

Order—Fab Allocation Plan

B01705027 陳冠綸

B04507027 林亭

B04701201 歐向容

B04701213 莊夢蝶

B05705020 張喬詠

Executive Summary

- 訂單處理原則—"時間就是金錢"

首先，我們以 order 為單位進行配置，而在處理順序上，我們認為要達成最高獲利，首先要找出哪些 order 的**成本效益最高**。為此，我們以"**時間就是金錢**"的觀念，根據產品的工廠產能(取幾何平均)和十個 order 所需的總量，計算出每樣產品相對的**時間成本**。接著以這個成本計算每個 order 的"**應有價值**"，將其和**實際的可賺得收益**比較後，優先處理效益較大的 order。

而對於為特定 order 找出**最適配的工廠**，我們優先考慮是否能在單一工廠生產完，因此以工廠"有無生產"order 要求的所有產品，作為比較優先度的最優先規則，各工廠剩餘產能(小時數)則作為比較不出來時使用的第二規則。

- 訂單處理程序&效益分析

在應有價值倍率的基礎上，我們嘗試了許多種不同 order 順序，淨利多半在兩萬多元。但對於淨利最高的組合[2 7 10 1 4 8 5 3 6 9]，跳過不可行或虧損的 order，此順序總共能接下[2 7 1 4]這些 order，其中[7 8 10]都是由單一 fab 完成，**總淨利**可到達**\$31,300**。

觀察 order 個別收益會發現，這樣的**高效益**多歸功於單一廠房完成 order，它們帶來的收益較有 split(拆訂單生產)的 order 高出不少。

A. 訂單配置結果：

Orders	Allocation	Completion	Splitting
1	Fab 6, 7	yes	1 time
2	Fab 2, 7	yes	1 time
3	Fab 1, 5	yes	1 time
4	Fab 1, 2, 3, 8	yes	3 times
5		no	
6		no	
7	Fab 5	yes	0 time
8		no	
9		no	
10	Fab 2, 3, 4, 6	yes	3 times

→ 工廠產能使用結果：

Fab	Allocation	Utilization	Remaining hours
1	Item 3, 7, 8, 9	Hours : 681	Hours : 159
2	Item 1, 10, 11	Hours : 363	Hours : 387
3	Item 4, 8, 12	Hours : 610	Hours : 0
4	Item 12	Hours : 267	Hours : 203
5	Item 2, 3, 6, 12	Hours : 550	Hours : 10
6	Item 5, 8, 10	Hours : 240	Hours : 0
7	Item 2, 4, 5, 6, 11	Hours : 1250	Hours : 0
8	Item 2, 5, 9	Hours : 930	Hours : 0

- 結論

根據**成本效益最大化**的原則，我們經由**數學模型**運算及**程式測試**，找出了最佳訂單配置方式(如上左圖)。此方法證實能在各工廠既有產能限制下，有效篩選出合適訂單(營收>成本)，並處理完這六筆訂單的生產，帶來**\$31,300 總淨利**，同時使**產能利用率**達到 **87%**。

Method Description

I. 訂單(Order)處理順序—數學模型求最佳解

我們以訂單(order)為單位處理，透過嘗試五種不同的篩選方式後(見附錄一)，我們發現找尋**訂單處理順序**的最佳方法如下：

i	(Price i - splitting cost) / Xi	Priority
1	467.74	2
2	637.57	1
3	355.60	6
4	416.26	3
5	321.56	7
6	231.41	10
7	394.68	4
8	320.16	8
9	307.47	9
10	390.23	5

首先，令 order i 所需之 item j 數量為 X_{ij} ，取可生產 item j 之所有 $Fab\ k$ 。 $i \in \{1, 2, \dots, 10\}, j \in \{1, 2, \dots, 12\}, k \in \{1, 2, \dots, 8\}$ 。令 $Fab\ k$ 生產 item j 的每小時產量為 Y_{jk} ，令所有大於 0 之 Y_{jk} 進行幾何平均，得出 Y_j 。算出 Y_1, Y_2, \dots ，以此類推。再令 $Z_j = \sum_{i=1}^{10} X_{ij}$ ， Z_j 代表所有 order 所需的 item j 數量總和。 Z_j / Y_j 為我們假定須完成生產此 item j 所期望花費之時間比較值。再令 $X_i = \sum_{j=1}^{12} X_{ij} / (Z_j / Y_j)$ ，此值為假定 order i 要完成此單所有 item 需求必須花費的時間比較值。假定 split 次數均為一次，最後再算出 $(Price\ i - splitting\ cost) / X_i$ ，值愈大者，代表獲利的效率愈高，則優先順序愈高。(結果如左表)

II. 訂單-工廠(Fab)配置程序—程式自動運算(大略邏輯敘述如下)

1. 找出和 order 適配的所有 Fab (依據有無生產該項產品)

我們先將 fab 的十二項產品資料簡化為**有/無生產**該項產品，化為共十二個 0 / 1 的值，之後四個一組、分配為三組數據，並求出其二進位下的值(0~15)。接著以**十六進位**計算各組須乘以十六的幾次方，最終將這十二個 0 / 1 值整合成為一個整數，另外儲存。order 的產品需求同樣以上述方法儲存。

此時在電腦儲存空間中，這些整數各自以二進位儲存，且每一個 bit 的二進位數都表示有/沒有該項產品。接著，將 fab 的生產狀況以二進位碼比對到 order 的產品狀況，計算其值相同的 bit 數，以此做為第一階段的適配程度。適配程度越高，排序就越前面。

2. 找出**最適配**的 Fab (依據 Fab 對該產品的產能)並下單

求出第一階段適配度後，經常有適配度相同的情況，此時我們從中選擇**總產能(capacity)**最大的 fab。這個階段基本上幾乎不會出現同等適配度的情形。

選出 fab 後，即下單給該 fab。如果可以將訂單全部做完，就紀錄下各個 item 上的所需時間；如果這個 fab 不能全部做完，就以優先完整做完其中一項產品、再做下一項的方式，直到將所有剩餘時間(產能)用完為止。

- 依此邏輯持續將 Order 的剩餘產品分配到剩下的 Fab 生產

A. 若 Order 全處理完，則計算總 splitting cost

- 若 $\text{cost} > \text{Price}$ → 棄單 → 產能還 Fab
- 若 $\text{cost} < \text{Price}$ → Order complete

B. 若 Order 無法由 Fab 處理完 → 棄單 → 產能還 Fab

3. 算出總 Profit (程式執行結果如下圖(部分))

```
-----Order[2] is completed
Order[2] to Fab[1]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[2]: 1000 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[3]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[4]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[5]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[6]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[2] to Fab[7]: 0 3000 0 0 0 1000 0 0 0 0 3000 0
Order[2] to Fab[8]: 0 0 0 0 0 0 0 0 0 0 0 0
ifComplete: 1 Split: 2
order[2] value: 6300

-----Order[1] is completed
Order[1] to Fab[1]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[1] to Fab[2]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[1] to Fab[3]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[1] to Fab[4]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[1] to Fab[5]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[1] to Fab[6]: 0 0 0 0 2250 0 0 0 0 6000 0 0
Order[1] to Fab[7]: 0 0 0 2000 750 0 0 0 0 0 0 0
Order[1] to Fab[8]: 0 0 0 0 0 0 0 0 0 0 0 0
ifComplete: 1 Split: 2
order[1] value: 6500

-----Order[4] is completed
Order[4] to Fab[1]: 0 0 0 0 0 0 0 0 60 0 0 0
Order[4] to Fab[2]: 0 0 0 0 0 0 0 0 0 0 2000 0
Order[4] to Fab[3]: 0 0 0 0 0 0 0 0 0 0 0 3000
Order[4] to Fab[4]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[4] to Fab[5]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[4] to Fab[6]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[4] to Fab[7]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[4] to Fab[8]: 0 4000 0 0 5000 0 0 0 7940 0 0 0
ifComplete: 1 Split: 4
order[4] value: 1500

-----Order[7] is completed
Order[7] to Fab[1]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[2]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[3]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[4]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[5]: 0 1000 500 0 0 5000 0 0 0 0 0 500
Order[7] to Fab[6]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[7]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[7] to Fab[8]: 0 0 0 0 0 0 0 0 0 0 0 0
ifComplete: 1 Split: 1
order[7] value: 7700

-----Order[10] is completed
Order[10] to Fab[1]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[10] to Fab[2]: 0 0 0 0 0 0 0 0 0 0 1560 0
Order[10] to Fab[3]: 0 0 0 2000 0 0 0 1100 0 0 0 0
Order[10] to Fab[4]: 0 0 0 0 0 0 0 0 0 0 0 4000
Order[10] to Fab[5]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[10] to Fab[6]: 0 0 0 0 0 0 0 900 0 1440 0 0
Order[10] to Fab[7]: 0 0 0 0 0 0 0 0 0 0 0 0
Order[10] to Fab[8]: 0 0 0 0 0 0 0 0 0 0 0 0
ifComplete: 1 Split: 4
Order[10] release Fab[1]: due to negative
Order[10] release Fab[2]: due to negative
Order[10] release Fab[3]: due to negative
Order[10] release Fab[4]: due to negative
Order[10] release Fab[5]: due to negative
Order[10] release Fab[6]: due to negative
Order[10] release Fab[7]: due to negative
Order[10] release Fab[8]: due to negative
order[10] value: -3800
```

III. 結論

總淨利：31,300；產能利用率：87%

(詳細配置方式請見 Excel 附檔)

附錄

- **補充說明：Order 順序選擇方式**

我們實際上共嘗試了以下**五種**選擇 order 的方式：

(1) **增加收入**：價格高的優先。以 order price 的高低順序，以 price 最高的優先處理。

→ 此方法實行簡單，效果也不錯，但只能接下三筆 order。

(2) **降低成本**：以 splitting cost 最高的優先，以減少 splitting cost 高的 order 反而必須被 split 多次的情況。這個做法假設多數的 order 都要 split，且目標是處理完所有的 order，因此忽略了 order 本身的價格。

→ 事實證明，此方法效果相當不佳，收益只有用 price 排序的一半左右。

(我們另外也嘗試過 splitting cost 低者優先，得到的收益比高者優先更低)

(3) **同時考量 price 與 splitting cost**：以一比一的影響比例，同時考量上述兩者。由於上述兩者都是高的優先，因此我們嘗試將各個 order 的 price 除以平均的 price，再加上 splitting cost 除以平均的 splitting cost，作為優先級順序。

→ 實測的結果和單純以 price 排序相去不遠、甚至更差，因此最後也不採用。

(4) **考量時間成本與收入的比例**：首先對十二項產品，分別計算其收益率，計算方法如下：
十個 order 總需求量 / 8 個 fab 全生產此產品下，單小時最高產量 = 全生產完畢所需時間

如果一個產品的此項數值低，代表它生產相對不花時間，所以我們想要先生產這樣的產品。實際的情況中，就是優先處理大部分是這類產品的 order。我們這樣計算優先度：

加總十二個產品的 (order 要求的產品數*此產品全生產完畢所需時間)和上面的數值不同，這次算出的數值越小越好。

→ 然而實際上，這個方法的表現並不好，甚至不如單純考量 price。我們討論後認為這是因為產品時間成本計算方法不對，修正並加入更多想法後有了後來的第五種方法。

(5) **以幾何平均計算的時間成本、各單價值 & split 後價值差距**，即我們最後採用的方法。主要發想是藉由產品時間成本，求出各單"該有的價值"，並比較其實際的 price 與我們計算出的價值間的倍率。詳細說明參見" Method Description"。