

半導體製造業之供應鏈網路管理與資源分配

陳靜枝 國立台灣大學資訊管理學系
計畫編號：NSC-90-2218-E-002-045

一、摘要

由於全球經濟體系的成形，企業不僅必須加強本身的生產技術與成本控制，更必須在整個供應鏈中創造更高的價值。對於臺灣半導體代工的產業而言，供應鏈管理具價值之處，即為能將所有牽涉產品生產、配銷過程的個體及其相互鏈結與影響均列入考慮，當晶圓生產廠商接獲訂單，其供應鏈管理者會根據產能、需求量、交貨日與成本等考慮因素，決定此訂單應由何晶圓廠區生產、由何測試分類廠做分類、由何封裝廠做封裝與最後由何檢驗廠做檢驗。本研究的主要目的為在各種限制條件下，以最小成本為目標，找出最佳廠商組合。本研究將引用圖學理論與網路流量模式，找出並建構解決問題的演算法，最後並將此演算法與整個半導體產業決策品質促成者結合。

二、計畫說明

一個具有競爭優勢的企業必須能夠以最短的時間、最大的彈性與最低成本來生產多樣化與高品質的產品。以臺灣的半導體產業為例，歷經 60 年代萌芽、70 年代技術改進、80 年代的自立與擴張、到 90 年代的成熟與完備，已經成為台灣最重要的產業之一。在半導體製造產業的供應鏈中，可將所有個體區分為四大類[21]：晶圓設計 (design)、晶圓生產 (fabrication)、封裝(assembly) 與分類測試檢驗 (sorting and testing)。各類的個體擁有不同的設備與技術，各司其職、互相支援，其中晶圓生產所佔的比重最大也最具協調能力，因此可將整個半導體製造產業供應鏈視為以晶圓生產廠為中心廠、而其他的個體為協力廠商的架構。因此，對不同的訂單選擇適當的廠商組合的問題就非常重

要。本研究的主要目的，就是在有限的產能限制下，以最小的生產與運送成本或是最短的生產週期，滿足所有訂單與未來的需求預測，找出最佳的廠商組合。

Uzsoy [8]指出，制定半導體生產策略的主要目的就是：降低生產成本、增加生產率、改善產品品質以及滿足交貨期。Uzsoy 等[8, 9]將解決半導體生產控制與排程等問題之相關文獻，總分成三大類：(1)績效評估；(2)生產計畫；(3)現場管制。其中與生產計畫相關的幾個重要的文獻，如 Golvin [5]指出生產計畫很難選擇正確目標函數，因此參考 Bitran 的建議，採用「階層步驟」(hierarchical approach)，用累積性的資訊做長期的決策，如以長期資料估計某一個產品線需要多少的產能，再往下做更仔細的作業排程，其限制條件則為已確定的長期的

決策。這樣的作法非常類似目前供應鏈管理的概念。此外，Leachman[7]則將依晶圓四個生產步驟視為一虛擬公司的四個流程，且以存貨作為四者的連結。將整個虛擬公司的生產計畫模式化為一線性規劃問題（Linear Programming Problem），這個虛擬公司的想法就是供應鏈網路的整合，解決問題的方法也類似目前應用於供應鏈管理問題的解決技術。Hadavi and Voigt [6]則使用不同階層（Level）的抽象觀念來作計畫，在不同階層用不同的模式來制訂決策。這個作法是 Siemens 所發展出來的，同時也應用於 Siemens 的一家晶圓廠中。這個方法介於傳統的 MRP 與目前發展的供應鏈管理中間。

從上述與半導體的生產計畫（Production planning）相關的文獻中，可以看出來其實半導體的生產計畫研究並未跳脫供應鏈管理的研究範疇，因此可以引用供應鏈管理的研究成果。供應鏈管理的研究這幾年在國內外均引起很大的迴響，Geoffrion & Powers [4]在他們 1995 年的調查報告中指出，20 年來供應鏈的演進使得運籌管理成為公司中一項重要的功能，主要的原因來自電腦與通訊科技的進步、演算法的成

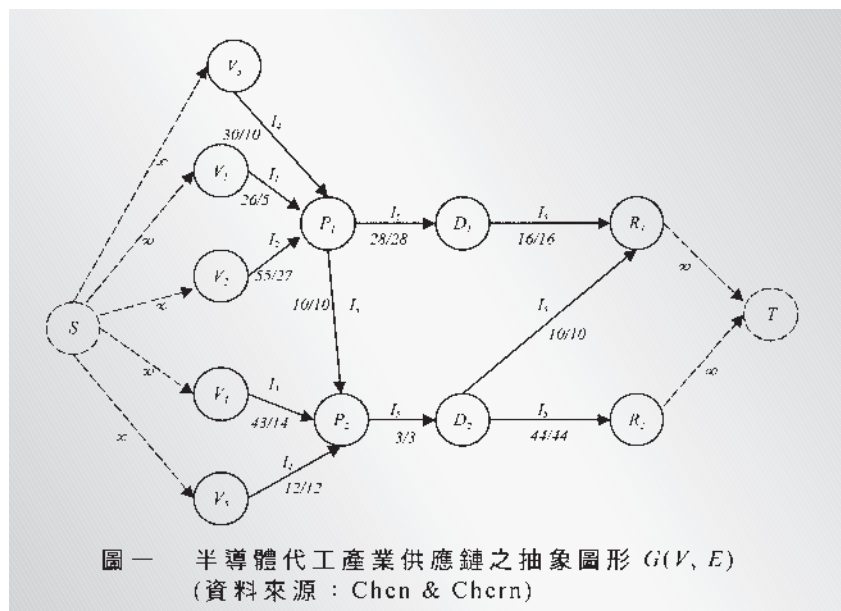
熟、資料與知識庫管理工具的改良、與管理科學各種運算模式的引進並軟體化。解決供應鏈中各種問題所使用的方法是管理科學的研究主題之一，Vidal & Goetschalckx [10]在他們 1997 年的調查報告中，整理了大多數與供應鏈相關的問題模式與解決方法，大多數即藉用混合線性與整數規劃的模式，由於問題的規模太大無法有效率的找到最佳解，許多作者便提出啟發性搜尋法則，將目標放在接近最佳解的近似解上。同樣的結論也可以在 Erengüç, Simpson, & Vakharia [3] 研究中發現。Arntzen, Brown, Harrison, & Trafton [1] 在 1995 年的研究中就以 DEC 為例，建構了一解決全球供應鏈問題的決策支援系統，採用混合線性與整數規劃的模式為主要的解決引擎。然而其決策支援系統雛型只包

括計算部份，並不算完整的系統。對於網路管理模式運用於解決供應鏈管理問題上，Chen & Chern [2]在其 1999 年的研究中提出一演算法，將單一成品但多原料與多層級的供應鏈網路轉換成可以運用最短路徑法解決的網路模式，並將此演算法運用於解決一般供應鏈網路廠商指派或組合的問題上，其提出的演算法不但可找出最佳解而且為 polynomial。

本研究基於可行解與計算速度的考量，引用 Chen & Chern [5]的研究，提出另一種解決問題模式，將供應鏈的廠商指派與資源分配的問題以網路流量管理的啟發性演算法解決。

三、模式建構

本計畫以半導體產業的供應鏈為研究之主要目標，





先分析半導體代工產業供應鏈相關特性，及不同半導體代工產業供應鏈管理策略，再將這些問題建立成數量模式—亦即線性規劃模式。

半導體代工產業供應鏈會形成一個網路，每個階段都有不同的廠商可提供加工服務，因此可以用抽象圖形 (graph) $G(V, E)$ 來表示 (如圖一) 來表達。成本的定義與計算時要考慮產品結構 (如圖二) 的問題，亦即將產品結構樹運用於半導體代工產業供應鏈網路上，使成本的加總反應出真實成本，這也是本計畫研究的重點之一。又由於半導體產業每期的產品組合相當複雜，因此與產品結構結合成一網路結構圖 (如圖三)。其中，供應鏈、產品結構樹與產品組合之抽象圖形 $G(V, E)$ 與其中相關的參數與決策變數的定義如下。

1. 參數或變數的下標

i, j, g : 用來表示節點。

r : 用來表示訂單。

l : 用來表示時距。

k : 用來表示完成產品。

k' : 用來表示組成物料。

2. 參數部份

V : 代表節點所形成的集合。

E : 代表邊所形成的集合。

LT_{ijk} : 任何二節點 (i, j) 間傳送產品 k 的前置時間。

D_{rkl} : 產品 k 的訂單 r 在時距 l 之需求量。

v_{ikl} : 節點 i 在時距 l 生產產品 k 的產能限制。

u_{ijkl} : 二節點 (i, j) 間在時距 l 傳送產品 k 的產能限制。

c_{ijk} : 在二節點 (i, j) 間運送產品 k 之單位加成成本。

c'_{ik} : 在節點 i 製造產品 k 之單位加成成本。

p_{ijk} : 在二節點 (i, j) 間運送產品 k 之單位加成時間。

p'_{ik} : 在節點 i 製造產品 k 之單位加成時間。

h_{ik} : 在節點 i 儲存產品 k 之單位存貨成本。

w_r : 訂單 r 之重要權數的比重。

DD_r : 訂單 r 之到期日，以時距計。

PT : 總計畫時距， $PT \geq Max_r DD_r + \pi$ 。 π 是可延展的時距數。

3. 決策變數部份

x_{ijkrl} : 在時距 l 於二節點 (i, j) 間運送產品 k 的訂單 r 之數量。

y_{ikrl} : 在時距 l 於節點 i 製造產品 k 的訂單 r 之數量。

F_r : 訂單 r 之生產週期，以時距計。

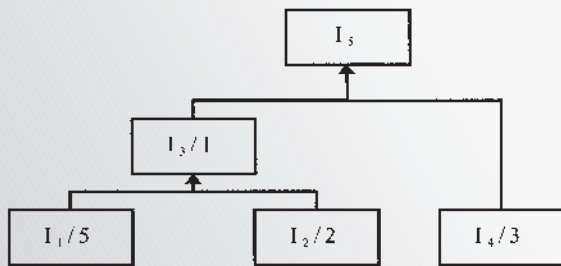
TD_r : 訂單 r 之延誤時間，以時距計。

ST_{kr} : 訂單 r 之產品 k 的開始時距。

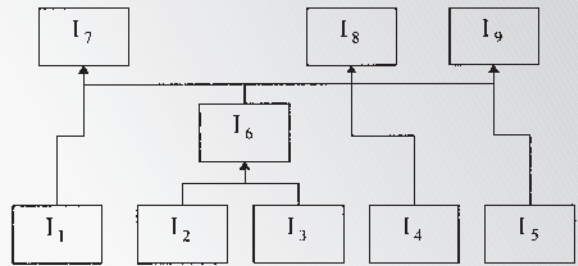
CT_{kr} : 訂單 r 之產品 k 的完成時距。

CT_{kr} : 訂單 r 之產品 k 的完成時距。

LT_{ijkrl} : 任何二節點 (i, j) 間傳送訂單 r 之產品 k 的前置時間，以時距計。



圖二 半導體產業供應鏈產品結構樹之抽象圖形(資料來源: Chen & Chern)



圖三 半導體產業供應鏈產品結構樹暨產品組合之抽象圖形(資料來源: 本計畫)

4. 線性規劃模式部份

最小化製造與運送成本，但不考慮存貨成本。在這個模式中，所有的訂單都必須在到期日內完成，每個廠或供應商都有其製造與運送成本，且會有產能限制，每個最終產品都有其已知的產品結構。

• 目標函式：

$$\text{Min } \sum_i \sum_j \sum_k c_{ijk} \sum_r \sum_l x_{ijkl} + \sum_i \sum_k c'_{ik} \sum_r \sum_l y_{ikrl}$$

• 限制式：

$$(1) \sum_i \sum_r x_{ijkl} + \sum_i \sum_r y_{ikrl} = \sum_i \sum_r x_{jikl} + \sum_i \sum_r y_{ikr(l+1)}$$

for all k, l , and j where k 's are the final finished items and where i and j are not starting and ending points or $i, j \neq s, t$ 。

$$(2) \sum_r x_{ijkl} \leq u_{ijkl} \text{ for all } i, j, l, \text{ and } k。$$

$$(3) \sum_r y_{ikrl} \leq v_{ikrl} \text{ for all } i, j, l, \text{ and } k。$$

$$(4) \sum_j \sum_l x_{sjkrl} \geq \sum_l D_{rkl} \text{ for all } r \text{ and } k \text{ where } l \text{ is from } 0 \text{ to } DD_r \text{ for all orders。}$$

$$(5) \sum_i \sum_l x_{itkrl} \geq \sum_l D_{rkl} \text{ for all } r \text{ and } k \text{ where } l \text{ is from } 0 \text{ to } DD_r \text{ for all orders。}$$

$$(6) \sum_l y_{skrl} \geq \sum_l D_{rkl} \text{ for all } r \text{ and } k \text{ where } l \text{ is from } 0 \text{ to } DD_r \text{ for all orders。}$$

$$(7) \sum_l y_{ikrl} \geq \sum_l D_{rkl} \text{ for all } r \text{ and } k \text{ where } l \text{ is from } 0 \text{ to } DD_r \text{ for all orders。}$$

$$(8) \sum_r \sum_g x_{gikr(l-Lgik')} + \sum_r y_{ik'r'l} = \sum_r \sum_k \sum_j x_{jikr(l+Lgik')} + \sum_r \sum_j y_{ikr(l+Lgik')} \text{ for all } k, l, \text{ and } j \text{ where } k' \text{'s are the component items of finished item } k (k \neq k')。$$

$$(9) x_{ijkl} \geq 0 \text{ and } y_{ikrl} \geq 0 \text{ for all } i, j, l, r, \text{ and } k。$$

目標函式的部份只考慮原料、組裝及運送成本最小的途徑。其中限制式(1)代表網路模式中流量均衡的要求 (in flows = out flows)。限制式(2)代表二節點間連結的流量限制要求，限制式(3)代表節點本身產能的限制要求。限制式(4)、(5)、(6)及(7)則代表訂單需求量一定要在每筆訂單之到期日內滿足的限制要求。限制式(8)代表物料與完成品間的流量均衡的要求。最後限制式(9)是所有決策變數均為非負變數的要求。

透過線性規劃模型解決這個問題，會有以下幾個缺點：

1. 線性規劃模型複雜解題時間長。本研究要解決的問題，會與供應鏈網路節點、連結數、產品結構節點、層數與訂單數相關。

當同時多個維度增加元件，相關的變數或限制式個數就會呈倍數擴充，致使模型的複雜度快速增加。

2. 線性規劃模型無解時，無法知道哪些訂單或限制式造成無解。
3. 目標唯一。依前所述的線性規劃模型，可能找出一組生產成本最小的生產組合與搭配，但此廠商組合卻可能不是以最接近訂單需求時距的時距生產。

因此本研究將提出一套啟發性演算法以解決此複雜的問題。

四、啟發性演算法

由於供應鏈網路管理與資源分配問題內所牽涉到的變數太多且限制式太複雜，無法利用一般的線性規劃軟體解決，因此必須發展啟發性 (heuristic) 演算法。本計畫引用 Chen & Chern [5] 在其 1999 年所提出之供應鏈網路最短路徑演算法，並加入訂單需求細項，發展一供應鏈網路管理與資源分配最短路徑演算法。在解釋本研究演算法過程時，需計算某些特別數值。

(一) 定義名詞



1. AC_{it} : 節點 i 在規劃時距 PT 內, 其所剩餘的最大產能。
2. EB_i : 在訂單交貨時距 DD_i 限制下, 節點 i 的產能在 EB_i 受到上下游節點產能限制以及等待時距限制仍為有效, 稱為節點 i 的有效生產時距。
3. FEB_i : 節點 i 的最大之 EB_i , 稱為節點 i 最終有效生產時距。
4. $FEC P$: 完整生產樹 S 中每個節點都利用最終有效時距的剩餘產能生產, 所能生產最大產量。即是取每個節點在其最終有效生產時距的剩餘產能之最小值。
5. $MSLB_i$: 節點循 i 所有可能路徑到 END 節點所需要最短前置時距。

(二) 演算法包含步驟

1. 產能限制分離與產能初始化: 此步驟主要的工作是解決供應鏈網路中迴圈的問題。
2. 產能轉換: 此步驟主要的工作是統一供應鏈網路中單位標準, 全部以最終產品為單位標準。
3. 訂單排序: 訂單的重要性可因使用者的定義或供應鏈的特性不同, 而有不同的訂定方式。本研究也

將提供不同的排序演算法, 以配合之前數量模式中不同的目標。

4. 訂單的資源分配: 每筆訂單依序規劃排程。

其中第 4 步驟是本演算法中最複雜的部份, 以下詳細解說其訂單規劃排程 (主要如圖四):

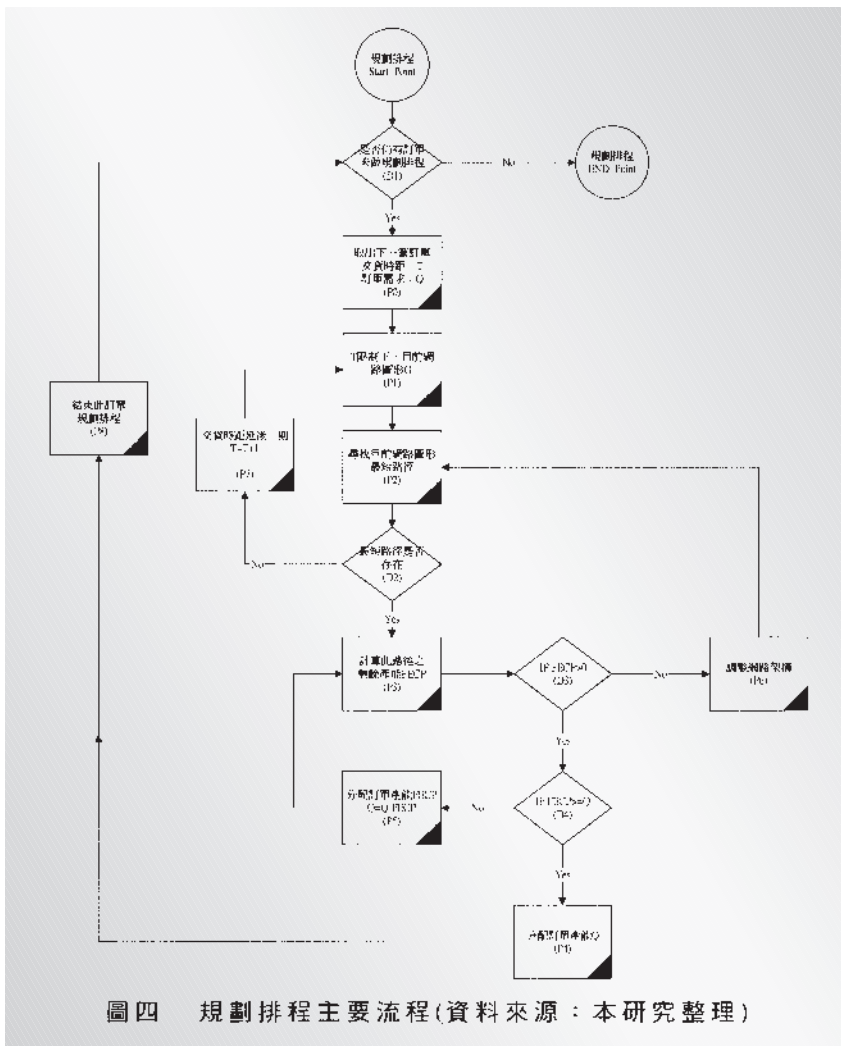
(P1) 是否仍有訂單未做規劃排程, 若『是』則到程序 P1, 若『否』則結束規劃排程。

(P2) 尋找最佳生產樹 S : 在

產品組成結構與供應鏈網路確定後, 採用 Chen & Chern [5] 的演算法尋找最佳的生產樹 S 。若 S 存在則到 (P3), 若否則到 (P7)。

(P3) 計算生產樹 S 之 $FEC P$ 。若生產樹 S 之 $FEC P=0$, 到 (P6); 若 $FEC P < Q$, 到 (P5); 否則則到 (P4)。

(P4) 安排此生產樹 S 生產 Q , 此訂單需求已被滿足, 結束此訂單規劃排程, 到 (P1)。



圖四 規劃排程主要流程(資料來源: 本研究整理)

(P5) 安排此路路徑生產 FECP，訂單需求減為 $Q = (Q - FECP)$ ，必須重新計算此路徑的 FECP，到(P3)。

(P6) FECP=0 表示此路徑已無任何有效剩餘產能可以生產最終產品，而訂單需求仍未滿足，必須調整網路結構，以幫助尋找新的生產樹，P6 完成後回到(P2)。

(P7) 若目前網路結構中，無任何一條路徑存在，則表示此供應鏈網路已經無法從起點走到終點，而訂單需求還未完全滿足，表示在交貨時距前此供應鏈已無法多生產任何一個最終產品，因此需將交貨時距往後順延一時距，再還原網路架構，需重新尋找生產樹，回到(P1)。

在程序(P3)中，計算生產樹 S 之 FECP 的詳細步驟可以分成以下兩個子演算法：

子演算法(一)：

(1) Set Node N_j to be END node. So the FEB_i of node $N_j = T$ 。

(2) For ALL $i N_i \rightarrow N_j$ (Lead time = L, FEB_i of node $N_j = K$)。

• FEB_i of node $N_i = K - L$

• For AC_{it} of N_i at $FEB_i = 0$,
 $FEB_i = FEB_i - 1$

• $N_j = N_i$

子演算法(二)：

(1) FOR all Node N_i to N_n find FEB_i 。

(2) IF FEB_i of All Nodes on the tree Exist, compute FECP of tree S where $FECP = \text{Minimum } AC_{it}$ of all node of tree S at FEB_i 。

在程序(P6)中，必須調整網路結構，以幫助尋找新的生產樹，詳細步驟可以分成以下兩個子演算法：

子演算法(三)：

(1) Minimum System Lead-Bucket ($MSLB_i$) = ∞ for all nodes。

(2) $MSLB_i$ of nodes linking to END point = 0。

(3) FOR ALL Node i 。

• FOR ALL Node j

• IF $MSLB_i < MSLB_j + \text{Lead time bucket of (Node } i \rightarrow j)$, then $MSLB_i = MSLB_j + \text{Lead bucket of (Node } i \rightarrow j)$ 。

子演算法(四)：

• Edge Removal:若生產樹 S 中根據從 $MSLB_i$ 到 DD_r ，任一個 edge ($i \rightarrow j$) 的產

能如果無效，即表示無法再利用此兩點間路徑生產則可以去邊。

• Node Removal:若生產樹 S 中根據從 $MSLB_i$ 到 DD_r 的產能，任一個節點 i 如果已無產能，即表示無法再利用此節點生產則可以去點。

• Copy and Cut Path: 若生產樹 S 無法去邊也無法去點，即表示生產樹 S，雖然每一個 Edge 都仍有產能相連，但此生產樹 S 的有效剩餘產能為 0，在交貨時距限制下，以無法利用此生產樹 S 生產。將生產樹 S 拆解成數個子生產樹 S_1, S_2, \dots, S_M ，且 $S_m \subset S_n$ for all $m \neq n$, $S_2 \subset S_1$, and $S_1 \cup S_2 \cup \dots \cup S_M = S$ 。演算法如下：

(1) Let Node N_j be the final node and γ be an empty set。

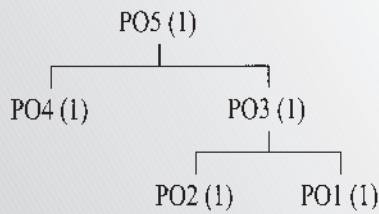
(2) While $N_j \neq \text{Starting Point}$

• Find a sub tree S_m of S, which could product the final product from the most upper Node N_i to N_j node at the earliest time bucket。

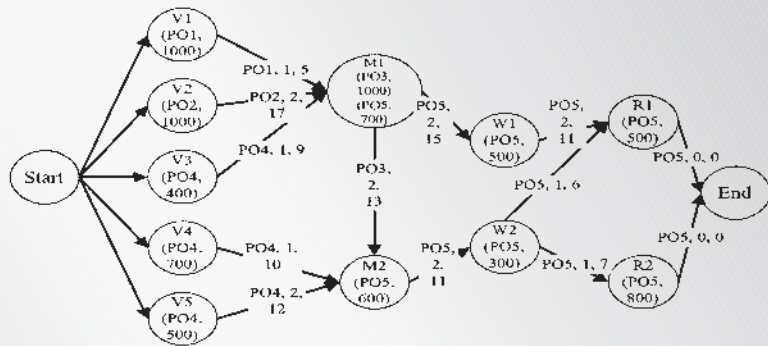
• IF $S_m \notin \gamma$, then include S_m in set γ and duplicate N_i 。

• Let $N_j =$ the upper stream node of N_j 。

(3) DELETE tree S。



圖五 PO5 產品結構圖(資料來源：本研究整理)



圖六 供應鍊網路圖形(資料來源：本研究整理)

本演算法預先為所有的訂單重要性排序，重要性高的訂單將優先滿足，且在尋找最小成本最佳解的同時，安排最接近交貨時距的生產時距分配產能。當需求若無法滿足時，以最少的延遲時距安排產能。效率方面，本演算法不會隨著變數的增加，而大幅影響解決問題的效率並提供高度的彈性以面對關係式的改變。且在產能限制與交貨時距限制下，當需求若皆能滿足時，本演算法之結果相同於線性規劃模型之最佳解。

五、模式實例分析

本章以一個實例，在 PIV-1.7G CPU 與 1G RAM 的環境下比較啟發性演算法與線性規劃模型的排程結果。其中線性規劃模型以 ILOG 計算，啟發性演算法以 FoxPro 6.0 建構。實例一之產品結構如圖五，供應鍊架構如圖六

，處理兩張訂單如表一，啟發性演算法的演算流程如下：

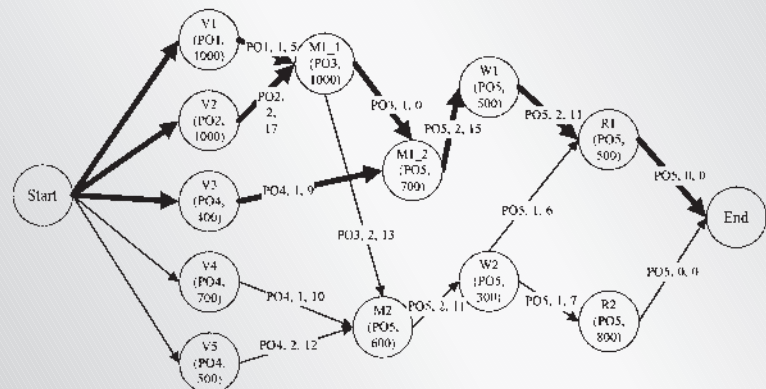
1. 取出訂單 1，需求 400，交貨時距 8。而供應鍊網路尋找出生產樹 S1 如圖七，單位成本 57，可用產能 FECP 為 400；可分配產能 400，完成訂單 1 排程。
2. 取出訂單 2，需求 600，交貨時距 8。還原供應鍊網路圖形，最佳生產樹尋

找相同生產樹 S1，此時 FECP 為 100， $FECP < 600$ 。訂單 2 安排此路徑生產 100、交貨時距 8，且發現此時原最短路徑 $FECP = 0$ ，無法在利用此路徑安排任何產能。而訂單需求尚未完全滿足，因此需調整供應鍊圖形。

3. 調整後圖形如圖八，並尋找最短路徑對應回原圖形。此路徑成本 62，但

表一 訂單需求(資料來源：本研究整理)

訂單編號	需求	交貨時距	訂單排序次序
1	400	8	1
2	600	8	2



圖七 最短路徑一(資料來源：本研究整理)

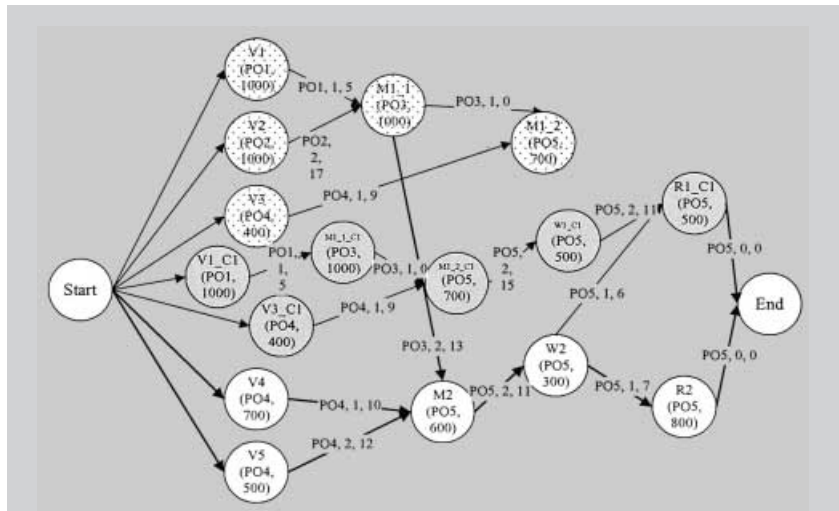
目前 FECP 為 0，無法安排任何產能，因此必須再調整一次供應鏈網路圖形。

- 調整後圖形如圖九，並尋找最短路徑對應回原圖形。此路徑成本 63，目前 FECP 為 300。FECP ≤ 500 = (600 - 100)。為訂單 2 安排此路徑生產 300、交貨時距 8，此訂單需求尚未完全滿足。此時新最短路徑 FECP = 0，無法再利用其安排產能。需再一次調整網路圖形，且發現調整供應鏈圖形已經無法從起點生產到終點。

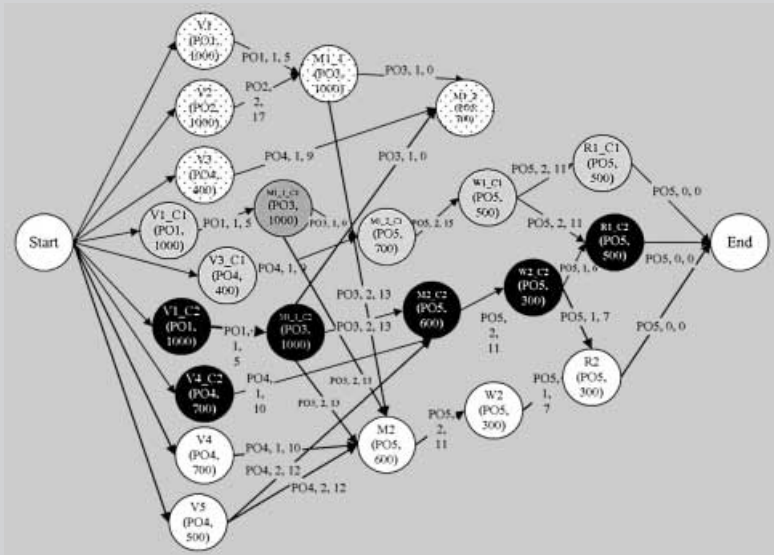
- 訂單需延遲一期，供應鏈圖形還原成圖六。再一次尋找最短路徑如圖七此路徑成本 57，目前 FECP 為 400。FECP ≥ 200。訂單 2 安排此路徑生產 200、交貨時距 9，訂單 2 需求已經完全滿足。

- 所有訂單需求皆已滿足，結束訂單規劃排程。最後結果如表二。

與線性規劃的結果比較發現線性規劃模型共使用 88 個變數，以及 78 個限制式。多增加一張單子增加 44 個變數、32 個限制式。利用 ILOG 軟體結果無最佳解，表訂單需求無法在交貨時距要求下全部滿足。



圖八 調整後供應鏈圖形一(資料來源：本研究整理)



圖九 調整後供應鏈圖形二(資料來源：本研究整理)

表二 訂單分配結果(資料來源：本研究整理)

訂單編號	分配產能	交貨時距	生產成本	是否延遲	交易編號
1	400	8	57	否	1
2	100	8	57	否	2
2	300	8	63	否	3
2	200	9	57	是	4

若將第二張需求改為 400，調整後共使用 88 個變數，以及 78 個限制式。利用 ILOG 結果如下：

$$\begin{aligned}
 x_{W1,R1,po5,1,6} &= x_{M1,2,W1,po5,1,4} = \\
 x_{M1,1,M1,2,po3,1,3} &= x_{V3,M1,2,po4,1,3} \\
 &= x_{V2,M1,1,po2,1,1} = x_{V1,M1,1,po1,1,2} \\
 &= 400, y_{M1,2,po5,1,4} = y_{M1,1,po3,1,3}
 \end{aligned}$$



$$\begin{aligned}
&=y_{V3,po4,1,3}=y_{V2,po2,1,1}=y_{V1,po1,1,2} \\
&=400, x_{V2,M1_1,po2,2,1}=x_{V1,M1_1,po1,2,2} \\
&=400, x_{W1,R1,po5,2,6}=x_{M1_2,W1,po5,2,4} \\
&=x_{M1_1,M1_2,po3,2,3}=x_{V3,M1_2,po4,2,2} \\
&=100, x_{W2,R2,po5,2,7}=x_{M2,W2,po5,2,5} \\
&=x_{M1_1,M2,po3,2,3}=x_{V4,M2,po4,2,4} \\
&=300, y_{M1_1,po3,2,3}=y_{V2,po2,2,1}= \\
&y_{V1,po1,2,2}=400, y_{V4,po4,1,4}=300, \\
&y_{V3,po4,1,2}=y_{M1_2,po5,1,4}=300。
\end{aligned}$$

總成本為 47400。與之前本演算法，在不考慮延遲訂單，則成本相同。而在解題時間上，啟發性演算法需要少於 10.97 秒，線性規劃模式則需要 264 秒，啟發性演算法明顯地較快速。

六、結論

本研究針對半導體代工產業供應鏈相關特性，以及不同半導體代工產業供應鏈

管理策略，綜合分析其效能與評估方案，再將這些問題建立成數量模式——亦即線性規劃模式，但是本問題若使用線性規劃模型解決會有以下缺點：

1. 變數與限制式相當多且複雜。
2. 目標唯一。
3. 不能保證一定有解。

因此本研究提出一套演算法，以解決此複雜的問題，演算法首先為所有的訂單重要性排序，重要性高的訂單將優先滿足，且在尋找最小成本最佳解的同時，安排最接近交貨時距的生產時距分配產能。當需求若無法滿足時，以最少的延遲時距安排產能。

作者簡介



陳靜枝

國立台灣大學資訊管理學系副教授
美國德州州立大學達拉斯分校管理科學博士

專長：供應鏈管理；作業管理

電話：(02)23630231 轉 3488

傳真：(02)23621327