人工智慧導論
Homework 2

系所: 資訊所

學號: P76114545

姓名: 張少鈞

暴力法實作

問題分析

在此工作指派問題中，所有擁有的任務數量與工作人員數量相同，也就是有 N 件任務會由 N 個人來處理，並且每個人必須處理一件任務，每件任務也必須被一個人處理，因此共有 N!種指派方式，

從上述的分析可以發現，欲尋找的最佳指派方式一定存在於上述的 N!種指派方式中，因此最簡單的暴力解即為將所有指派方式列出，並從中尋找最低成本的指派方式。

找尋所有組合

我們可以將所有的排列組合以樹的方式來進行表達（如 Figure 1），每條從樹根(root)到葉節點(leaf node)的路徑即為一組可能的排列組合。將排列的問題表達成樹的形式後，利用深度優先搜尋(Depth-First Search, DFS)的概念來搜尋所有的解。

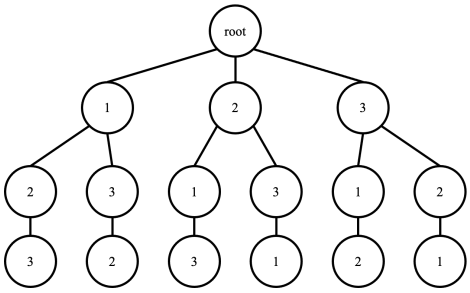

Figure 1 將 N=3 的所有組合以樹的方式表達

基因演算法實作

基因演算法是一種受到自然選擇機制所啟發的演算法，將問題的解模擬成一條染色體(chromosome)，進而模仿生物學中的交配(crossover)、突變(mutation)等現象，並根據適應度(fitness)來表達染色體的優劣。整體的流程如 Figure 2。

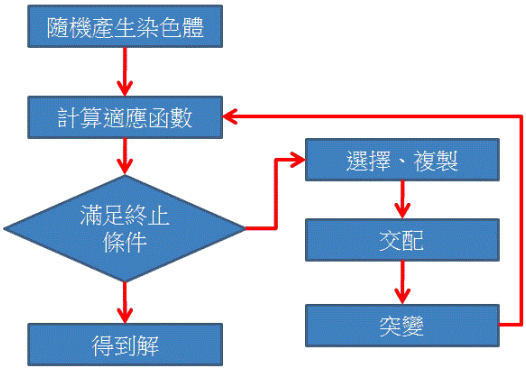

Figure 2 基因演算法流程圖

在執行基因演算法之前，要先將我們的問題表達成染色體的形式，每條染色體代表的是工作指派中的一組解，此解由一串整數來進行表示，在這串數字中第 i 個數值代表第 i 個任務要分配給哪位工作人員。

將問題表達成染色體後，會利用亂數來隨機產生多組染色體，這些染色體會當作第一輪的母體(population)，從中挑選染色體成為父母(parent)，挑選的方法可以依據適應度來進行選擇，但本次採用隨機選擇的方式來挑選父母。

將父母兩兩進行交配已產生後代(offspring)，本次交配的方式採用部分匹配交配(Partially matched crossover, PMX)的方式來進行。在整數編碼方式的染色體中，PMX 擁有比大多數的交配方法還要優異的表現。PMX 首先會隨機挑選染色體的一段區域，將父母區段內的基因進行對調產生後代，為了避免後代的染色體內出現重複基因，需記錄下區段內染色體的對應關係，依據此對應關係來映射使染色體合法化(基因數字不重複)，整體的流程圖如 Figure 3。

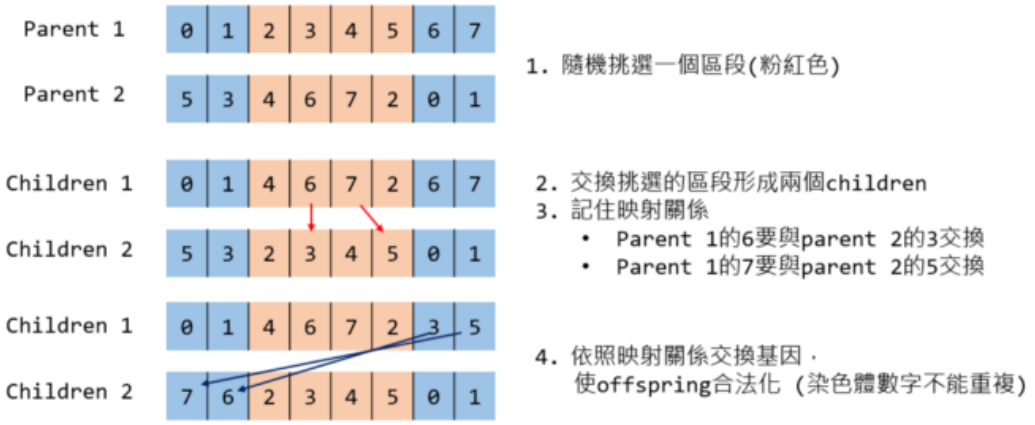

Figure 3 PMX 流程與範例

為了增加染色體的多樣性，在交配完產生後代之後，隨機選擇染色體進行突變，用意在擴大搜尋解的空間。本次採用的突變方式為反轉突變(Inversion mutation)，隨機挑選染色體的一段基因，將此段基因逆序列排列如 Figure 4。

Figure 4 反轉突變流程與範例

在產生完新的母體後，依據適應度來評估染色體的生存能力，在工作指派問題中成本越低的指派方式越好，由此可知染色體的適應度與我們的成本成反比，因此將適應度的計算公式設計為成本的倒數。

最後在決定基因演算的中止條件設計上，本次採用最簡單的方式來實作，也就是設定迭代次數，當達到設定的迭代次數後終止基因演算法的計算。

基因演算法實作觀察

在透過基因演算法求解工作指派問題實發現，每次求出來的解並非是最佳解(optimal solution)，重新分析基因演算法後發現，在每次迭代所產生出新的母體會完全覆蓋掉前一次的母體，新產生的母體適應度無法進行掌握，而舊的母體中高適應度的染色體卻不會被保留下來，因而每次迭代後的結果並不一定會比原本的好，如 Figure 5。針對以上的問題進行基因演算法的改良，為了能保留前一次迭代中高適應度的染色體，在產生新的母體時將舊的母體納入考慮與新生代的染色體進行比較，將高適應度的染色體進行保留，低適應度的染色體則捨棄，如 Figure 6。

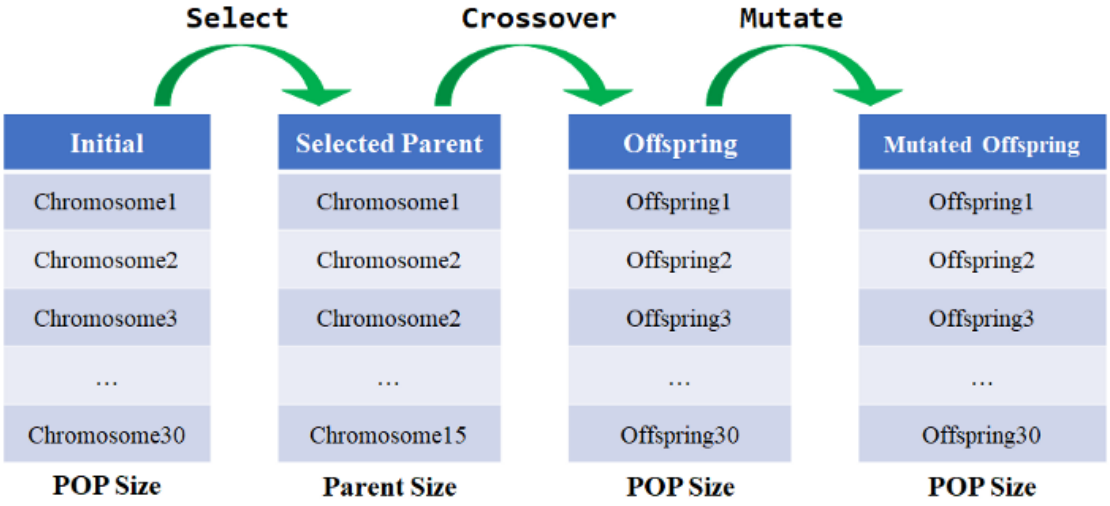

Figure 5 改良前的基因演算法示意圖

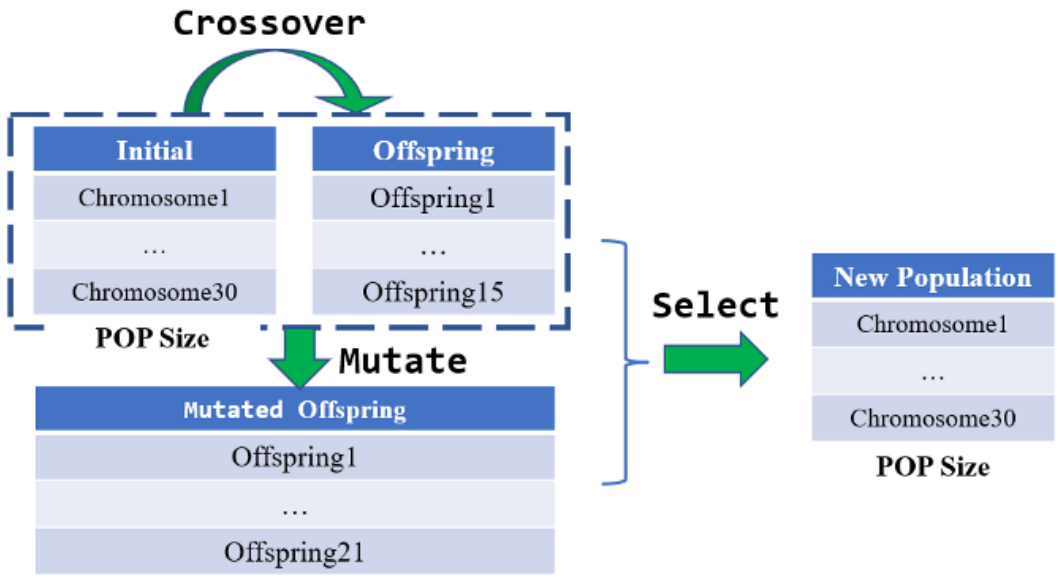

Figure 6 改良後的基因演算法示意圖

針對此兩種不同架構的基因演算法進行分析，將每一次迭代所取得的解進行

記錄（如 Figure 7），發現隨著迭代的次數增加，原始的基因演算法明顯出現大量偏移，相比之下，改良過後的演算法能夠穩定的維持在最佳解。

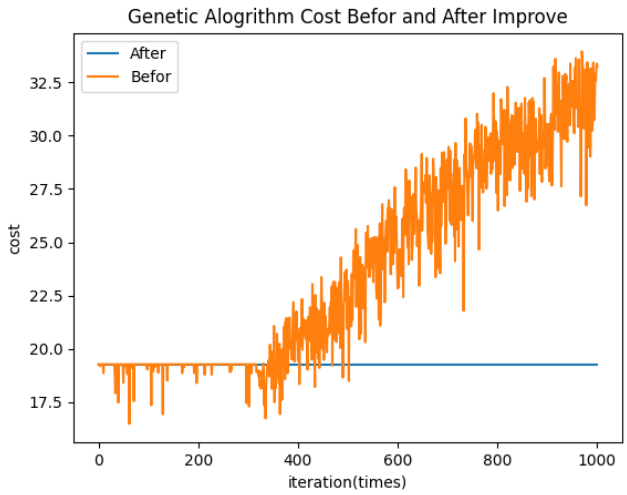

Figure 7 基因演算法改良前後比較

總結

在解決工作指派問題上，在相同的任務數與迭代次數下，兩者方法的執行時間均為定值，因此將任務數與迭代次數分別定為 10 與 1000 並測試兩者演算法的執行時間（單位為秒），分別紀錄 10 次執行時間如 Table 1，可以得到兩者的平均執行時間分別為 22.718 與 0.326 秒，暴力法所需的平均執行時間大約是基因演算的 70 倍，但兩者在這 10 次測試中的正確率均為 100%。基因演算法雖然隨著任務數的增加，迭代的次數也要隨之增加，但求解效率仍然比使用暴力法好。

Table 1 暴力法(BF)與基因演算法(GA)運行時間比較表（單位為秒）。總共測試 10 次，每次輸入的任務數均為 10，並且將 GA 的迭代次數定為 1000 次。

次數	1	2	3	4	5	6	7	8	9	10
BF	22.50	22.26	24.31	23.81	22.34	22.31	22.34	22.28	22.31	22.72
GA	0.32	0.33	0.32	0.32	0.33	0.33	0.32	0.33	0.33	0.33

參考資料

[1] https://medium.com/qiubingcheng/以 python 實作基因演算法-genetic-algorithm-ga-並解決工作指派問題-job-assignment-problem-jap-b0d7c4ad6d0f