

# 選課分發系統之設計

鄭嘉齡 劉道光 曾黎明

國立中央大學電子計算機中心

中壢市五權里三鄰三十八號

TEL : (03)4227151 EXT. 7524

EMAIL : center8@cc.ncu.edu.tw

## 摘要

為解決排隊選熱門課的問題，選課分發系統因此產生。由於分發問題是一個很難的演算法，再加上各科條件日趨複雜，因此需審慎的加以規劃設計。本文的目的即在介紹本校選課分發系統的設計原則及演算法。

## 1. 前言

早期的選課系統為單純之登錄系統，通常是透過劃卡或終端機輸入，將各學生之選課記錄忠實的記錄下來。後因學生選課的自主性提高及課程的供需不平衡，產生排隊選熱門課的情形，為解決排隊所造成的問題，選課分發系統的需求，就因應而生。

就選課分發系統，本校經歷了體育分發系統，第一版選課分發系統，及現在的系統。此系統於八十五年由本校電算中心重新設計，經歷了一學年的使用與修正，目前已趨於穩定。本文的目的，即在介紹此系統的設計原則及分發演算法。

## 2. 系統架構

本校的選課系統可畫分為下列數個子系統：

課務管理子系統：主在維護課程、開課、教師等資料，及產生相關報表。

電子選課子系統：提供同學透過網路選課之功能，並提供選課各階段結果之公告，及問題反應之用。

分發及人工加退選子系統：主在對選課記錄進行分發。由電子選課子系統轉出同學所登錄之選課資料，進行分發，然後將結果轉給電子選課子系統進行公告。

目前分發及人工加退選子系統是以 dBASE III 為開發工具。因為課程可以單科，單群方式進行分發，因此分發處理可分散至多台 PC 執行，再將結果合併。

### 3. 分發處理之設計

#### 3.1 如何決定同優先權同學之分發順序

一門課有時會有多個分發的優先群組，如本班同學優先，本系同學次之，外系同學再次之。而對同一優先群組，又必須加以區別錄取之先後。區別的依據常見的有依資料登錄之先後，及抽籤法。依資料登錄順序之優點，在於規則明確，有鼓勵同學儘早選課的效果，但也造成需求壅塞，系統之公平性倍受同學矚目。例如網路上所有電腦所發出之服務請求是否具有相同之優勢。若否，則較優勢之區域勢必產生人潮，此時使用者進入此區域之公平性會受到考驗。若不幸系統毀損致使部分已選課同學必須重新來過，此時如何去平撫這些既得利益者，又是一大難題。

抽籤法之困難在於如何利用電腦達到類似抽籤的效果，而其優點，在於避免前述方法之缺點。在一個公平的抽籤系統下，可減少服務需求集中之情形，且在同學需重新登錄選課資料之最差情形下，所遭受之反彈也較小，因為籤尚未抽，沒有既得利益的問題。因此本校所採用的為抽籤法。

#### 3.2 抽籤法之注意事項

待選課登錄截止後，我們批次給每一筆選課記錄一個籤號，將籤號記錄下來，再進行課程分發作業。在設計整個抽籤法，有下列幾點需要考慮：

1. 可重覆驗證分發作業：在前一版的選課系統，相同的輸入會得到不同之輸出，使得除錯非常困難。其中一個原因，在於抽籤的機制。抽籤的設計在解決一分發優先群組中同學之錄取先後，而作法上可能為有需要時再抽籤決定，或不論有無需要全部抽籤，另一考慮因素為籤號是否儲存下來。為了達到可重覆驗證分發作業，使程式容易偵錯之目的，我們採取了批次全部抽籤之方式，且將籤號全部記錄下來。因此如果分發有誤，則不再進行抽籤，僅對有誤的地方進行重分發及核對之工作。
2. 唯一性：抽籤法的目的，在區別同一修課優先群組之錄取先後，因此不能再有無法區別先後之情形。也就是說，同一修課優先群組之選課記錄，彼此之籤號必需不同。
3. 公平性：透過抽籤法，我們希望不要對某些特定同學有偏好，例如先登錄之同學，或登錄順序為奇數號(或偶數號)之同學。

為了滿足前述的條件及簡單的目的，我們採用了批次及分科抽籤的方法。對單一的科目，假設登記此科之人數為  $n$ ，我們用一演算法將  $1, 2, \dots, n$  這  $n$  個數指定給這  $n$  個記錄。對一筆記錄，先取一界於  $1$  至  $n$  的亂數  $k$ ，若此數已被指定給此門課之其它記錄，則找出  $k+1$  至  $n$  中最接近  $k$  且未被指定之數值，若找不到，則從  $1$  至  $k-1$  找。

為了達到公平性，對一門課，一同學至多只能登錄一次。若否，則對一門課多登錄幾次就愈容易選上，不僅造成不公平，還造成後續處理上之困難。

### 3.3 其他原則

以往本校課程限制條件，由各系所提出，而系統維護者盡力為之的模式運作。由於沒有申請核定程序，因此條件趨於複雜，且許多條件有不明確，過於複雜，甚至矛盾的現象。為求系統的穩定性及可交性，我們認為分發處理應注意下列各點：

1. 課程條件的可程式化：此種設計，讓使用者自行設定條件，系統據以分發，而不是將條件寫死在程式中。在不超出系統之條件設計範疇，程式可不需作大幅修改而趨於穩定。
2. 可單科重分發：由於課程條件可能設定錯誤，因此必須能單科重跑，而單科重跑的先決條件是，任二科的分發彼此互為獨立，不因一科重跑而其他科必須重跑。舉例來說，衝堂自動退選會違反單科重跑原則，因為甲科重跑會產生新的衝堂情形或解除舊的衝堂情形，而造成其他科之結果必須作調整，如此循環影響下去。此時不但違反單科重分發原則，而且分發處理之收斂情形亦難以預期。另一會破壞單科重分發原則的為最高選課學分的檢核與自動退選。
3. 確定條件修改程序：如前所述，為確保系統之穩定性及維護性，此點是很重要的。

為了達到上述的目的，我們將現存的課程條件予以整理分類，找出具有彈性且系統又容易處理之條件表達方式，去除矛盾及目前無法處理之條件，然後設計演算法。

## 4. 課程條件表示法

依分發處理的方式來分，本校課程共分為一般課程及興趣選項課程二類。一般課程內的每一課程，可單獨分發。而興趣選項課程又分成數群，如體育，大一國文，大一英文，通識課程等。此類課程的主要特性，在於增加了志願的因素，分發時必須將課程條件與志願同時考慮。以下先就一般課程的條件表示法加以說明，再將興趣選項課程之條件設定，補充於後。

1	2	3	4	5	6	7	8
優先	$\geq$	人數	學院	系所	年級	班級	♀♂
0	=	0	*	*	4	*	*
1	$>=$	0	*	*	2	*	*

圖一：一體育課程之條件設定範例

### 4.1 一般課程

對於一門課之限制條件，我們是以圖一之型式表示，其意義條列如下：

1. 對一門課，可以設定多個優先群組，不同之優先群組以第一欄 [優先] 欄之值區別之。可設定每一優先群組之錄取人數上限於 [人數] 欄。
2. 對於一個優先群組，又可由一至多個限制條件所組成，一限制條件即為圖一中之一列，用以代表一學生群。一由多個限制條件所組成之優先群組，這些條件具有相同之優先值，代表所指定之多個學生群，對本門課具有相同之分發優先權。我們只能設定一優先群組之錄取人數上限，不能對各個學生群再加以人數限定。若需要如此，不應將其合併為一優先群。

3. 一限制條件即為圖一中之每一列，用第 2，4-8 欄來指定所代表之學生群。[ $\geq$ ] 欄有四種值可選，=， $<$ ， $\geq$  及  $\leq$ 。= 代表具有欄 4-8 所示性質之學生群，而 $<$ 則反之。 $\geq$ 僅對年級有效，代表一學生群具有欄 4，5，7，8 之性質，而其年級 $\geq$  [年級] 欄之值。 $\leq$ 亦同。舉一個實例來看，如圖二。代表著英文系(代號為 102)四年級優先，英文系二，三年級次之，而外系又次之，且外系至多五名。([人數]欄為 0 代表此科不限人數，\* 代表對該欄不加以限制)

1	2	3	4	5	6	7	8
優先	$\geq$	人數	學院	系所	年級	班級	♀ ♂
0	=	0	*	102	4	*	*
1	=	0	*	102	2	*	*
1	=	0	*	102	3	*	*
2	$<$	5	*	102	*	*	*

圖二：一英文系課程之條件設定範例

## 4.2 興趣選項課程

在電子選課子系統，除了讓同學登錄資料外，還有資料檢核，志願給定，批次抽籤等功能。所謂志願給定，就是當同學送出選課資料時，針對一興趣選項課群(如體育課)之所有選課資料，系統會根據其排列之先後，給予志願編號，代表此生修習這些課之意願高低。在興趣選項課程之分發處理，有下列幾點需要注意：

1. 興趣選項課程可分為多選一及多選多二類。體育課即為多選一之課群，代表一學生至多會選上一門體育課。而通識課程為多選多之課群，代表一學生可選上多門通識課。
2. 志願及分發優先群組孰先孰後。以體育課為例，規定大四優先，大二大三次之，則在一門課中，一大四同學以第二志願登錄，是否優於一大二同學以第一志願登錄?我們目前一律採用志願優先於分發優先群組之方式，所以前例中，大二同學是優於大四同學。使用此種方式的一主要原因，是對課群中的每一門課可設定不同之條件，在特定之條件組合下，若採用優先群組優先於志願之方式，會有矛盾產生，因此不提供優先群組優先於志願之方式。

因為興趣選項課程之分發較為複雜，因此興趣選項課程在人數限定上，我們僅保證單科總人數上限，至於優先群組之人數限定，則不建議使用。

## 5 分發處理之演算法

在利用一限制條件篩選學生時，需先將其轉換為所使用工具之語法。假設使用關連式資料庫，則圖一中之第一條件將被轉為 where grad=4。以下將一般課程及興趣選項課程之演算法，說明如下。

## 5.1 一般課程

相關檔案說明：

condition：課程條件檔

std\_choice：學生選課檔

程式變數說明：

course\_priority：目前分發優先群組之優先值

available：本科目前可再收之學生數

group\_upperBound：本分發群組之錄取人數上限

max\_slot：本分發群組實際之錄取人數

stop\_flag：本科是否處理完畢

符號說明：

CR(C)：資料集 C 中之目前記錄

CR(C).fieldName：資料集 C 中目前記錄之某欄位值

C.EOF：函數，傳回資料集 C 是否已至最後一筆

C.RecordCount：函數，傳回資料集 C 之資料筆數

C.Update：函數，將資料集 C 所作之更動寫回檔案

C.FetchNextRec：函數，擷取資料集 C 之下一筆記錄

假設目前所處理之課號為 X

處理流程：

```
1. update std_choice set status='未處理' where course_id='X'
2. C = select * from condition where course_id='X' order by priority
3. #course initialization
   set course_priority=0
   set available=X 課程之總人數上限
   set stop_flag=FALSE
4. #check if condition set is empty
   if C.EOF then

       update std_choice set status='候選' where course_id='X'
       set stop_flag = TRUE
       set group_upperBound = 1000
       goto step 10
5. # next priority group
   if CR(C).priority > course_priority then
       course_priority++
       goto step 10
```

```

6. set group_upperBound = CR(C).upperBound
translate CR(C) to string c'
7. update std_choice set status='候選' where course_id='X' and c'
8. C.FetchNextRec
9. if C.EOF
    set stop_flag=TRUE
    goto step 10
else
    goto step 5
10. #得到一完整之分發優先群組
S=select * from std_choice where status='候選' and course_id='X' order by random_no
11. set max_slot = Min( available , S.recordCount)
12. set max_slot = Min( max_slot , group_upperBound)
13. set available = available - max_slot
14. for j=1 to max_slot
    set CR(S).status='中選'
    S.Update
    S.FetchNextRec
next j
15. while not S.EOF
    set CR(S).status='額滿'
    S.Update
    S.fetchNextRec
16. if NOT stop_flag , goto step 6
17. update std_choice set status='資格不符' where course_id='X' and status='未處理'

```

## 5.2 興趣選項課程

興趣選項課程之處理，系統是採一群群分發，處理時需注意下列幾點：

1. 先處理此群中每一門課第一志願的選課記錄，將**中選同學**此群中之未處理選課記錄註記為'已上另一門'，然後處理每一門課之第二志願，第三志願，直到最後一志願。
2. 對於多選多之興趣選項課程，可同上述方法處理，唯省去'已上另一門'之註記動作。
3. 對於多選一之興趣選項課程，無法針對其中一科單獨作分發，而多選多之興趣選項課程理論上可作單科重分發。

## 5.3 第二次分發之前置處理

本校選課共分為二次的電子加退選及一次的人工加退選，前二次的電子加退選皆是由電腦進行分發。第二次分發處理在進行前，需經過前置處理。此前置處理需將第一次結果之中選名單，剔除第二次選課記錄中退選之部分，並據此中選名單計算各科目目前之錄取狀況。如某一科目目前已錄取多少人，及此

科各個優先群組目前已錄取多少人，以便計算各科、各優先群組可再錄取多少人。此處容易造成困擾的是，對於一門課，第一次分發及第二次分發之條件可能不同，條件可能放鬆，亦可能變緊，甚至截然不同。因行政人員疏失而強制第一次已中選之同學退選會造成極大之反彈，所以第二次所設定之條件僅會影響第二次選課記錄之分發。

## 6 分發處理所需之時間

在分發作業上，我們將課程劃分為一般課程以及興趣選項。在一般課程的分發，我們可以單科分散至一機器分發，而實際上我們是以系所為單位，一系之課程分散至一機器執行分發程式。但是興趣選項部分，我們只能以群作區隔，依不同群作分發，因此群的分發成為整個分發處理的瓶頸。每群所需之分發時間，依該群的課數多寡，各個課程的條件數，以及學生的選課人數而定。根據跑了兩個學期的經驗，興趣選項課群中以體育課需時最久，大概需二~三個小時(作業人員的 PC 為 pentium 133 or pentium 166, 一百門體育課，九千筆體育選課記錄)。因此在最佳狀態下，整個分發過程可在一個工作天甚至於半天內完成。但是作業人員仍必須作 double check，檢查結果是否正確，是否有條件設定錯誤，導致結果有誤差的情形發生。

## 7 其它

選課系統必須在相當短的時間內，通常是 7 至 10 天，登錄並處理全校數千同學之選課資料。另一方面，上課人數限制、某群組優先、依志願分發等需求，將選課系統由傳統的登錄資料系統導向複雜的分發系統。在短短數天內要完成資料登錄、前置處理(指資料登錄時所做之處理)、後置處理及資料輸出，因此各階段必須以有效的方式進行。將各階段所考慮之重點，分述如下：

1. 在資料登錄上，為求快速及資料之正確性，採使用者線上登錄之方式，請其自行檢核。此法有身分確認之問題，我們以可信賴主機上帳號為依據，作為將來責任歸屬之用。
2. 在前置處理上，為減輕系統負荷，縮短回應時間，我們僅作必要之前置處理，如使用者通行密碼檢核、加退課程動作是否合理等，以保證資料的有效性。檢核出不正確之情形，系統會將整個登錄需求退回。
3. 在後置處理上，因為選課分發的演算法十分複雜，為求系統之正確性，我們公佈處理原則，並將籤號、各科分發結果予以公佈，讓全校一起檢核，而非僅是公佈個人之選課結果。再加上可單科單群重分發之特性，使得可以在最短的時間內修正錯誤。
4. 在資料輸出上，採用電子佈告欄系統公告方式，讓使用者自行上線查詢。

本系統非即時線上選課系統，不採取即時線上選課的原因如下：

1. 對於有設優先群組之課程，以及可填志願之課程，必須待所有選課資料登錄完畢後，才能處理。因此只有不設限或僅有對象人數限制之課程，始可即時線上處理。而所有課程採同一處理模式有助於求系統之簡化。
2. 即時線上選課伴隨著先選先上之策略，而此策略不僅使壅塞的使用現象更趨惡化，並造成各科選修狀況變化極大。一方面同學們寧可多選些課，再慢慢退，而另一方面，今日額滿的課明日

不見得沒有機會。所以同學們會抱著一線希望，一試再試，造成系統持續性之過度負荷。萬一投諸了人力時間還選不中，此時同學們心中之不平是可想而知的。即時線上選課之原意，在於提升服務品質及使用滿意程度，可惜事與願違，目前多數的線上選課系統，反而遭致同學之質疑。是否能藉由訂定選課數之上限及上線登錄次數之上限來解決上述問題，則需要進一步的考慮。

3. 即時線上選課和銀行線上存款是相同的，只不過一個戶頭在同一時間有多筆交易去更改其金額的現象，遠小於一個熱門課在同一時間有多筆交易去更改其修課人數。也就是說，選課系統必須提供和銀行系統一樣精確的資料鎖定 (locking) 程序，而鎖定所造成的排除其它交易情形比銀行系統還嚴重，政策導致同學們持續送出交易資料更是雪上加霜。而此可見，即使提供和銀行業界相當之軟硬體，使用者等待的時間仍會長出許多。另一方面，購置近千萬之軟硬體以因應短時間的選課處理，其恰當性是很受質疑的，除非能找出在非選課期間的有效運用方式。

本校電算中心重新開發此系統所秉持的一個理念是，系統必須穩定、可用，以解決大多數之選課需求。對於其它目前無法納入之需求，則以半自動或人工之方式處理之。與其它學校之系統相較，本系統提供之功能或許稍有不足，例如非線上即時處理、學生登錄資料之次數受到限制等，但以系統穩定及可用性來看，本校學生不需花太多時間在登錄選課資料上。

## 8 結論

在整個設計過程中，主要的幾個關鍵有：

### 1. 學生選課資料的正確性

避免造成 garbage in garbage out 的狀況，因此我們嘗試用 BBS 的公告系統，作為學生即時選課的工具。

### 2. 選課流程的確認

經過幾度商討會議之後，請相關單位訂定選課流程，並確認之。

### 3. 課程條件表示法的包含度

綜合整理過去幾學期的選課條件，歸納出 20~30 種選課條件類型，依此設計出有彈性且系統可以處理的條件表達方式，並去除矛盾與無法表達的條件。

### 4. 分發處理的演算法

雖然我們已簡化許多條件，分發處理之演算法仍非常複雜。在可重新驗證結果之設計理念及制定課程條件修改程序下，系統已進入穩定狀態。另外值得一提的是，在興趣選項課程中條件優先亦或選課志願優先曾經造成困擾。在程式比較缺乏彈性的一次處理狀況下，條件優先造成了某學生以第  $n$  志願選上了一興趣選項課程，而其他同學的前幾志願卻未能選上該課的不合理情形。

選課分發是各個大專院校每學期都必須執行的業務，以上敘述敝校開發選課分發系統之經驗，希望能提供參考。