

Homework 3: A Personal Assignment Problem 閱讀心得

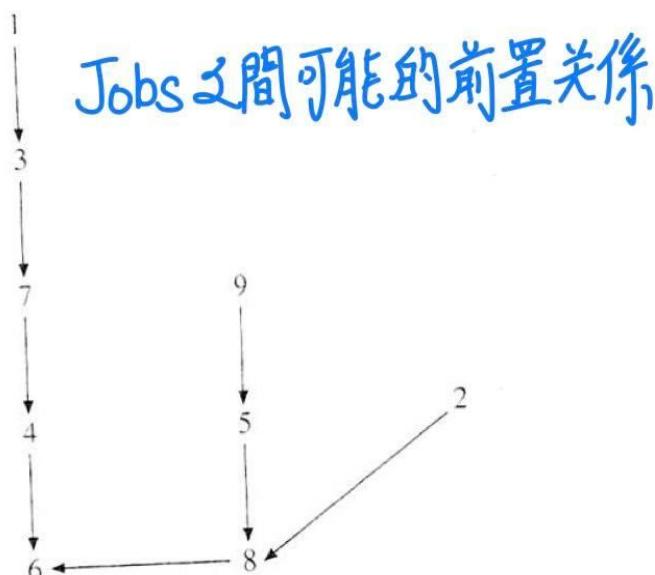
R10942152 游家權

二、問題定義

有一群員工 $P = \{P_1, P_2, \dots, P_n\}$ ，與一些工作 $J = \{J_1, J_2, \dots, J_n\}$ 。我們需要指派員工工作，使得每個員工只能有一個工作，一個工作也只能指派給一個員工，也不能兩個員工作同一個工作，員工跟工作基本是一對一的映射關係。另外，題目給定 Cost matrix，裡面設定了任某個員工做某個工作所需要花費的代價，根據這個 Cost matrix 我們可以輕易的找出某個指派方式一共會付出多少代價，而我們的目標就是找出某個指派工作的方式讓代價最小。下圖是 Cost matrix 的範例。

		Jobs			
		1	2	3	4
Persons					
	1	29	19	17	12
	2	32	30	26	28
	3	3	21	7	9
	4	18	13	10	15

除了前面描述的限制之外，題目還有一個條件。就是某些工作之間是有前置條件的關係，也就是要先完成A工作，才能開始B工作，或是如果不先完成A,B工作，就不能開始做C工作等等這類條件。接著，我們可以把n個工作者想成是1~n的不同時刻到達的員工，這些員工一旦被指派任務就會馬上完成工作，讓需要這個前置條件的工作可以在下個時刻被考慮，依此類推。



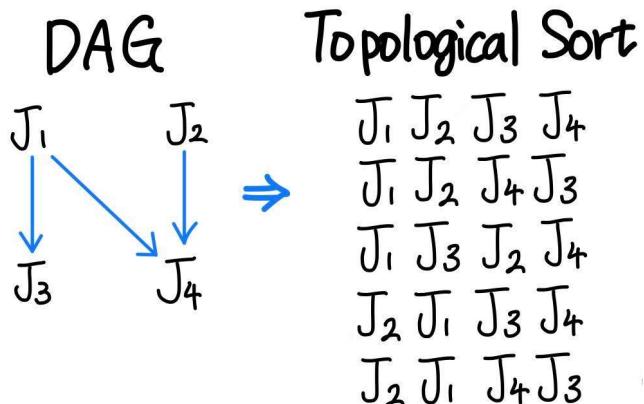
綜上所述，我們每次都是依序指派員工 P_1, P_2, \dots, P_n 工作，而 P_1 能做的工作就是那些不需要前置條件的工作， P_1 指派完成後輪到指派 P_2 ，而 P_2 能做的工作就是不需要前置條件的工作以及因為 P_1 工作完成而新解鎖的工作。長此以往，當指派完 P_n 工作後，答案就出來了。

二、觀察問題

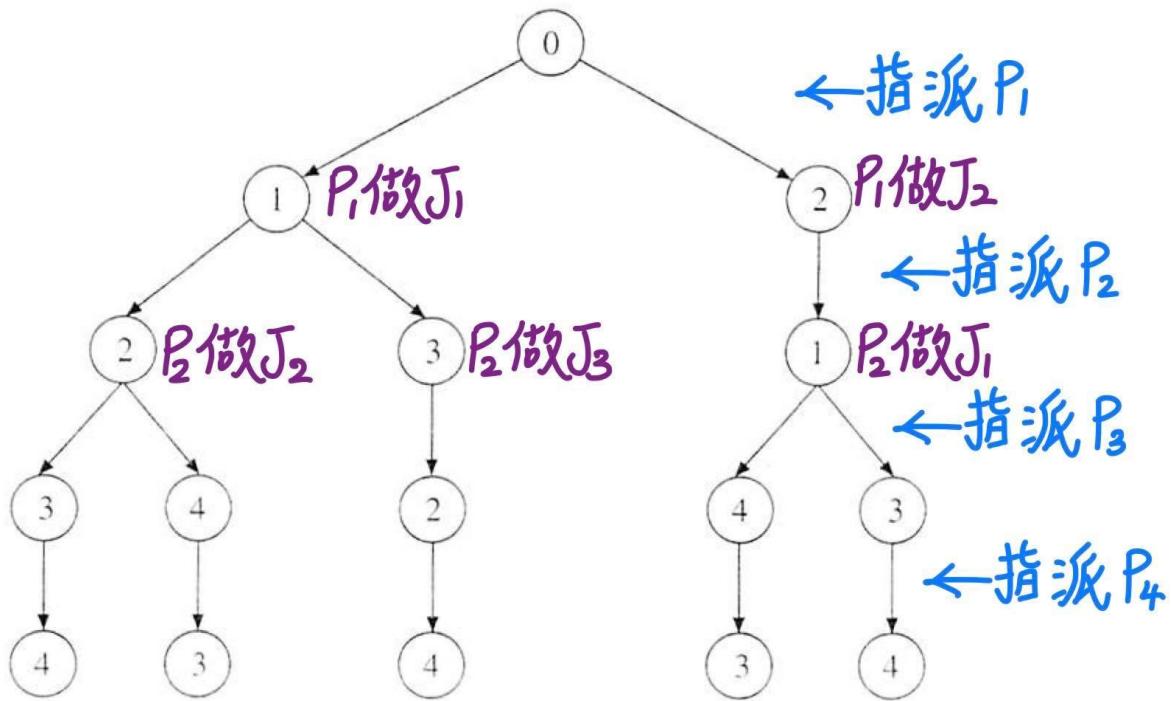
Observation 1: Topological Sort of Jobs

當題目給定了一種工作之間的前置條件關係後，我們可以用這個關係畫出一張無環有向圖(Directed Acyclic Graph)，在這張圖上如果某個工作的In-degree=0，代表這是沒有前置條件的工作，如果In-degree=x，代表做這個工作之前，要先完成前面x個工作才行。

我們可以在這張無環有向圖上找出topological sort的所有可能的指派方式



而這些可能的指派方式又可以被畫成Tree的表現形式，Tree上的每個邊表示指派某個人去做某個工作，因此如果只有N個工作，則Tree的最大深度就是4。Topological sort 有幾種可能，這棵樹就會有幾個Leaf。



經由這個Observation 我們成功的把指派工作的問題，轉換成一棵樹上的路徑探索，往後的問題解法都會在這棵樹上進行操作。

Observation 2: Low bound from Reduced Cost Matrix

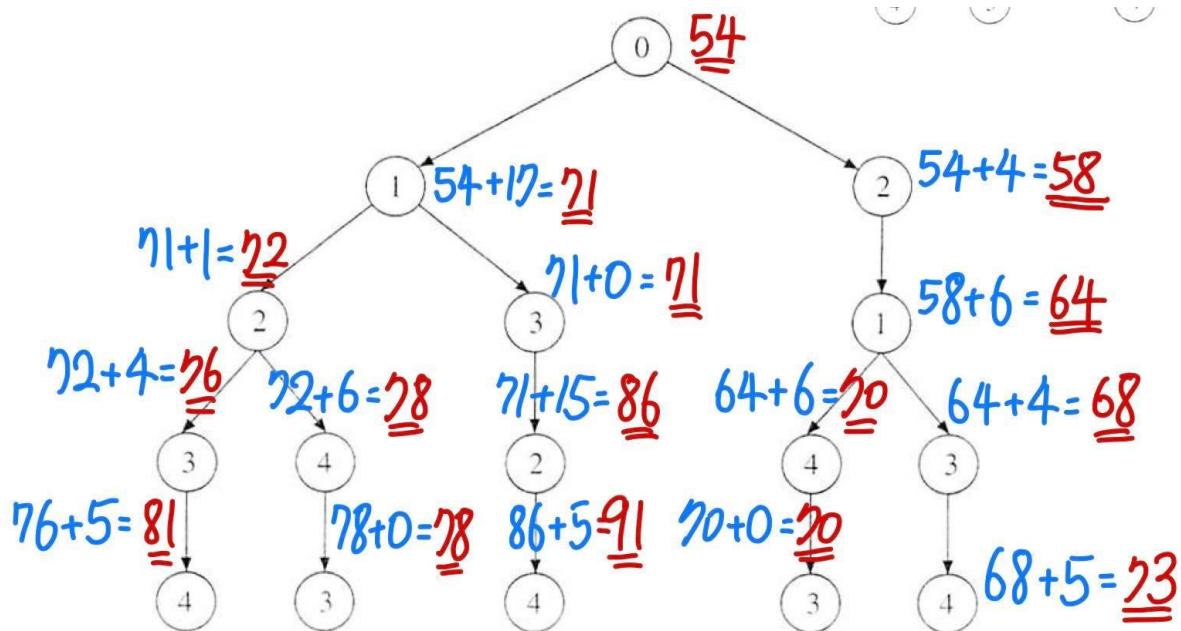
接下來我們要想辦法估計樹上節點的Lower Bound, 也就是這個節點的子樹至少會花費多少的Cost。考慮Cost Matrix中的橫列，橫列裡面的最小值，代表這個員工執行任務最少會花費的代價。考慮Cost Matrix中的直行，直行裡面的最小值代表這個任務要完成至少要花費的代價。

為了簡化Cost Matrix, 我們可以把各個橫列的每個元素減掉橫列裡面的最小值，直行也可以做一樣的操作，由於減掉的數字跟怎麼選擇指派方式完全沒有關係，單純只是把Cost的"公因數"提出來，簡化Cost Matrix的形式，換言之，把Cost Matrix換成Reduced Cost Matrix完全不會影響Solution。下圖是簡化的Cost Matrix的範例。

Cost Matrix				
Persons	Jobs	Reduce Row	Reduce Column	Reduced Cost matrix
1	1 2 3 4	$(17, 7, 5, 0) + 12$	$(17, \cancel{4}, 5, 0) + 12$	$\begin{bmatrix} 17 & 4 & 5 & 0 \\ 6 & 1 & 0 & 2 \\ 0 & 15 & 4 & 6 \\ 8 & 0 & 0 & 5 \end{bmatrix}$
2	2 3 0 2 8	$(6, 4, 0, 2) + 26$	$(6, 1, 0, 2) + 26$	
3	3 2 1 7 9	$(0, 18, 4, 6) + 3$	$(0, 15, 4, 6) + 3$	
4	18 13 10 15	$(8, 3, 0, 5) + 10$	$(8, 0, 0, 5) + 10$	
with total reduced cost: $12 + 26 + 3 + 10 + 3 = 54$				

這時我們可以發現任何決定都還沒做之前，整個指派任務的最小Cost已經是54了，這個54就是我們的Decision Tree中的Root的Lower bound。而往後的node的Lower bound

也可以用Reduced Cost Matrix 直接得到，因此我們可以用這個方法得到整棵樹每個節點的Lower bound。下圖是找出所有節點的lower bound的範例。



三、解法敘述

Step1: 將題目給定的Cost Matrix, 用前面Observation 2 的方法, 化簡成Reduced Cost Matrix。

Step 1:

Job 1 2 3 4 5

$$\begin{array}{c}
 \text{Job 1 2 3 4 5} \\
 \begin{matrix}
 1 & 17 & 13 & 4 & 23 & 12 \\
 2 & 32 & 30 & 45 & 19 & 31 \\
 3 & 7 & 11 & 12 & 3 & 15 \\
 4 & 10 & 17 & 13 & 9 & 11 \\
 5 & 8 & 19 & 9 & 6 & 7
 \end{matrix}
 \xrightarrow{\quad} \begin{matrix}
 13 & 9 & 0 & 19 & 8 \\
 13 & 11 & 26 & 0 & 12 \\
 4 & 8 & 9 & 0 & 12 \\
 1 & 8 & 4 & 0 & 2 \\
 2 & 13 & 3 & 0 & 1
 \end{matrix}
 \begin{matrix}
 +4 \\
 +19 \\
 +3 \\
 +9 \\
 +6
 \end{matrix}
 \xrightarrow{\quad} \begin{matrix}
 12 & 1 & 0 & 19 & 7 \\
 12 & 3 & 26 & 0 & 11 \\
 3 & 0 & 9 & 0 & 11 \\
 0 & 0 & 4 & 0 & 1 \\
 1 & 5 & 3 & 0 & 0
 \end{matrix}
 \begin{matrix}
 +4 \\
 +19 \\
 +3 \\
 +9 \\
 +6
 \end{matrix} \\
 \text{Person}
 \end{array}$$

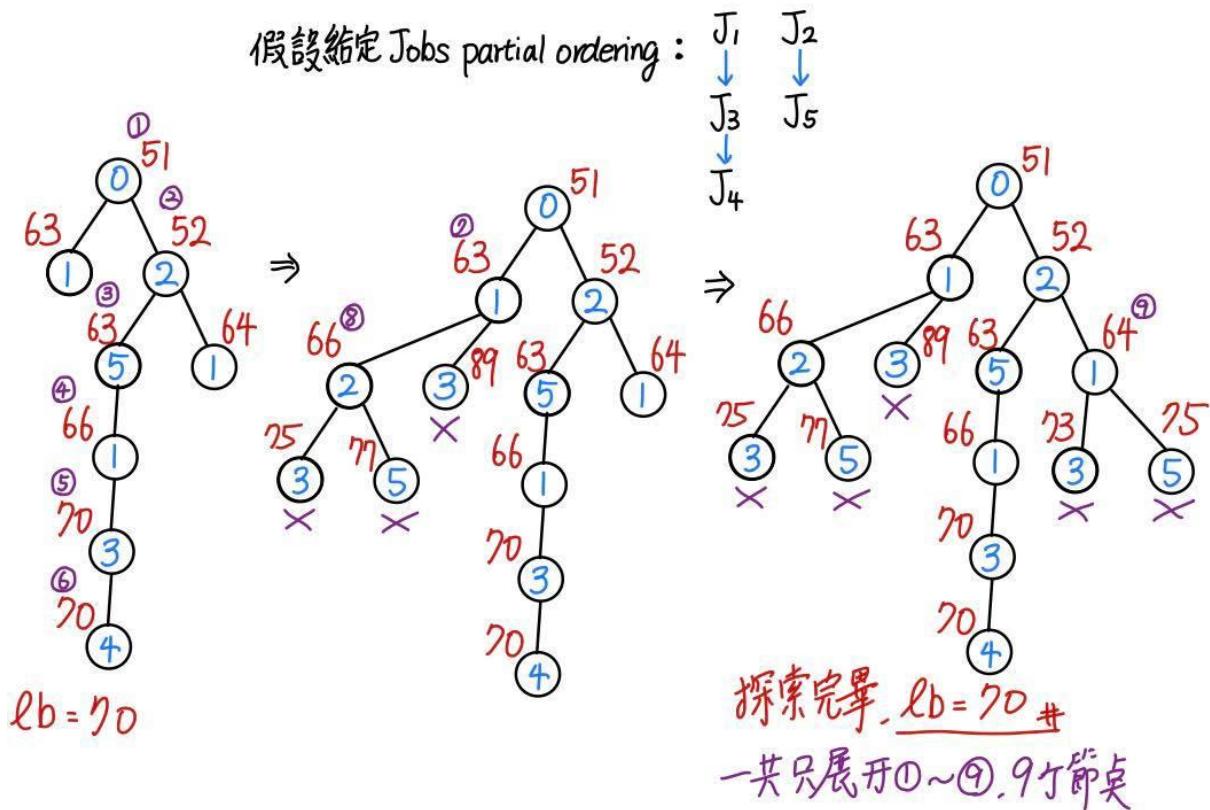
Reduced Cost:
with $4+19+3+9+6+$
 $1+8+1 = 51$

Step2: 從樹的根部開始探索Decision Tree, 並且把剛剛Reduced Cost Matrix所提出來的Lower bound 當作Lower bound 的初始值。每往下一層探索就需要去確認Jobs之間的Topological order, 找出有可能的分支是那些, 而每個分支的Lower bound都可以用

Observation 2 的方法求得。另外，探索的順序應該先展開Lower bound 較小的節點，因為他們更有可能找到代價更小的解。

Step3: 當抵達Leaf時，再次更新Lower bound，並且把大於這個Lower bound的節點全部拋棄，因為已經確定這個Leaf的解一定比這些還沒展開的子樹的所有解都來的好。

Step4: 重複Step2, 3直到Decision Tree內已經沒有可以展開的節點，回傳Lowerbound就是答案。



時間複雜度分析:

Branch and Bound的問題大多都是在解NP-Complete的問題，因此時間複雜度大多如同這個問題都是Exponential，雖然沒有保證時間會收斂到合理的bound，但我認為Branch and Bound有趣的地方就在於大多都會執行的很快，換言之，大部分的節點都會成功的被拋棄而不會展開，雖然沒有辦法用嚴謹的數學證明到底時間複雜度會Bound在哪裡，但是我們還是可以有信心地說這是一個很有效率的演算法。

四、讀後心得

看了這篇文章跟實作了期末考題的B&B之後，我發現如何去尋找Lower bound 是這類問題的關鍵，只要Lower bound能快速的上升，我們就可以確保在展開大部分的節點之前，我們已經可以把這些節點全部丟掉，從而得到有效率的演算法。也因為這樣，在設計B&B的時候，也要想辦法盡快探索到樹葉，因為一旦找出一個確切的解，我們就可以用它來更新Lower bound，讓接下來探索更加順利，捨棄更多不必要的節點展開。

另外，我也在實作過程中發現B&B的表現的不穩定性，有些問題需要展開30%的節點，有些只需要展開0.000001%的節點，雖然兩者因為問題大小不同，所以都在合理時間內解完，但這足以說明B&B的效能還是會受到"運氣"影響。