同時考量多種需求及限制，因此這樣的問題需要花費相關人員許多時間進行此項工作。這之中包含：1.校內通識課程課表、2.系上各學程之課程、3.授課老師的可行的授課時間、4.可用專業教室之時段，及5.學生可行時間。多樣化的需求與條件，使排課工作成為一個真實排程 (scheduling) 問題。在本研究中，我們首先發展一個排課系統，在這系統之上加入智慧型的方法，找出最適合排課結果，盡可能符合上述需求，以提供一個有效的排課工具。研究中將分別使用爬山演算法 (hill climbing)、模擬退火演算法 (simulated annealing)、及遺傳基因演算法 (genetic algorithm)進行排課，並將排課結果數值化並進行比對，以找出較好的排課結果。本研究的實驗將以國立宜蘭大學105學年度課程資料，與資訊工程系之教師所開設課程為基礎，進行模擬與測試。實驗結果說明模擬退火法在此系統，將可以找出較佳的排課結果。

**關鍵詞：啟發式演算法、排程問題、排課系統。**

## 簡介

排程問題是一個傳統的最佳化問題，其應用可以回溯之早期的工廠機器排程及各式各樣的工程應用 [6]。一個好的排程方法，可以有效的節省工廠機器工作成本。近年由於雲端計算技術的成熟，已被許多研究領域及企業廣為使用，如何將雲端計算平台上的各項資源，有效的排程，亦是許多研究學者關心的重點議題 [7] 。本研究的重點主要著重在於排課系統中，各項課程的排程，以提供一個有效的工具，協助系所在進行此項繁複且冗雜的工作時，可以規劃出一個較符合多數人需求的課表。

排程問題 (scheduling problem)是一個NP-complete的問題 [3]，主要是因為這類問題的複雜程度，目前並沒有一個有效的方法可以在合理（polynomial）的時間找到最優解 (optimum solution)。一般系所的排課問題，可以被視為一種排程問題的特例。其主要工作是將多種工作 (課程) 及機器 (教師) 疊合在一起進行規劃，且不能違反規則。

排課問題依照不同區域有不同定義，例如在英國的部分教育單位，排課不只要排課程，同時也要排定考試的時間 [2]。另一方面，在台灣的大專院校大部分的考試時間則是排定在授課時段，交由授課老師自行安排，大部分的情況不需要特地安排考試的時間與教室。不論在哪個區域，大部分的情況排課問題的共通點如下: 1.就老師而言，同時間不可有兩門課，2.就班級而言，必修課不可以衝堂，3.就教室而言，同時間不可有兩門於課同教室。若違反以上安排，則無法上課，我們將他稱為硬性規則。相反的，即使排課方式可能違反一些需求，但還是可以上課，我們稱為軟性規則。例如：課程的時段是分散的，或老師連續上課的時段過長。若違反這些安排，雖然會造成老師或學生的困擾，但並不會造成課程無法進行。一個較人性化的排課策略，將是提升教學效率的一個有效答案。

為提供一個有效的排課工具輔助系上排課工作，本研究首先建置一個完整的排課系統，將所有的課程資訊、教師資訊、學生資訊、教室資訊輸入至本系統，以利後續排課作業。本研究並且在此排課系統之上，導入啟發式演算法 (metaheuristic algorithm) 對於真實排課問題進行排課，以找出更合適的排課策略，輔助系上排課工作。其主要貢獻如下:

1. 建置一個完整的排課系統，可以實際讓系所排課使用。
2. 透過 啟發式演算法的計算與搜尋，此系統可以提供一個比規則式方法 (rule-based method) 更合適的排課結果，經由進一步的微調排課結果之後，將可以提供更有效且快速的排課方式。

## 相關研究

* 1. **排課系統**

如下表1所示，目前國內已有許多大專院校釋出及開放其排課系統給大眾使用，其中包含：南華大學 [8]、中華大學 [9]、STC [10]、

**表1. 排課系統比較**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 排課系統 | 南華大學 | 中華大學 | STC | AutoClass | 欣河系統 | 悟空 | 高校智能 |
| 資料存檔功能 | ● | ● | ● | ● | ● | ● | ● |
| **教室性質資料** |  |  |  |  | ● |  | ● |
| 老師詳細資料 | ● |  |  |  | ● |  | ● |
| 課程詳細資料 | ● | ● | ● | ● | ● |  |  |
| **老師有空時段** |  | ● |  |  |  | ● |  |
| 圖表式課表 |  | ● | ● | ● |  | ● |  |
| **拖拉排課功能** |  |  | ● | ● |  |  |  |
| 輸出各種課表 | ● | ● | ● |  | ● |  |  |

AutoClass [11]、何欣系統 [12]、悟空 [13]、高校智能 [14] 等排課系統。在我們先前的研究中，我們依據各項排課系統所提供的功能進行整理，從這之中可以發現，有三項功能是目前國內大部分開放式排課系統比較少具備地的，包含:「教室性質資料」、「老師無課時段」以及「拖拉式排課介面」。雖然較少系統具備這些功能，但就一個完整的排課系統而言，這些功能是不可或缺的。因此這幾項功能將是本研究發展之排課系統目標。其中，老師無課時段將會是排課系統的其中一個主要考慮因素，在後續的章節有更詳細的介紹。

* 1. **啟發式演算法**

近年因許多資訊、工程、商業及科學等研究中發現，許多需要解決的複雜問題是NP-hard或NP-complete的最佳化問題 (optimization problem)，如使用傳統的窮舉法 (exhaustive search) 所需花費的時間，以目前的電腦系統，仍無法在有限的時間內找到最好的結果。而啟發式演算法 (metaheuristic algorithm) [1] 提供我們另一種可行的方案，其主要的概念是透過有策略性的猜測，往比較可能方向進行搜尋。透過反覆執行: 解的轉換 (transition)、解的評估 (evaluation)、及決定搜尋方向 (determination) 三個步驟 [7]，相較於窮舉法，啟發式演算法將能在有限時間內，找到一個或多個接近最佳解的近似解 (approximate solution)，以解決複雜或巨大的最佳化問題。

1. **模擬退火演算法**

這種方法最初的設計理念示模擬冶金學中「退火」的演算法。模擬退火演算法 (simulated annealing) [5] 與爬山演算法相同的是共有三個主要的運算過程 (operator)，改變解 (transition)、計算解迭代 (evaluation)、決定解 (determination)。在每一個迭代 (iteration)，藉由產生一個或多個候選解（數量依照參數設定），然後在決定解的部分 (determination)，有機率P（**參考公式1**）選擇該解成為目前的解。若新產生出來的解*v*的結果比較好，則一定選擇該解為目前解*s*到下一個迭代；若新產生的解*v*的結果較差，還是會有機率選擇成為目前解*s*，但是機率會根據溫度的降低而減少，此方法用以跳出區域解。

## *P* = (1)

相較於爬山演算法，由於模擬退火演算法使用公式 (1)來取代原有爬山演算法只能接受更好的解的搜尋方式，使得模擬退火法，有機會跳脫出區域最佳解，進一步找到更好的解。

1. **遺傳基因演算法**

遺傳基因演算法 (genetic algorithm) [4] 是以達爾文所提以出的「適者生存，不適者淘汰」進化論概念，所設計出的啟發式搜尋演算法。由於大自然中，每一個生物都要繁衍下一代，我們將上一代稱為父代，下一代稱為子代。父代在繁衍的時候，有可能會發生染色體交換，也有可能會發生突變，或者是兩者皆有，這些變化會經過自然規則的淘汰，漸漸的留下的都是適合環境。以這個概念模擬出自然環境，會存活下來的大多是比較適合環境的生物 (解)，用在最佳化問題之上，有利於找到較適合的解。如圖1所示，我們將基因遺傳演算法分為四個部分，1.計算適應度（fitness）、2.選擇（selection）、3.交配（crossover）、4.突變（mutation）。

*s* = Initialization

= Fitness (*s*)

Do

= Crossover(*s*)

= Mutation()

= Fitness ()

*s* = Selection (*s*, , )

While

Output *s*

**圖1. 遺傳基因演算法**

## 智慧型排課系統

本章節首先將介紹本研究所發展的智慧型排課系統設計，並詳細說明本系統之核心排課演算法的實作方式。

* 1. **系統整體架構**

為發展一個好用的排課系統，我們歸納幾個發展重點，其中包含: 1.考量老師需求，2.以智慧型演算法進行人性化排課，3.允許排課結果可以用拖拉式介面進行微調，4.提供教師一個簡單的介面，隨時可以查看最新排課結果，及5.所有資料將會儲存在資料庫以供後續使用。

排課程序分為三個作業。第一個作業是事前作業，事前作業包含搜集老師資訊（例如:優先度、可排課時段及其他相關資訊），課程資訊（例如:負責老師、開課堂數等相關資訊），及教室資訊。事前資料的蒐集將會透過API至相關系統蒐集，以及透過網頁方式蒐集相關資訊，上傳至排課系統及資料庫。第二個作業階段是將相關資訊進行整理，並以啟發式演算法進行排課。當啟發式演算法排課結束後，系統將利用網頁方式顯示初步排課結果，而排課人員可透過我們所發展的系統，進行調整。第三個作業階段是當老師及排課人員對於排課方式有其他看法，則排課人員可以透過相關的協調、在系統介面上進行手動調課，直到排課結果是所有人可接受方式，系統將輸出最終結果。

* 1. **啟發式排課演算法**

在介紹本研究的排課演算法前，我們先說明在本文後續會使用到的相關參數，其定義如下:

*k*：任意正整數。

*TRk*：編號是k的老師。

*COk*：編號是k的課程。

*Clsk*：編號是k的班級。

*Ocls*：班級的順序。

*OTRk*：編號k老師的時間順序。

*STRk*：編號k老師的得分。

*TTRk*：編號k老師的總課堂數。

*PTRk*：編號k老師的優先度。

*Pk*：正規劃後編號k老師的優先度。

*Pmax、Pmin*：最大、最小優先度。

*x*：一組解的型態。

1. **解的定義及衡量方式**

雖然排課問題類似排程問題，但因其考量的部分有所不同，因此我們需先定義計算解的目標值 (objective value) 方式。首先，我先假定有以下資料，如表2所示。三位老師編號分別為*TR1、TR2、TR3*，優先度分別為1、2、3，優先度是以排名計算，越低代表優先程度越高。

**表2. 老師資料**

|  |  |
| --- | --- |
| 老師編號 | 優先度 |
| *TR1* | 1 |
| *TR2* | 2 |
| *TR3* | 3 |

如表3所示，課程資料編號分別是*CO1、CO2、CO3，CO1、CO3*是必修課（同班級的必修課間不可以衝堂），*CO2*則否。開課堂數都是3堂課。負責的老師（老師不可同時間有兩門課）分別為*TR2、TR1、TR3*。開課班級（同班級的必修課間不可以衝堂）分別是*Cls1、Cls2、Cls3*。

**表3. 課程資料**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 編號 | 必修 | 堂數 | 老師 | 班級 |
| *CO1* | ■ | 3 | *TR2* | *Cls1* |
| *CO2* |  | 3 | *TR1* | *Cls2* |
| *CO3* | ■ | 3 | *TR3* | *Cls3* |

如表4所示，系統加入人性化條件的設定。在排課時，不一定會達成，但是會影響結果，結果會盡量趨向人性化條件。例如:課程不要連續，主要的目的是讓老師可以有休息時間。

**表4. 人性化條件**

|  |  |  |
| --- | --- | --- |
| 老師編號 | 課不要連續 | 課不要同天 |
| *TR1* | ■ |  |
| *TR2* |  | ■ |
| *TR3* |  |  |

在排課演算法中，一個排課的結果，我們稱為一組解 *x*，如公式(2) 及公式(3)。

x = (2)

x= (3)

其中*x*包含1組班級的排課順序，以及*n*組老師的時間順序，如表5所示。以下簡稱為老師時順，我們將時間分成14個部分，將1~14重新排列即是老師時順。

**表5. 老師時順（14部分）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 一 | 二 | 三 | 四 | 五 | 六 | 日 |
| 上午 | 1 | 3 | 5 | 7 | 9 | 11 | 13 |
| 下午 | 2 | 4 | 6 | 8 | 10 | 12 | 14 |

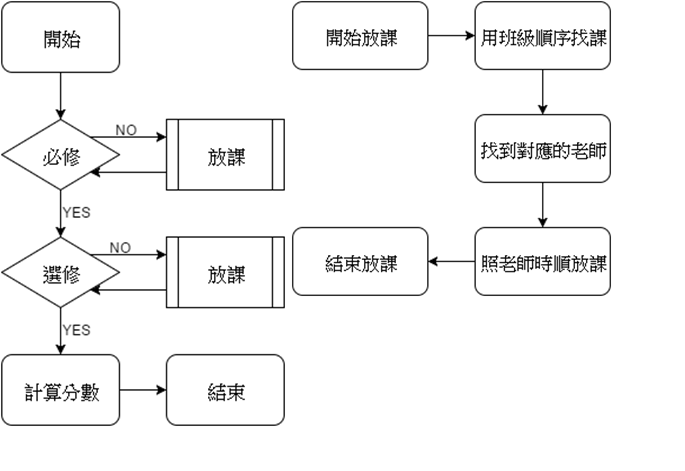
我們用*x*裡面的資料放入課程，若成功放入得STRk分，以公式(4)進行計算，否則不得分。*TTRk*就是編號*k*老師的總課堂數，也稱為該教師所有課的數量，*Pk*，以公式(5)進行計算。是將原本的優先度數值全部正規劃1至2之間。例如：*TR1*的總課堂數是3堂，因為*CO2*這門課有三堂；*TR1*的優先度是1，套入公式(5)可得到2。當*TR1*排入一堂課，得到6分。

*STRk = TTRk × Pk*.(4)

*Pk* = (5)

1. **排課系統初步放課的流程**

本研究所發展的排課系統，將用以下方式把演算法的一組解*x*排成課表，完成課表後才可計算總得分。如圖2所示，此系統將課放入課表的流程如下: 1.先放一輪必修課，2.再放一輪選修課，3.最後才計算分數。如圖2所示，放課的方式，是用班級順序找到該班級的必/選修課，再用課找到對應的老師，依照老師的時序，依序查看是否可以放課，可以放課則放該堂課進入開時段，若都不能放課進任何時段，在最後計算分數，即沒有該門課的分數。所以放的課越多，分數越高。



**圖2. 放課的方式**

以公式(3)作為範例，先從*Cls3*的必修中拿到*CO3*，連結到*TR3*的時序1，有空堂所以時序1放入CO­3的一節課，得到分數*STR3 =* 3 \* 1 = 3。再依序把*CO3*放入時序2.3，共得分9。再來要拿*Cls2*的必修，但是*Cls2*沒有必修，所以繼續拿*Cls1*的必修課*CO1*，連結到*TR2*的時序1，當要排入時順1、2、3時發現有*CO3*這門課，繼續尋找直到時序7，放入並得分STR2 = 3 \* 1.5 = 4。3堂課放完得12分，總得分共21分。最後我們拿到*Cls2*的選修*CO2*，他的老師是*TR1*，依照時序從4開始放課，得分是STR1 = 3 \* 2 = 6，該老師的總得分是18。如表6所示，放課的結果是全部接放入，最先放入*CO3*，*CO1*次之，最後則是*CO2*，整體的總得分是39。

**表6 .老師排課結果（14部分）**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 一 | 二 | 三 | 四 | 五 | 六 | 日 |
| 上午 | *CO3* | *CO3* | *CO2* | *CO1* | *CO1* |  |  |
| 下午 | *CO3* | *CO2* | *CO2* | *CO1* |  |  |  |

* 1. **智慧型排課演算法**

本研究將啟發式演算法套用並實作成排課系統，故選擇較經典且具不同特色的三種演算法。爬山演算法確認落入區域解的程度；利用模擬退火演算法以及基因遺傳演算法跳出區域解，且兩者分別屬於單解型 (single-solution) 與多解型 (multi-solution) 的啟發式演算法。

1. **爬山演算法應用**

爬山演算法一個迭代共有三個部分，改變解、計算解、決定解。計算解的做法即是使用公式(4)計算總分，決定解是將現在計算的總分跟當前最佳解比較擇優。在改變解的部分，我們採用的是隨機取兩個點互相交換。例如將公式(3)的x做改變，得到公式(6)。在班級順序部分，*Cls1*跟*Cls3*互換，在*TR1*的時順，1跟2互換位置；在*TR2*的時順，4跟12互換位置；在*TR3*的時順，3跟10互換位置。班級順序與每位老師時順都會隨機挑兩點互相交換，這是改變解的方式。

x= (6)

1. **模擬退火演算法應用**

模擬退火演算法比爬山演算法多三個參數，分別是初始溫度、最小溫度以及候選解的個數。做一個迭代也是三個部分，改變解、計算解、決定解，改變解與計算解都與爬山演算法相同。最大的差異在於決定解，此部分在章節2.2.A模退火演算法，已有詳細的說明。

1. **遺傳基因演算法應用**

基因遺傳演算法分為四個部分，1.計算適應度（fitness）、2.選擇（selection）、3.交配（crossover）、4.突變（mutation）。1.計算適應度的部分與前面兩個演算法的計算解相同，將每一條染色體套用公式(4)計算總分。2.選擇方法是以輪盤法，將每條染色體依照分數放在輪盤上，亂數選擇染色體，若分數越高則被選擇的機會越高。3.交配的部份我們是使用「遮罩」的方式交配。最後我們有一定機率會突變，4.突變的實作方式與前兩個演算法的改變解相同，隨機取兩點作交換。以上即基因遺傳演算法的一個世代。

## 系統實作與模擬

在本章節首先介紹系統環境，再介紹各項演算法的的實驗參數，並對實驗結果並對此加以分析，找出比較適合排課的演算法。最後以將介紹本研究所實作的排課系統。

* 1. **系統環境**

本研究所發展的排課系統，主要是以Java進行撰寫。實驗環境的電腦作業系統是Windows7，CPU是Intel(R) Xeon(R) CPU E3-1230 V2， RAM為8GB。實驗的使用範例資料是以國立宜蘭大學資工系105學年度與本科系相關之教師所開設課程為基礎，模擬設定各項排課時間之需求，共計11名教師，39堂資訊相關課程，11個含本系與非本系之班級，2項指定教室需求。

演算法的參數設定。在模擬的部分，每一個演算法都執行30次取平均值。且為了比較演算法的公平性，採用產生相同數量解的方式，讓每個演算法不會因為產生的解比較多，而有較好的結果。我們做兩個實驗，分別是產生32,000組與48,000組解。SA-1每代產生4組鄰近解，初始溫度是100，最小溫度是0.00001；SA-2每代產生16組鄰近解，其餘同上。GA-1每代有4條染色體，交配率是0.6及突變率是0.01， GA-2每代有16條染色體，其餘與GA-1相同。

* 1. **實驗結果**

實驗結果（如表7所示）最好的是SA-1，實驗結果是652.1636。實驗中的計分有三個權重相同的主要因素，1.不要同天開兩門課、2.不要連續開兩門課，以及3.課是否排進老師所希望之時段。若每項因素都能完成，就可以取得三個單位分數。主要差異在於第3項，課是否排進老師所希望之時段。在所有演算法中，結果最好的是SA-1，也就是使用模擬退火演算法，產生4個鄰近解的結果最好。在我們實作的排課系統中，模擬退火演算法的結果比爬山演算法、遺傳基因演算法好，因為模擬退火具有區域搜尋的機制，又有跳出區域解的退火機制，讓此方法得到最好的結果。 至於SA-2的結果並沒有比SA-1好是因為SA-2產生16組鄰近解，能執行的世代是SA-1的1/4倍，在世代少的情況下尚未完全收斂。SA-2尚未降至低溫，還會有較大的機率選擇較差的解，沒有對特定區域做更深度的搜尋，結果比SA-1差一些。

**表7. 實驗結果**

|  |  |  |
| --- | --- | --- |
|  | 數值(32000組) | 數值(48000組) |
| HC | 619.8273 | 619.9727 |
| SA-1 | 652.1636 | 659.9000 |
| SA-2 | 642.2091 | 648.4545 |
| GA-1 | 636.6727 | 643.2272 |
| GA-2 | 633.8090 | 635.2727 |

相較之下遺傳基因演算法的結果較差是因為交配（crossover）改變幅度過大，可能會大幅度的破壞結果，不僅沒辦法在特定的區域持續尋找，甚至造成結果破壞太大，要重新尋找一次結果。在實驗中的兩組結果GA-1的結果比GA-2好。雖然GA-2使用比較多條染色體，但是GA-2的結果尚未完全收斂，從實驗結果-32000（**如圖3**）可以看出，GA-2完全就在GA-1的下面，表示在產生相同數目解的情況下，染色體數量多並不會產生較好的結果。

為更加了解這些方法在排課系統的上特性，我們再做一次實驗，並把產生解的數目增加到48,000組解，用來觀察各演算法的適合度。從實驗結果（如表8所示）可以觀察出，HC已經沒有辦法再有大幅度變動，因為已經落入區域最佳解；其他演算法皆有再繼續爬升中，尤其以SA-1與SA-2最為明顯，且從實驗結果圖（**如圖3**）可以看出，SA-1的10000至20000迭代（Evaluation），相當於第2500代（Iteration）有迅速的爬升，那時候的溫度對於跳出區域解是很有效果的。在此排課系統中，模擬退火演算法是比較適合的，其中又以產生鄰近解數量為4的SA-1效果最好。

**圖3. 實驗結果-32000**

* 1. **系統成果**

在本研究中，所實作出排課系統共分為三個部分，1. 管理者介面、2. 使用者介面、3.排課介面。管理者介面（如圖4所示）是以Java為主撰寫，主要用途是讓管理者管理課程資料，左邊有八個按鈕，分別功能是新增/刪除老師、新增/刪除課程，上傳至資料庫、資料庫內容載入、班級管理、特殊教材管理，建立完成課程即可給予使用者填寫相關資訊。使用者介面（如圖5所示）是給予教授選取希望開課時段、其他人性化設定以及排課完成觀看課表。考慮到便捷性使用者介面是撰寫網頁版，讓使用者隨時隨地皆可以填寫，填寫完畢上傳至資料庫即可讓管理者使用。

除了上述的收集資料功能，排課系統（如圖6所示）運算核心也是使用Java撰寫。此系統可以選擇觀看老師課表、班級課表或是多重疊合觀看，若排課有衝突（同班級同時間有兩門以上必修課、同老師同時間有兩門以上的課），則以紅色標示，綠色代表安排合理，藍色代表那部分是老師有空閒的時間，透過這些標示可以讓使用者進行手動微調，排出理想的課程。



**圖4. 管理者介面**



**圖5. 使用者介面**



**圖6. 排課系統介面**

## 結論

本研究的主要目的是提供一個更好的方式解決排課問題。藉由實作一個排課系統，並採用啟發式演算法進行排課。實驗使用三種演算法，五種參數比較。根據實驗結果，我們觀察到模擬退火演算法，可以在較合理的時間內，得到最好的結果。遺傳基因演算法的結果則是次之，原因在於基因遺傳演算法需要收斂的時間比其他演算法都要長。如表8所示，染色體的相似度結果分析，GA-1有三成相似，但GA-2幾乎不相似，所以GA要收斂皆還需要更多的計算時間。我們還發現在這個問題之上，模擬退火鄰近解或跟遺傳基因演算法染色體4個的結果會比16個好，因為以測量解的次數為基準，一個世代產生的解比較少，可以執行比較多世代，所以只產生4個鄰近解或染色體的結果比較好。在未來的研究部分，我們將繼續嘗試設計更好的啟發式演算法，並將本系統套用至其他相關的資訊系統，以使資訊取得更加完善，更進一步減輕排課工作的負擔。

**表8. GA染色體相似度**

|  |  |  |
| --- | --- | --- |
|  | 32000組解 | 48000組解 |
| GA-1相似度 | 29.3% | 31.8% |
| GA-2相似度 | 0.3% | 0.1% |

**致謝**

本研究由科計部計畫MOST 105-2221-E-197-015補助支持，特此誌謝。

**參考文獻**

1. C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” ACM Computing Surveys, vol. 35, no. 3, pp. 268–308, 2003.
2. E. Burke, K. Jackson, J. H. Kingston and R. Weare, “Automated University Timetabling: The State of the Art,” The Computer Journal, vol. 40, no.9, pp. 565-571, 1997.
3. T.B. Cooper and J.H. Kingston, “The Complexity of Timetable Construction Problems,” in Proceeding of the Practice and Theory of Automated Timetabling, pp. 283-295, 1995.
4. J. H. Holland, Adaptation in Natural and Artificial Systems. MIT Press, Boston, MA, 1975
5. S. Kirkpatrick,, C.D. Gelatt,, and M.P. Vecchi,, “Optimization by simulated annealing,” Science, vol. 220, no.4598, pp.671–680, 1983
6. M. Pinedo. Scheduling: Theory, Algorithms and Systems. Prentice Hall, Inc., 1995
7. C.W. Tsai and J.J.P.C. Rodrigues, “Metaheuristic Scheduling for Cloud: A Survey,” IEEE Systems Journal, vol.8, no.1, pp.279-291, 2014.
8. 南華大學－線上排課系統, <http://csie2.nhu.edu.tw/files/archive/1343_39ba77b4.pdf> , 2016/08/20.
9. 中華大學－老師排課系統, <http://people.chu.edu.tw/~mioffice/chinese/ul02_project/project_94/21_text.doc> , 2016/08/20.
10. STC排課系統, <http://web.lcjh.ptc.edu.tw/front/bin/ptdetail.phtml?Part=arjuh016> , 2016/08/20.
11. AutoClass自動排課系統, <http://toget.pchome.com.tw/category/education/children/2967.html> , 2016/08/20.
12. 欣河資訊有限公司－電腦排課管理系統, <http://www.shin-her.com.tw/products/productslist.asp?ID=22> , 2016/08/20.
13. 悟空排課系統, <http://www.kajyh.com/pd.jsp?id=12&_pp=3_13> , 2016/08/20.
14. 高效智能排課系統, <http://2bysj.cn/Article/delphi/200905/1769.html> , 2016/08/20.