

# 人工智慧導論 HW2

F84086171 黃盈盛

## Section1.

此部分使用暴力解，實現方法為列出所有可能之排列組合，並計算 Cost，使用一變數紀錄最低 Cost，每當出現更低的 Cost 便更新該變數以及紀錄產生該變數的排列組合，最終一定可以得出一最低 Cost 的排列組合

## Section2.

此部分使用基因演算法，我定義了 6 個主要變數以及實作了四個主要函數。

變數部份分別為：

1. pop\_size: 族群大小，預設為 30
2. chromosomes: 紀錄族群中的染色體
3. best\_chromosome: 紀錄運算過程中 Cost 最低的染色體
4. fitness: 紀錄族群中各染色體的 Cost
5. best\_fitness: 紀錄運算過程中的最低 Cost
6. iteration: 設定遞迴次數，預設為 100

函數部份分別為：

1. select

此函數主要用於選擇要用於交配的染色體，我實作的演算法將根據 Exponential distribution 的結果挑選染色體，在此之前會先將 chromosomes 根據 fitness 做遞增排序，而 Exponential distribution 容易出現接近 0 的值，意即將有更高的機率選到 Cost 較低的染色體，且各染色體可選擇複數次，如此可留下更多好的染色體。

2. crossover

此函數主要做交配，交配方法為每兩個染色體為一組，以下稱染色體 1、2，並設定一斷點，斷點以左一段，斷點以右一段，將染色體 1 的左半段與染色體 2 的右半段組合起來，剩餘部分同理，如此可得兩個新染色體，不過由於此題規定染色體中數字不可重複，因此需檢驗新染色體是否有數字重複，並將重複的數字更換成未出現的數字才符合規範。

3. mutate

此函數主要做突變，實作方式為以一機率決定是否突變，突變的行為是染色體基因反轉，如 123 會變為 321。

4. CalFitness

此函數計算各染色體的 fitness，作法為計算各染色體的 Cost，Cost 越

低，fitness 越高。

將上述函數以

select → crossover → mutate → CalFitness 的順序執行數個循環，循環數為給定的中止條件，執行完指定循環次數即會結束。

最終，可得到運算過程中得出之最低 Cost 的排列組合(染色體)

## Section3.

### 如何將問題表示為染色體：

由於題目規定數字不能重複，因此我的想法是以不同的排列組合作為染色體，並在交配中改變排列，逐漸使排列組合趨近最佳解。

### 不同決策所造成的變化：

在 P2 中我對 crossover 斷點、mutate 機率做了幾種，變形以下將列舉變形後的結果，為分析各部分變形產生的影響我將以基本作法為基礎，一次改變一個部分。

以下作法之最終結果以範例測資中最複雜的 7 人 7 工作為例，每種作法將執行 10 次，紀錄平均到達局部最佳解所需遞迴次數、平均 Cost，以及局部最佳解為全局最佳解的比例。

### 基本做法：

Crossover: 斷點採隨機，且為均勻分布，因此，假設染色體長度為 3，則斷點隨機出現在 0~3 之間，將染色體切成(0,3)(1,2) (2,1)(3,0)等情況，而後步驟如 P2 中所述。

Mutate: 給定機率為 10%，當需要突變時，將染色體做 reverse，假設染色體為(0,1,2)，則突變後變為(2,1,0)

### 最終結果：

平均局部最佳解所需遞迴次數:35.6

局部最佳解為全局最佳解比例:60%

平均 Cost:1.512

變形一:Crossover 採固定斷點，使切分後左右半段染色體基因數一

致或差 1。

### 最終結果：

平均局部最佳解所需遞迴次數:23

局部最佳解為全局最佳解比例:70%

平均 Cost:1.49

變形二:Crossover 採常態分佈斷點，即有較高機率斷在中間，但仍

有機率使切分後的染色體左右部分相差較大。

最終結果:

平均局部最佳解所需遞迴次數:34.5

局部最佳解為全局最佳解比例:70%

平均 Cost:1.47

變形三:突變機率調為 50%。

最終結果:

平均局部最佳解所需遞迴次數:51.1

局部最佳解為全局最佳解比例:70%

平均 Cost:1.50

變形四:突變機率調為 100%。

最終結果:

平均局部最佳解所需遞迴次數:41.6

局部最佳解為全局最佳解比例:50%

平均 Cost:1.50

變形五:pop\_size 調為 10。

最終結果:

平均局部最佳解所需遞迴次數:49.6

局部最佳解為全局最佳解比例:30%

平均 Cost:1.61

變形六:iteration 調為 10。

最終結果:

平均局部最佳解所需遞迴次數:4.6

局部最佳解為全局最佳解比例:0%

平均 Cost:1.78

由上述變化造成之結果可以總結出以下三點。

1. 族群數量大小與遞迴次數會顯著影響平均 Cost 與找到最佳解的機率，當族群越小，欲找到最佳解所需時間越長，當給定的時間越短(遞迴次數越少)，族群越大越容易找到最佳解。
2. 斷點使用 normal 最佳，固定次之，隨機最差，我想這是由於隨機相比於前兩者有較高機率產生較小的變異性，也因此適應度浮動較小，因此在給定相同族群大小與演化時間的情況下較難演化出最佳解，固定則能確保每次交配都能有最大的變異性，適應度浮動最大，但也因此，優良的基因可能無法很好的得到延續，易被較差的基因汙染，而 normal 則綜合了兩者，有較高可能產生比較大的變異性，但同時有一定機率能保存優良基因，因此 normal 得到了較好的平均 Cost。
3. 突變機率不宜太大，由 10%、50%、100%突變率所造成的結果中可以觀察到，10%的平均 Cost 最小，其餘並列，100%突變率找到最佳解的機率較低，50%則是找到最佳解的平均時間最久，突變率過高會導致族群適應度極其不穩定，也因此越難找到最佳解，演化時間也會顯著增加。

## Section4.

### 暴力解：

**優點：**必定可以找到最佳解，且實作簡單。

**缺點：**當工作數以及人數上升，時間複雜度以階乘上升，將難以在有限的時間內計算完成。

### 基因演算法：

**優點：**不論工作數以及人數如何上升都必定可以在有限時間中計算出局部最佳解(也可能為全局最佳解)，在多數時候只需產生一個足夠好的解即可使用，不一定非得最佳解。

**缺點：**可能陷入局部最佳解導致無法找到全局最佳解、將實際問題抽象化使其可以應用基因演算法有一定難度，且涉及函數眾多、作法多樣，需決定哪種情況更適合哪些做法。

**實作心得：**在查了很多基因演算法的資料後我發現真的有很多從框架上就截然

不同的實作方式，在這次實作中我也是選擇了最基本也最經典的版本，即便如此，各步驟中依然存在許多不同做法，在各步驟方法的選擇、實踐上也是遇到了些許問題，且原本在我的預想中基因演算法應該會很容易陷入局部最佳解，導致結果不好，不過實際做出後發現其實效果很不錯，大部分測資都能找到最佳解，在最複雜的測資中也能有 70%的機率找到最佳解，也許是因為還不夠複

雜，但能實作出一個效能如此好的結果我覺得相當有成就感。