

使用基因演算法分配考試座位

指導教授:陳同孝

學生:黃柏鈞

摘要

如何防範作弊一直是困擾教師許久的問題，從分配座位著手是一種常見的方式，但是在分配座位時多半是依照老師喜好或是採用隨機方式，前者可能有考量不全面且每次考試需要費時重新分配一次座位，導致效率低下之問題，而後者難以保證座位分配結果之品質，無法有效避免作弊行為的發生。研究期望通過實際撰寫程式來分析與測試結果，使演算法能融入、應用予現實問題之中，達到學以致用之目的。

本研究使用基因演算法並在分配考試座位時考量學生關係、性別、平時成績、操性等因素，期望改善或是避免作弊行為的發生。以關係來說，考生與周圍同學之關係應該要越差越好；以性別來說，考生與周圍同學應不同性別較佳(梅花座)；以平時成績來說，考生與周為學生成績應該越接近越好(學生程度近似)；以操性來說，考生操性分數較高則假設不易發生作弊行為，不太需要監視老師放過多的注意力在該名考生上，操行成績較低者則反之。

本研究在驗證時使用一組已知最佳解之資料進行多次測試，而研究結果顯示本研究使用方法多次到達最佳解，證明本研究方法能夠依據問題考量之因素進行座位分配之最佳化；能夠省去老師自行分配座位耗費的精力，座位分配考量的全面性、座位分配品質等問題也獲得保障，更進一步避免了作弊行為發生。

作品連結:<https://github.com/p208p2002/use-ga-assign-examination-seat>

關鍵字:基因演算法、座位分配、防範作弊

目錄

一、前言	3
二、基因演算法介紹	4
2.1 基本名詞解釋	
2.2 基因演算法流程說明	
2.3 研究使用方法	
三、研究假設與適應函數	10
3.1 研究假設	
3.2 適應函數	
四、最佳化驗證	144
4.1 驗證使用資料	
4.2 驗證結果	
五、座位分配結果與分析	177
5.1 適應函數變化值紀錄	
5.2 座位分配結果	
六、結論	199
七、附件	199
八、參考文獻	20

一、前言

世上最早的考試可追溯到夏商周時期，而發揚考試制度則是從隋朝的科舉制度開始，伴隨著考試，作弊行為也隨之出現，可以說是只要有考試的地方就會免不了作弊的發生。作弊不但傷害他人的努力與付出，也會使選拔單位無法找到合適之人才，不僅如此，與作弊有關的創作題材也不時出現在我們的生活之中(如：電影，模範生，2017)，時時刻刻提醒我們不要作弊與可能面臨的後果，大眾對於作弊行為的重視可見一斑，因此如何防範作弊行為已經成為各考試主辦單位討論之重點。

有鑑於大型考場或是多考場環境與單一小型考場環境差異甚大，不適合放在一起討論，所以本研究選擇專注在單一小型考場如何藉由有效的座位分配方法防範作弊行為的發生，而排除了大型考場與多考場環境，因此本研究下述之考場、考場座位，均指單一小型考試場地。

單一小型考試場地常見於班級內部考試，而改變考試座位為常見避免作弊發生的方法之一，然而多數在實行改變座位方法時缺乏全面性的考量(可能只考慮到性別或是僅依照老師喜好)與效率(使用人力在每次考試前重新進行分配)，為此本研究將專注在如何分配座位使得作弊行為不易成功或是實現，並盡可能在作弊發生時將傷害降至最低。

研究將藉由多項假設、資料參數演算法與基因演算法提供一套有效的考試座位分配解決方案。

綜合上述，本研究目的如下：

1. 以程式協助考試座位分配，找出考試座位分配最佳化之結果
2. 改善以往以用人力分配座位欠缺全面性考量與效率不足

二、基因演算法介紹

基因演算法最早由國外學者 [1](John Henry Holland, 1975)提出，其基礎理論源自達爾文的進化論，以物競天擇與適者生存之特性，篩選較好的母代進行交配來繁衍子代，並且重複步驟來達到期望的結果。基因演算法適合解決趨勢預測、分配、組合最佳化之問題，其最大特點在於能跳脫局部解，並朝向最佳化前進。

基因演算法一開始的時候會產生初始世代，接者進行交配產生準下世代，然後經由適應值函數計算適應函數值並且進行篩選，最後再經由突變、交配、菁英政策等方式產生下一世代，這個過程會持續到滿足終止條件為止。依據終止條件的設定，我們將可能得到一個期望的結果。

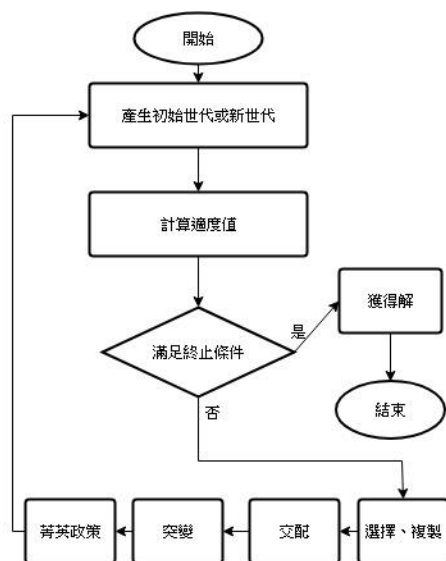


圖 2-1 基因演算法流程圖



圖 2-2 基因演算法染色體

2.1 基本名詞解釋

基因：染色體中每一個個別的變數 (圖 2-2)

染色體：由多個基因組成的群體

族群：由許多個染色體組成稱族群

2.2 基因演算法流程說明

基因演算法流程圖如圖 2-1 所示，以下將以文字詳細說明基因演算法流程。

1. 將問題編碼

將問題特徵進行編碼以利使用基因演算法進行運算，常見的編碼方式有二元(由 0 與 1 組成)編碼或是實數編碼(由 1,2,3...組成)。

2. 設計適應函數或計算適應函數值

適應函數是為了評估染色體對於問題的合適度。適應函數是作為保留染色體、達到終止條件之依據，所以應針對不同的問題設計不同的適應函數，並且所獲得的適應函數值必須能反映出染色體的好壞。

3. 進行選擇與複製

經由計算染色體的適應函數值，我們會優先選擇並複製適應值較高的染色體來進行交配，確保下一代的品質，而常見的選擇方式為俄羅斯輪盤法(俄羅斯輪盤法為一個隨機選擇的方法，但是會給予適應函數值較高的染色體較高的被選取機率，藉此增加保留優秀基因的機率)。

4. 交配產生下一代

在複製與交配完成後，被複製的染色體便在交配池裡進行交配的程序，以隨機方式選擇兩條不同的染色體進行配對，藉著交換彼此的基因位

元，產生量兩條新的染色體[5](邱雅鈴，2004)

5. 使用菁英政策

在產生下一子代的時候優先保留目前適度值最高之染色體，能夠使結果較快速的收斂也能夠將較優秀的染色體保留下來。

6. 染色體突變

突變指的是單一染色體發生變化，通常是染色體內基因互換，在一般基因演算法內突變機率並不會設定的太高以保持染色體的穩定性。突變是突破區域最佳解的重要方法之一。

7. 判斷終止條件

終止條件通常以適應函數值或是演化世代作為終止條件。而未達到終止條件則繼續從本節第 2 項輪迴，若達到終止條件則輸出結果。

8. 染色體解碼

在終止條件後我們會獲得一組染色體，但是染色體乃透過本節第 1 項將問題編碼而來，若要獲得答案則須將染色體進行解碼。一般解碼方式為編碼的反向操作。

2.3 研究使用方法

基因演算法依據問題與特性不同，在實作所使用之方法會有所不同，本節將針對本研究使用之方法進行舉例或說明。

2.3.1 編碼方法與舉例

本研究採用實數編碼方式，並且將考場教室視為一個二維陣列。以下

將圖 2-3 進行舉例，其中圖 2-3 為一隨機產生之考場座位分配，圖中編號表示學生座號。

2	7	1
4	5	6
8	3	9

圖 2-3 3x3 考場座位

將圖 2-3 考場教室座位視為一個二維陣列後，將二維陣列以列的方式橫向展開，如圖 2-4 所示，即完成編碼。

2	7	1	4	5	6	8	3	9
---	---	---	---	---	---	---	---	---

圖 2-4 3x3 考場座位編碼結果

2.3.2 交配方法與舉例

本研究交配方法使用 Mitsuo Gen 與 Runwei Cheng 在 1997 年提出的 PMX(Partially Matched Crossover)方法[2]進行交配來產生下一世代染色體，以下將示範如何使用 PMX 方法進行交配。

1. 假設染色體

先假設已經擁有下列兩組染色體，染色體 A 與染色體 B(圖 2-5)

染色體 A

2	7	1	4	5	6	8	3	9
---	---	---	---	---	---	---	---	---

染色體 B

4	6	1	2	3	9	7	5	8
---	---	---	---	---	---	---	---	---

圖 2-5 染色體 A 與染色體 B

2. 隨機產生兩切段

在染色體 A 與 B 上產生如圖 2-6 所示之灰色切段，其切段長度至少為 1 至大不超過染色體基因數量。一般切段長度不超過基因數量之 $1/3$ 。

染色體 A

2	7	1	4	5	6	8	3	9
---	---	---	---	---	---	---	---	---

染色體 B

4	6	1	2	3	9	7	5	8
---	---	---	---	---	---	---	---	---

圖 2-6 染色體 A 與染色體 B 基因切段

3. 交換切段

交換染色體 A 與 B 之切段(圖 2-7)，達成交配目的。

染色體 A

2	7	1	2	3	6	8	3	9
---	---	---	---	---	---	---	---	---



染色體 B

4	6	1	4	5	9	7	5	8
---	---	---	---	---	---	---	---	---

圖 2-7 染色體 A 與染色體 B 交換基因切段

4. 完成交配

交換後會在染色體內產生重基因，需要找到基因(圖 2-8)後進行復位，即完成交配(圖 2-9)。

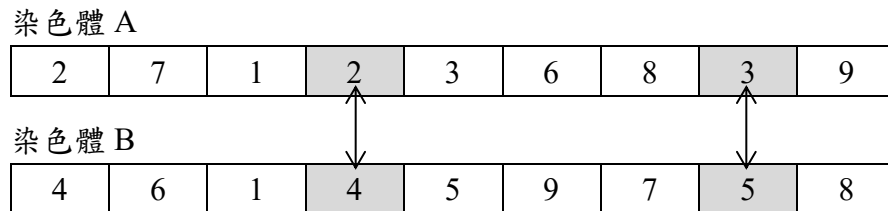


圖 2-8 染色體 A 與染色體 B 重複基因復位

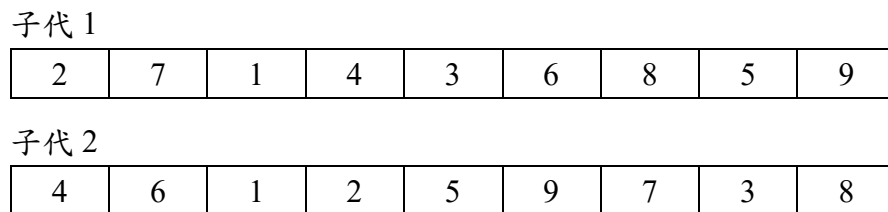


圖 2-9 染色體 A 與染色體 B 完成交配

2.3.3 突變方法

本研究在進行突變時會對染色體進行機率抽籤，如果抽中需要突變則隨機對調該染色體任兩個基因。

2.3.4 菁英政策方法

本研究在使用菁英政策時會先檢查目前交配池(完成交配與突變後所產生的族群)是否已存在目前記錄之最高適應值或高於紀錄之染色體，若有則不進行動作，否則將搜尋目前交配池中適應值最低之染色體，並將之取代。

三、研究假設與適應函數

本研究針對題目進行一些假設，並依此假設設計適應函數。

3.1 研究假設

研究假設為題目所假設可能影響座位分配結果之變數與環境因素。

3.1.1 考場假設

研究假設考場為一 $m \times n$ 之矩陣，並且前方有監試老師。如圖 3-1 所示，監試老師在前方進行考場監試，圖 3-1 中每一空格則為考生座位。

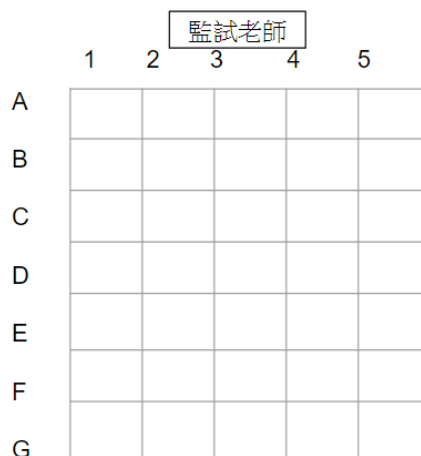


圖 3-1 教室格局與座位

3.1.2 變數假設

本研究假設下列幾項變數將影響座位分佈結果好壞：

1. 性別

研究假設如果座位分佈呈現男女交互可減少作弊發生機率。

2. 平時成績

研究假設如果周圍同學平時成績相近，可減少作弊發生時作弊

行為者獲得的不當利益。

3. 操行成績

研究假設監試老師 較易察覺前排學生之不當行為，且操行成績較低之考生較可能發生作弊行為。

4. 與其他同學之關係

研究假設如果與周圍同學關係較差可減少串通作弊行為發生。

3.2 適應函數

考量 3.1 節之假設，依提出的 4 個問題進行適應函數設計，並且加總 4 個問題所得之分數即為本研究所求之適應函數值。

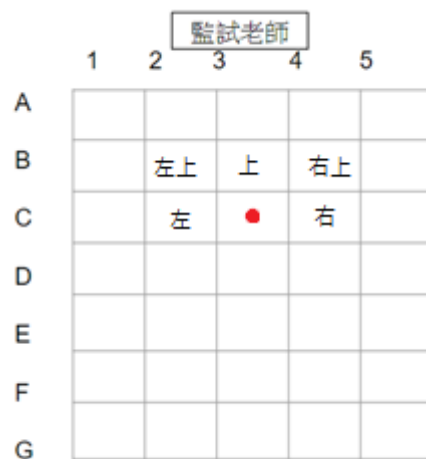


圖 3-2 座位周遭方位參考圖

圖 3-2 為下述問題座位周遭方位參考圖，圖 3-2 中之紅點(C3)表示目前參考之考生，但仍需注意根據設計若將紅點移動至邊界或是角落將有部分方位不存在。

1. 性別問題

若要達成假設要求則需考量與周圍同學之性別，此題須考量考生左方、上方、右方之同學性別，如果性別不同則獎勵，如果性別相同則懲罰。

以虛擬碼表達如圖 3-3 所示，其中 q1Reward 為自訂獎勵分數；

q1Punish 為自訂懲罰分數，q1Value 則為該題所求之適應值。

```
1  q1Value = 0 //Q1適應函數值
2  q1Reward = x //獎勵分數
3  q1Punish = y //懲罰分數
4  for(student in 1:totalStudent){
5      //與左、上、右學生性別不一樣嗎?是的話把獎勵分數加至Q1適應函數值，否則將懲罰函數加至Q1適應函數值
6      student.sex != student.left.sex ? q1Value += q1Reward : q1Value+= q1Punish
7      student.sex != student.up.sex ? q1Value += q1Reward : q1Value+= q1Punish
8      student.sex != student.right.sex ? q1Value += q1Reward : q1Value+= q1Punish
9  }
```

圖 3-3 計算性別問題適應值虛擬碼

2. 平時成績問題

考生之平時成績要能盡量相近，減低作弊行為者獲得的利益。此題需考量考生左方、左上方、上方、右上方、右方之學生，將五個方位同學的平時成績加總並平均後與中心學生成績相減取絕對值再乘上-1 作為懲罰，因此該值越接近 0 則越佳。若以虛擬碼表達計算過程則如圖 3-4 所示，其中 abs()為取決對值方法；avg()為求平均方法；generalScore 表示學生平時成績(複數型態 generalScores 表示平時成績的總和)；q2Value 則為該題所求之適應值。

```
1  q2Value = 0 //Q2適應函數值
2  for(student in 1:totalStudent){
3      score = abs(student.generalScore - avg(student.arround.generalScores))*-1
4      q2Value += score //將得分加至Q2適應函數值
5  }
```

圖 3-4 計算平時成績問題適應值虛擬碼

3. 操行成績問題

操行成績越低之學生，其座位安排能盡量靠近監考老師。此題需考量考生左方、左上方、上方、右上方、右方之學生，並且依據考生座位所在行(row)與操行成績分別乘上權重值，最後得到之分數取平方獎勵較高的分數，如此操行成績較佳之學生若被安排在較後面之位置則會使適應值較高。若以虛擬碼表達計算過程則如圖 3-5 所示，其中 q3Weight 為自訂權重值；performanceScore 為操行成績；q3Value 則為該題所求之適應值。

```
1  q3Value = 0 //Q3適應函數值
2  q3Weight = x //Q3權重值
3  for(student in 1:totalStudent){
4      score = ((student.performanceScore*q3Weight)*student.at.row*q3Weight)^2
5      q3Value += score //將得分加至Q3適應函數值
6  }
```

圖 3-5 計算操行成績問題適應值虛擬碼

4. 與同學關係問題

與周遭同學關係要越差則越不容易發生串通作弊。此題需考量考生左方、左上方、上方、右上方、右方之學生，並且給予每一位同學與其他同學關係量化值，在研究中假設 0 為關係最佳，5 為關係最糟，因此假若同學與其周遭同學關係值總和越大則越佳。若以虛擬碼表達計算過程則如圖 3-6 所示，其中 friendlyValues 為該生與周圍同學關係量化值；q4Value 則為該題所求之適應值。

```
1  q4Value = 0 //Q4適應函數值
2  for(student in 1:totalStudent){
3      //將與周圍學生關係量化值加總至Q4適應函數值
4      q4Value += student.arround.friendlyValues
5  }
```

圖 3-6 計算與同學關係問題適應值虛擬碼

5. 適應函數值

加總上述小節之適應值(q1Value、q2Value、q3Value、q4Value)，即可得到該染色體之適應函數值。

四、最佳化驗證

為驗證本研究方法及程式具有最佳化能力，在此章已依據問題特性(第三章)設定一組已知最佳解的學生資料(附件 2)，並對資料進行驗證，測試達到最佳解的狀況。

4.1 驗證使用資料

附件 2 為一已知最佳解之學生資料，其假設教室大小為 4x4，並且在最佳解的情況將有兩種情況如圖 4-1 至 4-4 所示，圖片中編號均表示為學生座號。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

圖 4-1 達到最佳解之座位表 1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

圖 4-2 達到最佳解之染色體 1

4	3	2	1
8	7	6	5
12	11	10	9
16	15	14	13

圖 4-3 達到最佳解之座位表 2

4 3 2 1 8 7 6 5 12 11 10 9 16 15 14 13

圖 4-4 達到最佳解之染色體 2

上述兩種狀況其適應函數值均為 439.1824，且各項分數如表 4-1 所示。

染色體															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
q1Value	q2Value				q3Value				q4Value				適應函數值		
270	-19.44				80.6224				108				439.1824		

表 4-1 達到最佳解染色體適應函數值

4.2 驗證結果

參數	設定值
驗證次數	500
教室大小	4x4
族群數量	6
交配長度	2
突變機率	5%
菁英政策	是
終止條件	演化 2000 世代或達到最佳解

表 4-2 驗證使用參數

驗證使用參數設定如表 4-2 所示。執行 500 次驗證後對每次基因演算法所得之適應函數值進行紀錄，並整理如圖 4-5 所示。其中橫軸為驗證次數；縱軸為該次驗證所得之適應函數值；紅線則為最佳解刻度。

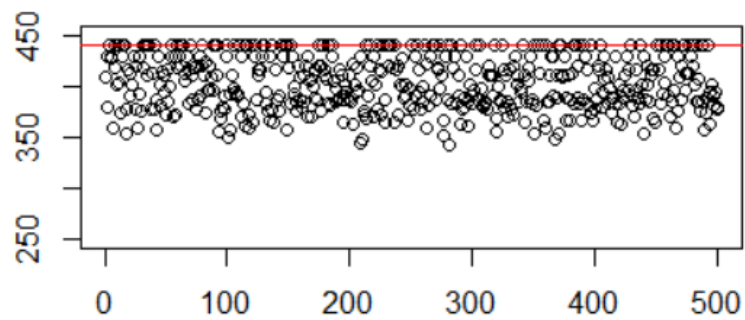


圖 4-5 500 次驗證達到最佳解之情況 資料整理:本研究

統計驗證結果數據(附件 3)，將相關結果整理如表 4-3 所示。

驗證目標	值
總驗證次數	500
最佳解適應函數值	439.1824
驗證結果	紀錄值
平均適應函數值	403.3656
適應函數值中位數	402.0512
適應函數值樣本標準差	25.74653
達到最佳解次數	92

表 4-3 驗證結果統計 資料整理:本研究

如表 4-3 所述，若在演化中適應函數值達到 439.1824 則表示得到最佳解，根據驗證紀錄結果計達到最佳解 92 次，佔總驗證次數比率 18.4%，並且計平均適應含數值達 403.3656，足以證明本研究方法及程式確實具有最佳化之能力。

五、座位分配結果與分析

本節使用表 4-2 之參數運行程式，並取一達到最佳化的結果的紀錄進行分析，並將運行結果之程式記錄檔附於附件 4。

5.1 適應函數變化值紀錄

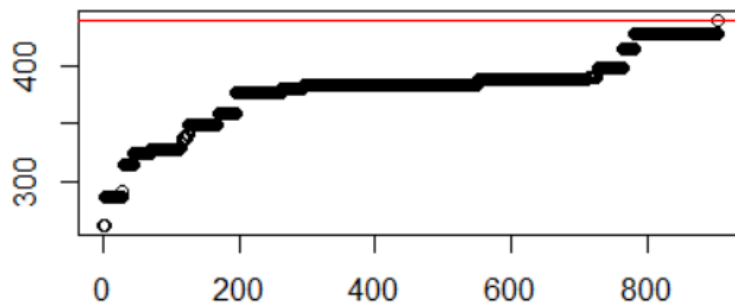


圖 5-1 達到最佳化染色體適應值變化紀錄資料整理:本研究

圖 5-1 為一達到最佳化之染色體適應值變化紀錄，其縱軸為適應函數值；橫軸為世代；紅線為最佳適應函數值刻度。可以看出約前 250 個世代其適應函數值快速成長，而 250-700 世代成長較為緩慢，可能是接近收斂或是陷入區域解，約 700 世代之後開始快速成長，可能是突破區域解並且在第 906 世代達到最佳解。

5.2 座位分配結果

由運行結果得到之最高適應值染色體如圖表 5-1 所示

染色體				
4	3	2	1	8 7 6 5 12 11 10 9 16 15 14 13
q1Value	q2Value	q3Value	q4Value	適應函數值
270	-19.44	80.6224	108	439.1824

表 5-1 達到最佳解染色體適應函數值

因染色體為編碼後之結果，若需要得到座位分配結果需對其進行解碼，解碼方式為編碼方式(2.3.1 節)之反向操作，以本次設定參數(表 4-2)與得到結果(表 5-2)作為解碼舉例；先將基因進行編號(編號 1 至 16)，並依左至右，上至下放入依 4x4 之矩陣可得如表 5-2 之座位分配結果，其中灰色網底為橫軸及縱軸編號，白色網底之數字為學生座號。

	1	2	3	4
	監考老師			
A	4	3	2	1
B	8	7	6	5
C	12	11	10	9
D	16	15	14	13

表 5-2 染色體解碼後之座位分配結果

表 5-2 依據表 5-1 經由解碼而成，為使用基因演算法得到之結果，依據表 4-2 所設定為 4 排 4 列可容 16 人佈局之教室。

六、結論

本研究使用基因演算法進行考試座位分配，實驗依據學生基本資料進行評估考量，以多項假設進行問題設計並此為目標進行座位分配，為求符合問題設定的座位分配結果。

為了驗證程式及方法具有最佳化能力，本研究使用一組已知最佳解之資料進行驗證，結果顯示程式多次達到最佳解，證明本研究使用之方法確實有最佳化之功能，並且達到問題設計目標。

本研究撰寫之基因演算法程式能在分配考試座位時從多方面考量變因，降低作弊行為的發生與傷害，且因為使用程式進行座位分配不再需要人力介入，進而提高分配座位的效率。程式設計時也帶有多項彈性可調參數方便因應各種不同環境需求。

在進行反覆驗證及測試研究使用之程式時我也發現一些可以改善的空間。如：程式缺乏使用者介面，在參數調整的時候都需要針對程式碼進行調整，若沒有程式背景的人想使用可能會遭遇困難；另一點則是學生與學生之關係的量化值，此值為座位分配的一項重要依據，但是此項數據較難取得，且難保在資料收錄過程因個人認知差異導致有所誤差。

七、附件

1. 作品連結：<https://github.com/p208p2002/use-ga-assign-examination-seat>
2. 已知最佳解學生資料：<https://ppt.cc/fDY0Mx>
3. 驗證 500 次記錄檔：<https://ppt.cc/fHH0mx>
4. 達到最佳化結果記錄檔：<https://ppt.cc/faYUmx>

八、參考文獻

1. John Henry Holland, Adaptation in natural and artificial systems, 1975
2. Mitsuo Gen and Runwei Cheng, Genetic Algorithms and Engineering Design, 1997
3. 邱宏彬、陳亦憲、楊培楷，「基因演算法在國民中學排課問題之最佳化研究」，南華大學資訊管理研究所
4. 曾文宏、吳秉威，「PCB 廠之生產排程以基因演算法」，南台科技大學工業管理研究所
5. 邱雅鈴，「基因演算法於股價走勢上之應用－以台積電為例」，國立高雄第一科技大學風險管理與保險系，2004
6. Genetic algorithms: a simple R example (<https://www.r-bloggers.com/genetic-algorithms-a-simple-r-example/>)