

## 附錄 C 遺傳演算法

### ● 遺傳演算法之介紹

遺傳演算法的理論基礎可回溯自 1859 年達爾文( Charles Darwin) 的「物種演化」(On the Origin of Species by Means of Nature Selection) 書中的「物競天擇，適者生存」的演化及淘汰觀念。在這種由自然選擇的演化機制中，生物界中的每個個體會把它們的特徵傳遞到下一代，而生物的特徵是由生物細胞內的染色體來決定的(染色體即是由基因所組成的基因鏈)，由於每個個體的特徵都不大相同，因此不同特徵的個體對環境的適應力也不大一樣，同時生物的突變及交配也會使得上下代個體之特徵不相同，而適應力較高的個體，即它們的特徵較適合於目前的環境，在後代的數目上由於競爭的緣故，適應力較高的個體的後代數目會比適應力較低的後代數目多，因此這會把整個族群的特徵引導向更適合生存於自然環境的方向發展，在長時間中，這種引導所發生的變異會越來越累積，最後演變至產生一整個特徵能適應於特別生態環境下的種族。

將這種自然界的選擇方法系統化並發展一可用之模式最早是由密西根大學的 John Holland 教授在 1975 年於 Adaption in Natural and Artificial System 文中所提出，發展出遺傳演算法搜尋技術的基本架構，並且由其學生 David Goldberg 成功地運用在工程問題上。之後，有許多研究亦證實了遺傳演算法在最佳化問題的求解上是十分有效率的，其有以下幾個優點：

1. 其可優選連續 (continuous) 及不連續 (discrete) 的參數。
2. 在優選的過程中，不需求得目標函數的導數。
3. 搜尋的方式不同於以往的單點搜尋方式，而是採用多點搜尋，因此

不容易掉入局部解 (local optimum)。

4. 可以處理多參數的優選問題。
5. 具有隱平行運算的能力，若在平行電腦中，可大量節省運算的時間。
6. 在優選複雜非線性的問題中，其演算機制可跳脫局部最佳解 (local optimum)。
7. 演算優選的結果，可提供一組最佳解，而非只有單一最佳解。
8. 參數優選需經由解碼的過程，而整個演算的機制是在解碼後的參數集合中進行，不是在參數集合本身，因此演算的機制不受問題函數型態的影響。

以上的優點，使得我們發現當傳統的最佳化方法無法解決一個問題或得到令人滿意的優選結果時，遺傳演算法便是一個很有趣且擁有很大的潛力去替代部份傳統的優選法。

對所有的問題而言，遺傳演算法並非都是一個最佳的方法，例如當在處理一具有凸函數型態，且僅有少量變數之問題時，一般傳統以微積分為基礎的搜尋法，即可比遺傳演算法快速的找到最佳解，別外一些簡單優選的問題，傳統的演算法亦都能很快的解決，然而當我們在處理實際的問題時，經常會遇見的是非凸函數且多變數型態或更複雜的問題，這是一般演算法不易解決的，而遺傳演算法就有解決此類問題的能力，且可得到近以全域最佳解。

## ● 遺傳演算法之架構

遺傳演算法將欲求解的問題變數或參數以一種類似染色體的資料結構 (Chromosome-Like Data Structure) 來編碼，並應用一些遺傳

運算元 (Operators) 如交換 (Crossover)、突變 (Mutation) 對大量的染色體作運算，運算後產生的子代除了保存親代中具優勢的特質外，也有可能因為基因的交換與突變而比親代的表現更佳。基本的遺傳演算法包含下列幾個步驟：

#### 一、將問題的變數編碼：

例如可以二進位字串 (Binary String) 的形式來表示變數，其間的轉換為二進位與十進位的對應，如將二進位字串 1001 解碼，則可對應於十進位的變數值 9，而 1100 對應於 12，1001 與 1100 可看作是兩條染色體。

#### 二、產生初始群集 (Initial Population)

以隨機的方式產生多條染色體作為初始解。



#### 三、計算目標函數值 (Evaluation)

將初始群集大量的染色體解碼後對應的變數值一一代入問題模式中，計算函數或目標函數值。

#### 四、計算適合度

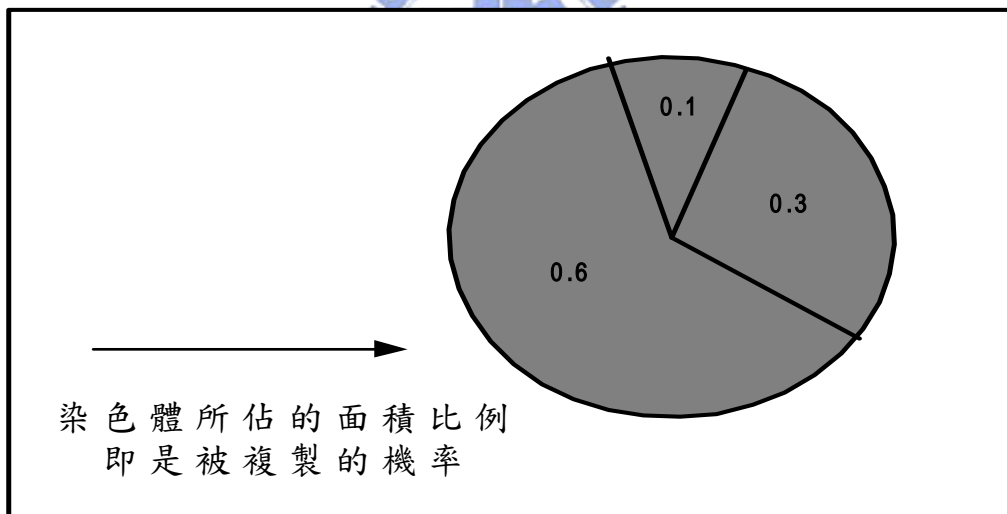
適合度愈高表示該染色體具有較優的特質，將來被複製 (Reproduction) 的機會也較大，以搜尋最大化目標值的問題來說，適合度可以目標函數來表示，若是應用於最小化目標函數之問題時，適合度函數則需由目標函數經適當的轉換而產生。

## 五、複製 (Reproduction) 或選取 (Selection)

為演化出更優良的個體，必須從原來族群中篩選出較佳的個體，組成下一代的族群，這就是複製。因此，擁有較高適應值的染色體，便有較高的機率被選擇出來進行複製。茲以下列兩種常用的方式說明：

### 1. 輪盤法 (Roulette wheel)

所謂輪盤法是假設一個可轉動的輪盤，在輪盤上劃分許多扇形區塊，區塊的面積大小正比於個體被複製的機率。因此，個體的適應值越高，適應值佔有族群適應值總和的比例也越高，在輪盤上所佔的面積也越大，而被選上的機率也越大。輪盤法的示意圖如圖附 C.1 所示。詳細步驟則如下所示：



圖附 C.1 輪盤法

- (1) 計算總適應值；為第一個個體適應值至最後一個個體適應值之總和。
- (2) 求個體適應值的份量；為個體適應值除以總適應值。
- (3) 繪製輪盤；在盤上劃分扇形區塊，區塊大小正比於個體適應值的份量。
- (4) 射靶；隨機亂數產生 0 至 1 的數字，此數落於何區，則該區個體被複製一次。
- (5) 重複 4，直到複製總數等於族群大小。

## 2. 比較選取法 (Tournament selection)

比較選取法，即是模仿自然界的生物彼此競爭情形，當某一個體的適應值愈高，其經由比較選取後，存活下來而被複製的機會愈高，此選取法有一好處，即是染色體被複製下來的機率與染色體間適合度的相對值大小無關而是取決於相對大小，因此較適合於個體間適合度值相對變化很大之問題，一般做法如下：

- (1) 依每代總數，設定一個合理的比較個數，假設為 2 個。
- (2) 每次從母代隨機選取 2 個染色體，比較其適合度較優者複製至子代。
- (3) 重複 2，直到複製總數等於族群大小。

## 六、基因交配 (Crossover) 或重組 (Recombination)

進行完複製的步驟之後，便接著要進行交配的程序。在基因演算法運作的過程中，程式會隨機產生一個交配的機率值，若是此值小於事先所定義的交配率，染色體便會進行交配的程序。將再前一步驟中所複製的個體，依亂數任意取出兩個，將其基因排列作重新的組合，

以產生新的兩個染色體，這就是交配的目的。基本上，交配的方式有三種，分別是單點交配、雙點交配和均一化交配。下面分別對這三種方式作介紹。

### 1. 單點交配

在進行單點交配時，程式會先依亂數決定一個切斷點，利用這個切斷點，將原先挑選出欲進行交配兩個的染色體切成兩部分，再將切開的部分重新組成一對新的染色體。

### 2. 雙點交配

雙點交配的步驟與單點交配類似，唯一的不同處是在進行雙點交配時，程式會先依亂數決定兩個切斷點，利用這兩個切斷點，將原先挑選出欲進行交配的染色體切成三部分，再將切開的部分重新組合成新的染色體。

### 3. 均一化交配

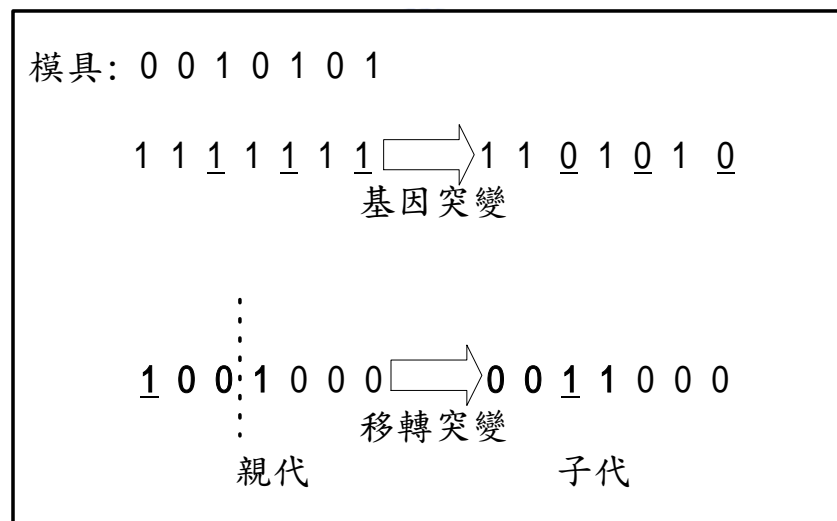
使用均一化方式進行交配時，首先會產生一個和染色體長度相等的二進位陣列，在這個二進位陣列中，每一個位元均會依亂數決定此位元的值為 0 或 1，如此一個完整的二進位陣列稱之為面具 (mask)。利用這個面具，可以決定染色體交配的位置。若是在面具中的位元值為 1 時，就進行染色體在此位元的互換；反之面具中的位元值為 0 時，則不進行互換。和單點交配、雙點交配不同處是，利用單雙點交配時，每次交配切斷點的位置不是固定的，而均一化交配則是利用面具的方式，使得同一世代之所有個體在固定的位置交配。

三種交配方式的示意圖如圖附 C.2~圖附 C.4。



## 七、突變

演化過程最後一個步驟是突變。在遺傳演算法的運作過程中，程式會隨機產生一個突變的機率值，若是此值低於事先所定義的突變率，染色體便會進行突變的程序。所謂突變，是隨機選定染色體的某些位元，將此些位元的值作 0 與 1 的互換，此步驟對染色體上的每個位元皆存在有突變的機率，可對染色體上各分段所代表的參數進行突變，將染色體的體質作一立即的改變，使搜尋的點更為零散，以防止過早收斂於局部最佳值，而無法獲得整體最佳解。突變示意圖則如圖附 C.5 所示。



圖附 C.5 突變示意圖