IE325K 作業研究 (下)

個案研究：範例四 單機排程問題

組別：第一組

組員：10822142 曾可欣、11024323 廖家廣、11024325 余尚晉、
11024327 莊子毅、11024333 邱寶樟

一、單機排程問題介紹

1.1 定義

單機排程問題(Single Machine Scheduling Problem)是一種在生產管理和作業研究中常見的問題，其目的是在一台機器上對一組作業進行排程，以最佳化某一目標函數。這些作業可能具有不同的處理時間、優先順序、到達時間、期限等約束條件。單機排程問題的通常包括幾種：最小化總完工時間、最小化最大完工時間、最小化總流程時間以及最小化加權總完工時間，而針對單機排程問題的解法有貪婪演算法、動態規劃以及啟發式演算法。單機排程問題是排程理論中的基礎問題，其解法和思路常被應用到更複雜的多機排程問題或生產計劃中。

1.2 文獻介紹

林家賢(2011)研究將機台視為生產資源，探討單一機台(Single-machine)的生產排程問題。假設每個工件有不同的可加工時間與交期，且工件只能在可加工時間到工件交期的時段內加工，不允許延期加工。其著重的生產排程問題為最大化權重排程問題。此研究提出三個方法來求解最大化權重排程問題，分別是：啟發式演算法(Heuristic Method)、混合整數規劃(Mixed Integer Programming, MIP)以及分枝界限法(Brand-and-bound Method)。其中，啟發式演算法可求得一個初始解，供混合整數規劃及分枝界限法利用。

張旭翔(2013)使用礦工基因演算法(Minor Genetic Algorithms, MGA)解總加權完工時間最小化之單機生產排程問題，並以基因演算法(Genetic Algorithms, GA)做為比較基準。實驗結果顯示，在一定時間內使用礦工基因演算法所求解出的值遠比基因演算法優良許多，收斂速度及收尋能力較為佳。目標是希望使總加權完工時間(Total Weighted Completion Time)最小，利用礦工基因演算法求得最佳的工作排序，用以達成目的。

陳晁誠(2016)致力於解決具有變動維修作業的單機排程問題，以最小化平均延誤時間、最大延遲時間、總流程時間和平均延遲時間等四個常見目標函數。在這些排程問題中，變動維修作業需在給定期限前開始，其所需時間是開始維修時間的非遞減函數，該論文針對此四個目標函數的具有變動維修作業單機排程問題，分別提出可求得最佳解的多項式時間演算法。

二、數學規劃模型

2.1 參數定義

P_t：第 t 個順序之加工時間，$t = 1, 2, \ldots, N$。

W_t：第 t 個順序之權重，$t = 1, 2, \ldots, N$。

J ：工作，$j = 1, 2, \ldots, N$。

T ：加工順序，$t = 1, 2, \ldots, N$。

N ：工作的數量。

M：極大值。

2.2 決策變數

x_{jt}：第 j 項工作在第 t 個順序是否進行加工，有則為 1；若無則為 0。

c_{jt}：第 j 項工作在第 t 個順序之完工時間。

f_j ：第 j 項工作之完工時間。

s_j ：第 j 項工作之開始加工時間。

2.3 最短平均完工時間

2.3.1 目標式

式(1)為最小化平均完工時間(Minimum Average Completion Time, MACT)。

$$\text{Minimize } \frac{\sum_{j=1}^{N}\sum_{t=1}^{N} c_{jt}}{N} \tag{1}$$

2.3.2 限制式

式(2)代表每項工作在每個順序之完工時間，式(3)則代表每項工作之完工時間。式(4)令工作皆得以在時刻為零時開始加工，式(5)為決定每項工作的開始加工時間。式(6)說明每項工作只能被加工一次，式(7)說明每個順序只能有一項工作進行加工。式(8)與式(9)確保該變數需為非負數。式(10)則為本模型之二元限制式。

$$c_{jt} \geq f_j + M \times (x_{jt} - 1) \qquad \forall j \in J, t \in T \tag{2}$$

$$f_j \geq s_j + \sum_{t=1}^{N} (P_t \times x_{jt}) \qquad \forall j \in J \tag{3}$$

$$s_0 = 0 \tag{4}$$

$$s_{j+1} \geq s_j + \sum_{t=1}^{N} (P_t \times x_{jt}) \qquad \forall j \in (J-1) \tag{5}$$

$$\sum_{t=1}^{N} x_{jt} = 1 \qquad \forall j \in J \tag{6}$$

$$\sum_{j=1}^{N} x_{jt} = 1 \qquad \forall\, t \in T \qquad (7)$$

$$c_{jt} \geq 0 \qquad \forall\, j \in J, t \in T \qquad (8)$$

$$f_j, s_j \geq 0 \qquad \forall\, j \in J \qquad (9)$$

$$x_{jt} \in \{0, 1\} \qquad \forall\, j \in J, t \in T \qquad (10)$$

2.3.3 求解結果

根據 2.1 節至 2.3.2 小節建構之 MACT 整數規劃模型，藉由電腦規格 Apple M1 CPU、記憶體 8GB、作業環境 macOS Sonoma 14.4.1，配合 Gurobi 10.0.2 版求解。共產生 84 個變數(整數變數 48 個、連續變數 0 個、二元變數 36 個)與 60 條限制式，且僅 0.02 秒即計算出最短平均完工時間為 41 分鐘，如圖 1 所示。此外，進一步將上述求解結果引入 Plotly 繪圖套件，透過視覺化呈現圖 2 之排程結果後，可以得出本研究 MACT 之加工順序依序為 3 → 5 → 6 → 4 → 1 → 2，與範例之表 4-2 結果相同。

```
Explored 1 nodes (114 simplex iterations) in 0.02 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 4: 41 44.5 47.8333 63.6667

Optimal solution found (tolerance 1.00e-04)
Best objective 4.100000000000e+01, best bound 4.100000000000e+01, gap 0.0000%
Number of variables:  84 (integer: 48, continuous: 0, binary: 36)
Number of constraints:  60
Objective value:  41.0
```

圖 1、MACT 求解結果

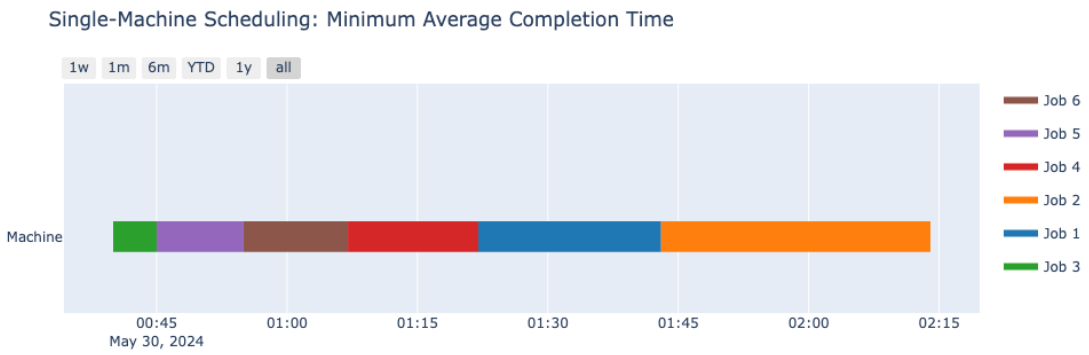

圖 2、MACT 排程結果

2.4 最短平均加權完工時間

2.4.1 目標式

式(11)為最小化平均加權完工時間(Minimum Average Weighted Completion Time, MAWCT)。

$$\text{Minimize} \quad \frac{\sum_{j=1}^{N}\sum_{t=1}^{N}(W_t \times c_{jt})}{N} \tag{11}$$

2.4.2 求解結果

此階段將原先 MACT 模型的目標式由式(1)更改為式(11)；其餘限制式則維持不變，沿用式(2)至式(10)。求解結果如圖 3 所示，共產生 84 個變數(整數變數 48 個、連續變數 0 個、二元變數 36 個)與 60 條限制式，並耗時 0.04 秒得出最短平均加權完工時間為 103.17 分鐘。接著同樣利用 Plotly 套件繪製圖 4 之排程結果，能得出本研究 MAWCT 之加工順序依序為 3 → 4 → 6 → 5 → 2 → 1，與範例之表 4-3 結果 3 → 6 → 4 → 5 → 2 → 1 存在些許差異。

```
Explored 1 nodes (220 simplex iterations) in 0.04 seconds (0.01 work units)
Thread count was 8 (of 8 available processors)

Solution count 5: 103.167 104 113.833 ... 218.167

Optimal solution found (tolerance 1.00e-04)
Best objective 1.031666666667e+02, best bound 1.031666666667e+02, gap 0.0000%
Number of variables:  84 (integer: 48, continuous: 0, binary: 36)
Number of constraints:  60
Objective value:  103.17
```

圖 3、MAWCT 求解結果

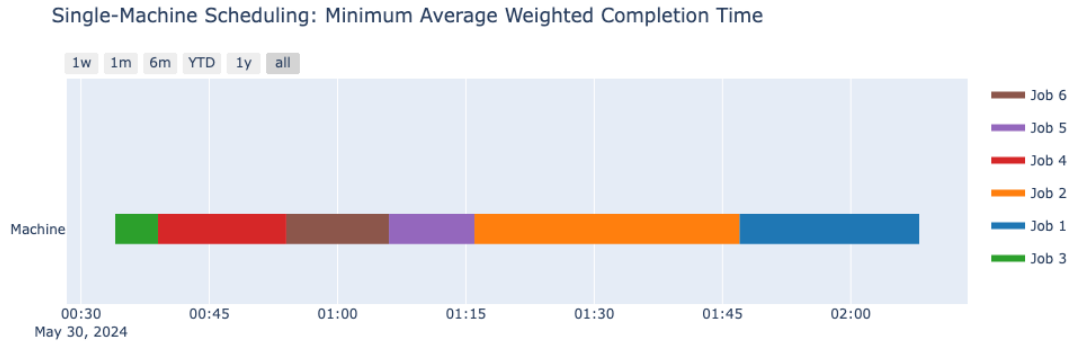

圖 4、MAWCT 排程結果

圖 5 為兩者詳細的比對過程，可發現在工作 6 和工作 4 的先後順序不同；然而，將其各自工作的完工時間與權重，帶入目標式執行驗證，式(12)與式(13)依序為範例表 4-3 與本研究求解結果，經數值比對皆可得出相同的最佳解為 103.17 分鐘。由此可證，本研究之結果為另一組可行解。

$$\frac{1\times 94 + 2\times 73 + 5\times 5 + 5\times 32 + 3\times 42 + 4\times 17}{6} = 103.17 \tag{12}$$

$$\frac{1\times 94 + 2\times 73 + 5\times 5 + 5\times 20 + 3\times 42 + 4\times 32}{6} = 103.17 \tag{13}$$

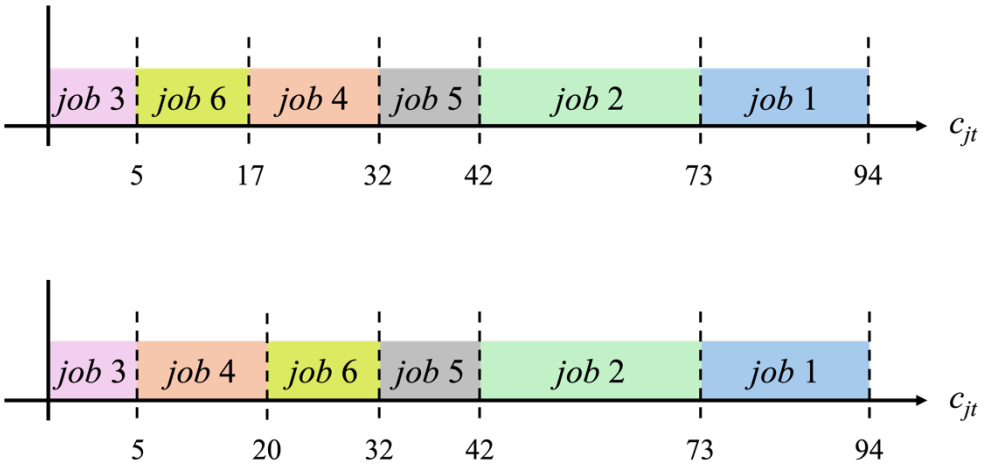

圖 5、MAWCT 於範例表 4-3 排程(上)與本研究排程(下)結果比較

三、派工法則介紹

派工法則(Dispatching Rule)是指在生產過程中，將工作分配給各個工作站或工人的法則，目的是最大化生產效率、降低成本、提高資源利用率並確保按時交付。不同的派工法則適用於不同的情境，選擇合適的派工法則對於企業來說是非常重要的。派工法則有很多不同的排序方法，本研究在此則主要介紹其中的四種方式：

- 最短加工時間法(Shortest Processing Time, SPT)是一種很常見的派工法則，是根據工作所需的處理時間進行排序，最短處理時間的工作優先處理。這方法可以有效地最小化平均完成時間和在制品數量，但可能會導致長時間處理的工作被長期延遲，影響到整體交付。

- 最早到期日法則(Earliest Due Date, EDD)則是根據工作到期日進行排序，到期日最早的工作優先處理。這種方法目標在減少延遲交付的風險，尤其適用於訂單交付時間嚴格的情況。但如果工作負荷過高，這種方法可能會犧牲短期效率。

- 關鍵比值法(Critical Ratio, CR)，這是一種更加動態的方法，計算每個工作的關鍵比率(剩餘時間與剩餘工作量的比值)，並以此排序。比值最低的工作優先處理，這種方法能夠動態調整優先級，適應性較強，但複雜度也相對較高。

- 最長加工時間法(Longest Processing Time, LPT)，則是優先處理時間最長的工作，通常用於資源平衡，確保每個工作站均勻負荷，但這可能會延遲短處理時間的工作。

其他還有像是依據剩餘作業數的多少或在製品數量等不同標準的派工法則，實際應用中只使用單一派工法通常難以滿足所有需求，因此企業常常結合多種方

法，例如：某些情況下可以優先使用 SPT 法則來提高效率，同時在緊急訂單出現時切換到 EDD 法則。工廠和服務提供者需要考慮具體情況，像是設備特性、工人技能和產品要求，靈活運用各種派工法則。總結來說，派工法則的選擇和應用會直接影響生產效率、資源利用和客戶滿意度。企業需要根據自身需求和條件，選擇適合的派工策略並不斷優化，以應對市場變化和提升競爭力。

四、派工法則數學規劃模型

4.1 參數與決策變數定義

本研究此階段將根據 SPT 和 LPT 派工法，分別建立兩種數學規劃模型，為此需新增一個決策變數 pt_t，用於排序加工時間；此外，也需新增一個參數 SP_j，以儲存前述經過排序後的加工時間。

pt_t：經排序後，第 t 個順序之加工時間，$t = 1, 2, \ldots, N$。

SP_j：第 j 項工作經排序後之加工時間，$j = 1, 2, \ldots, N$。

4.2 計算式

式(14)代表工作經派工法則排序後的加工時間。

$$SP_j = pt_j \qquad \forall j \in J \qquad (14)$$

4.3 最短加工時間派工法

延續 2.3 節建立之 MACT 數學規劃模型，加入 4.1 節之參數與變數、4.2 節之計算式，並修改部分限制式即可建構出 SPT 派工數學規劃模型。詳細的修正過程於 4.3.1 小節說明。

4.3.1 限制式

式(15)與式(16)分別自式(3)和式(5)進行修改；其中，式(15)代表每項工作之完工時間，式(16)決定每項工作的開始加工時間。式(17)與式(18)則為新增之限制式，式(17)為判斷每個階段的加工時間，式(18)則確保前項加工時間不大於後項加工時間。

$$f_j \geq s_j + \sum_{t=1}^{N} (SP_j \times x_{jt}) \qquad \forall j \in J \qquad (15)$$

$$s_{j+1} \geq s_j + \sum_{t=1}^{N} (SP_j \times x_{jt}) \qquad \forall j \in (J-1) \qquad (16)$$

$$pt_t = \sum_{j=1}^{N} (P_j \times x_{jt}) \qquad \forall t \in T \qquad (17)$$

$$pt_t \leq pt_{t+1} \qquad \forall\, t \in (T-1) \qquad (18)$$

4.3.2 求解結果

若按照 SPT 派工之 MACT 整數規劃模型，圖 6 為該模型之求解結果，共產生 90 個變數(整數變數 54 個、連續變數 0 個、二元變數 36 個)與 60 條限制式，且執行 0.01 秒即可得到最佳解為 41 分鐘。圖 7 則是該模型之加工順序，依序為 3→ 5 → 6 → 4 → 1 → 2；由此可證，此模型確實有依照加工時間短的工作優先派工之特性。

```
Explored 1 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 1: 41

Optimal solution found (tolerance 1.00e-04)
Best objective 4.100000000000e+01, best bound 4.100000000000e+01, gap 0.0000%
Number of variables:  90 (integer: 54, continuous: 0, binary: 36)
Number of constraints:  60
Objective value:  41.0
```

圖 6、SPT 派工求解結果

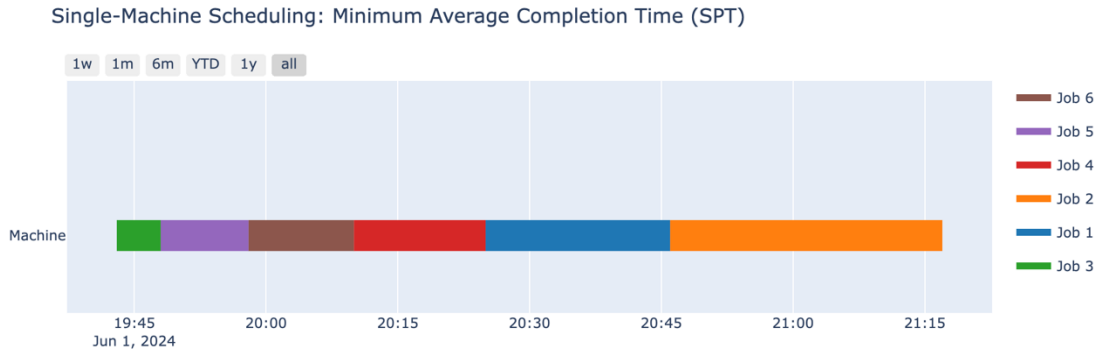

圖 7、SPT 派工排程結果

4.4 最長加工時間派工法

延續前述 4.3 節建立之 SPT 派工數學規劃模型，然而此時僅針對式(18)進行修改，其餘則依舊維持不變，即可將其改為 LPT 派工之數學規劃模型。修正後的限制式於 4.4.1 小節解釋。

4.4.1 限制式

式(19)令前項加工時間不得小於後項加工時間。

$$pt_t \geq pt_{t+1} \qquad \forall\, t \in (T-1) \qquad (19)$$

4.4.2 求解結果

以 LPT 派工之 MACT 整數規劃模型，圖 8 則為其求解結果，產生 90 個變數(整數變數 54 個、連續變數 0 個、二元變數 36 個)與 60 條限制式，並經 0.01 秒即得到最佳解 68.67 分鐘。模型之加工順序則如圖 9 所示，依序為 2 → 1 → 4 →

6 → 5 → 3；顯示此模型屬實為依照加工時間長的工作優先執行派工，與 SPT 派工之模型恰好相反。

```
Explored 1 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 1: 68.6667

Optimal solution found (tolerance 1.00e-04)
Best objective 6.866666666667e+01, best bound 6.866666666667e+01, gap 0.0000%
Number of variables:  90 (integer: 54, continuous: 0, binary: 36)
Number of constraints:  60
Objective value:  68.67
```

圖 8、LPT 派工求解結果

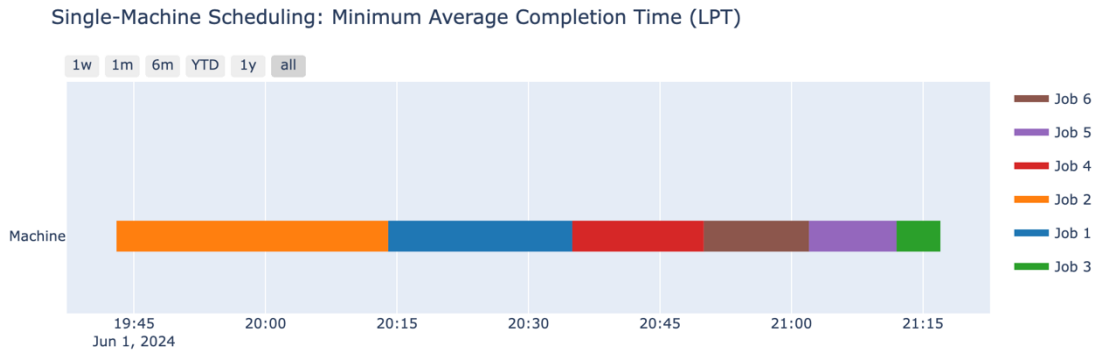

圖 9、LPT 派工排程結果

五、心得與感想

經由這次單機排程的個案研究，能夠發現訂單以不同的指派方式安排生產，會導致完工時間產生變化；若進一步放到現實的生產排程問題中，將各種訂單的限制條件、成本與時間的參數納入衡量，可能就會因此影響到企業最後的利潤跟成本。為了減少生產所花費的時間，絕大部分研究的目標式即為探討如何將排程的總完工時間或平均總完工時間最小化。

然而，在指派訂單時仍須將其他因素綜合考量，才得以符合實際情況。在此個案中，本研究目標式的設計也能模擬出這樣真實情境，像是：透過加入訂單權重，仿效需依照客戶的重要性或優先程度進行安排；以及，按照最短或最長加工時間的方式，控制產線上的在製品數量和平衡資源的使用。因此，透過上述幾種派工法的基本觀念，並結合自身數學規劃、程式與數據分析的能力，最終才得以做出最適合該情境的最佳生產決策。

六、參考文獻

林家賢，2011，產出最大化之單機排程問題之研究，國立清華大學工業工程與工程管理學系。

陳晁誠，2018，具有變動維修單機排程問題之最佳解法，國立臺北科技大學工業工程與管理系。
張旭翔，2013，以礦工基因演算法求解單機排程問題，國立臺灣科技大學工業管理系。
賴湘文，2020，運用派工法則改善半導體封裝廠生產排程，國立中央大學工業管理研究所。