

行政院國家科學委員會專題研究計畫 成果報告

供應鍊管理多目標主規劃排程問題之研究

計畫類別：個別型計畫

計畫編號：NSC92-2416-H-002-013-

執行期間：92年08月01日至93年07月31日

執行單位：國立臺灣大學資訊管理學系暨研究所

計畫主持人：陳靜枝

共同主持人：陳書儀

計畫參與人員：謝志祥

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 93 年 11 月 4 日

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

供應鍊管理多目標主規劃排程問題之研究

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 92-2416-H-002-013-

執行期間：92年 8月 01日至 93年 07月 31日

計畫主持人：陳靜枝

共同主持人：陳書儀

計畫參與人員：

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立臺灣大學資訊管理學系

中華民國 93 年 10 月 25 日

供應鏈管理多目標主規劃排程問題之研究

陳靜枝

Dept. of Information Management
National Taiwan University
50, Lane 144, Sec. 4, Keelung Road
Taipei, Taiwan 106
R.O.C.

摘要

本研究採用先進規劃排程的觀念，同步考慮訂單需求與產能分配下處理多張訂單在供應鏈中的主規劃排程；並釋放了訂單一定要在交期前完成的限制，成為在訂單延遲最少的前提下尋找總成本最低排程計劃的多目標最佳化問題。由於此類供應鏈網路的問題一般採用線性規劃模式或混合整數線性規劃模式搭配目標規劃方式來求取最佳解，但隨著供應鏈網路問題的複雜度增加，線性規劃模型中的變數與限制式個數快速成長，通常會使模型需要花費大量的時間來求解。因此本研究在描述問題同時也將提出問題的線性規劃模型，此外並提出一啟發性演算法，以快速地解決供應鏈網路下主規劃排程的問題。

關鍵字： 供應鏈管理、先進規劃排程、主規劃排程、線性規劃、啟發性演算法

ABSTRACT

This study focuses on the master planning of “Advanced planning and scheduling.” By considering a final product and its relationship with the global supply chain structure, the first objective is to minimize the total delays of orders, and the second objective is to minimize the sum of production cost, processing cost, transportation cost and inventory holding cost under the constraints of limited capacities. In the previous studies, “Linear Programming,” “Mixed Integer Linear Programming” and “Goal Programming” are popular used to solve these kinds of problems related to supply chain management. However, with the increasing complexities of the supply chain related problems, the numbers of variables and constraints in the LP models grow rapidly. It takes a lot of computer time to solve these problems if there are feasible. Nevertheless, if the LP models result to no feasible solutions, the cause of infeasible can not be identified. Therefore, this study proposes a heuristic algorithm that is more informative and flexible than LP to solve supply chain related problems. The heuristic algorithm can indicate the status of orders and allocations of capacities and searches out feasible solutions more quickly.

Keywords: Supply Chain Management, Master Planning, Heuristic Algorithm, Advanced planning and scheduling (APS), Multiple-goal Optimization

一、研究動機

在目前全球化競爭的環境下，顧客要求訂單快速回應，必須透過良好的供應鏈管理運作，才能有效率地規劃整體供應鏈的資源分

配使用，儘快地將最終產品送給顧客。而在供應鏈管理的生產規劃上，傳統生產規劃模式，將主規劃排程與執行規劃分開來獨立進行，使需求滿足與產能供給兩者無法完全配合，無法做適當的生產調整[4]，這樣的做法

產生的結果難以實際執行，也無法用於快速回應的供應鏈環境中。

先進規劃排程（APS）觀念[4]，整合了傳統主規劃排程與執行規劃，同步考慮訂單需求與產能分配，解決了傳統生產規劃模式分別運作造成的缺點，同時亦具有即時規劃、快速回應的特性。本研究就是屬於先進規劃排程中，處理多張訂單在供應鏈中的主規劃排程問題。

但是目前供應鏈網路中多張訂單生產主規劃排程的解決，普遍會存在下列問題：第一，未考慮產品結構[1]，使供應鏈問題中無法反映原物料與最終產品的相關運作；第二，未考慮完整的供應鏈架構[1][5]，只考慮供應鏈網路部份構面下的子問題，得到部份構面下的局部最佳解，而局部最佳解的加總並不同於整體的最佳化；第三，採用線性規劃（LP）或混合整數線性規劃模式（MILP）求解[2][5]，由於問題的構面龐大，涵蓋產品、網路節點、時間軸與訂單等多個維度，線性規劃模式的變數與限制式個數會大幅增加，增加解題的時間並需要大量的電腦資源，甚至於在混合整數線性規劃模式中，因為整數變數太多，更容易導致無法求解的狀況；第四，只針對單一目標最佳化發展，而無法滿足多種目標的規劃需求，例如只能使生產運輸成本最佳，但卻造成多張訂單延遲。因此本研究要考慮單一最終產品的產品結構下的完整供應鏈網路架構，在多張訂單與多期的環境中，以啟發性演算法有效率地解決多目標最佳化的主規劃排程問題。

二、文獻探討

學者 Rohde[10]等對先進規劃排程的系統架構，分為九個模組：包括長期規劃的策略網路規劃、中期規劃的主規劃排程和需求規劃與短期規劃的物料需求規劃、生產規劃、生產排程、配送規劃、需求滿足與運輸規劃，所有的模組間並不是各自獨立的，而彼此之間互相影響。本研究則是在給定網路架構與確定需求下，同步考慮需求滿足與物料與產能供給，以進行最小化整體成本的生產處理、存貨與運輸規劃，屬於先進規劃排程中的主規劃排程部份。

Cohen 與 Lee[7]將龐大的供應鏈問題切

割為四個子模組：包括物料控制、生產控制、完成品存貨管理與配銷網路控制，而整體系統最佳化，由四個子模組的最佳解，協調整合出整體的近似最佳解，整體系統的決策是由局部最佳解整合而成，可能不是真正整體的最佳解，但是這樣的方法簡單而實際可運用；本研究的問題則是針對完整的供應鏈網路，包括供給、生產與配送等主要部份，同時考慮來進行研究，而不是只由供應鏈中某幾塊子模組求解，再整合為整體的決策。

徐[1]的廠商組合問題研究，為在由供應商、製造商、零售商到配銷商的供應鏈網路架構中，考慮網路各成員生產、配送原物料、半成品與最終產品間產品的組成結構關係，採用最短路徑演算法尋找整體生產、運送成本最低的廠商組合。其研究只用來尋找一供應鏈網路中成本最佳的廠商組合，並沒有考慮組合成員的產能狀況，當廠商組合成員的產能不足時，尋找出的廠商組合並無法使用。

謝[5]的主規劃排程演算法研究，考慮單一最終產品的產品結構，由供應商、製造商、零售商到配銷商的供應鏈網路架構，在生產處理與運輸成本的最小化目標與產能和交期的限制下以啟發性演算法進行訂單的規劃排程，研究中以徐[1]的演算法尋找網路中最佳廠商組合，並進而考慮產能進行訂單排程。而研究模式中忽略存貨成本，將導致排程結果中可能存在大量存貨，本研究則補足存貨成本這項問題，並增加模式的彈性，允許部份限制式放鬆，進行多目標的規劃排程。

三 問題描述與線性規劃模型

3.1 問題描述

本研究屬於先進規劃排程中的主規劃排程部份，考慮完整供應鏈網路架構模型與單一最終產品的產品結構，以間斷時間模式來解決多目標下多張訂單多期的主規劃排程問題。本研究假設已知(1) 產品結構、(2) 供應鏈網路架構與(3) 訂單資料等三項資訊。產品結構為一表示產品元件組成關係與數量的樹狀架構，以圖 3-1 為例，表示最終產品 PO5 的產品結構。而訂單資訊則包括訂單需求商品、訂單需求量、交貨時距與單位延遲成本等。

供應鏈網路架構，主要由節點與連結所構成。節點包括供應鏈網路的起點、終點與

供應商、製造商、配銷商、零售商等四種成員，表現產品的生產、處理與儲存的發生，而節點之間則可能存在連結，表示產品的物流運輸，構成多層上下游關係的網路架構。而網路架構為不存在迴圈、各節點對單一產品只具有單一功能（例如只具有生產製造功能或只具有處理功能）的網路架構。圖 3-2 表示一供應鏈網路的狀況，涵蓋產能限制、生產運輸與存貨持有單位成本等資訊。

3.2 假設條件

本研究的假設條件如下：

1. 本研究為確定模式，所有的資訊，都是已知的資訊，不存在不確定性。
2. 供應鏈網路為中央控制型，網路中各節點與連結的產能與成本資訊可以完全分享，而對於整體規劃結果的生產運輸計劃也可以完整執行。
3. 考慮的成本為生產運送相關的變動成本，包括生產或處理成本、運輸成本、存貨持有成本與訂單延遲成本，不考慮固定成本部份；計算方式為單位成本乘以當期發生成本的數量。
4. 不考慮生產或運輸的經濟規模效應，因此不具有生產、處理或運輸最小批量的限制。
5. 訂單可以任意拆解其需求為數張子訂單，子訂單間可分別由不同的生產路徑來滿足。而各節點對於訂單的處理，每期處理多少，下一期期初就可以往下游節點運輸多少，具有無限切割特性[2]。
6. 供應鏈網路架構中產能與成本資訊各期相同。
7. 相同產品的存貨成本在下游節點較上游節點為高。
8. 本研究採間斷時間模式，供應鏈網路節點，期初收貨而期末送貨。

3.3 線性規劃模型

為了要反映訂單詳細生產規劃（訂單拆解為多張子訂單，而經由多棵生產樹處理），在建立線性規劃模型之前必須要先透過徐[1]的演算法，找出供應鏈網路上所有可能的生產樹，並將生產樹的維度加到線性規劃模型中，而線性規劃模型如下：

(1) 相關參數或變數的下標：

i, j, g ：用來表示節點
 r ：用來表示訂單，值為 $1 \sim R$ 。
 t ：用來表示時距，值為 $1 \sim T$ 。
 p, q ：用來表示產品或物料
 k ：用來表示生產樹，由步驟一徐[1]的演算法找出，值為 $1 \sim K$ 。

(2) 參數部份：

END：供應鏈的結束節點。
 V ：供應鍊網路中所有節點所構成集合。
 V' ：供應鍊網路中除了結束節點 END 外，其他所有節點所構成之集合。
 V^M ：供應鍊網路中具有生產功能節點所構成之集合。
 V^P ：供應鍊網路中具有運輸功能節點所構成之集合。
 L ：供應鍊網路中所有連結 (i, j) 所構想之集合。
 LT_{ip} ：任何二節點 (i, j) 間傳送產品 p 的前置時距。且 $(i, j) \in L$ 。
 DUE_r ：訂單 r 之到期日，以時距計。最大值為 T 。
 DEM_{rp} ：產品 p 的訂單 r 在時距 DUE_r 之需求量。
 CTX_{ipt} ：節點 i 在時距 t 生產產品 p 的產能限制。 $i \in V^M$ 、 t 為 $1 \sim T$ 。
 CTP_{ipt} ：節點 i 在時距 t 處理產品 p 的產能限制。 $i \in V^P$ 、 t 為 $1 \sim T$ 。
 CTS_{ijpt} ：二節點 (i, j) 間在時距 t 傳送產品 p 的產能限制。 $(i, j) \in L$ 、 t 為 $1 \sim T$ 。
 TC_{ijp} ：在二節點 (i, j) 間運輸產品 p 之單位運輸成本。 $(i, j) \in L$ 。
 PC_{ip} ：在節點 i 製造產品 p 之單位生產變動成本。 $i \in V^M$ 。
 QC_{ip} ：在節點 i 處理產品 p 之單位處理變動成本。 $i \in V^P$ 。
 HC_{ip} ：在節點 i 儲存產品 p 之單位持有成本。 $i \in V$ 。
 $Par(p)$ ：產品 p 的下游產品集合。即直接由 p 構成的產品集合。
 Prd_i ：在節點 i 生產的產品或半成品。
 Com_i ：運到節點 i 的原物料集合。
 USE_{pq} ：製造一單位產品 q 所需要產品 p 的數量。其中 $q \in Par(p)$ 。

RAT_{pq} : 製造一單位最終產品 p 所需產品 q 的總數量。

NK_{ik} : 為 1 表示生產樹 k 包括節點 i, 否則為 0。
 $i \in V$ 、k 為 1~K。

DC_{rp} : 訂單 r 物料 p 的單位缺貨成本, 每期累積計算。

(3) 決策變數部份:

X_{iprk} : 在時距 t 於生產樹 k 上節點 i 生產產品 p 的訂單 r 之數量。 $i \in V^M$ 、t 為 1~T。

S_{ijprk} : 在時距 t 於生產樹 k 上兩節點 (i, j) 間運輸產品 p 的訂單 r 之數量。 $(i, j) \in L$ 、t 為 0~T。

I_{iprk} : 在時距 t 於生產樹 k 上節點 i 產品 p 的訂單 r 之期末存貨數量。 $i \in V$ 、t 為 0~T。

● 目標函式:

$$\begin{aligned} \text{Min } P_1 & \left\{ \sum_r \sum_p DC_{rp} \sum_{i \in V^M} \left(DEM_r - \sum_{k=1}^K I_{END,prk} \right) \right\}, \\ P_2 & \left\{ \sum_{i \in V^M} \sum_p PC_{ip} \sum_{r=1}^T \sum_{k=1}^K X_{iprk} + \sum_{i \in V^P} \sum_p QC_{ip} \sum_{r=1}^T \sum_{(g,j) \in L, t \geq LT_{gp}} \sum_{k=1}^K S_{gipr(t-LT_{gp})k} \right. \\ & \left. + \sum_{(i,j) \in L} \sum_p TC_{ijp} \sum_{r=1}^T \sum_{k=1}^K S_{ijprk} + \sum_{i \in V^P} \sum_p HC_{ip} \sum_{r=1}^T \sum_{k=1}^K I_{iprk} \right\} \end{aligned}$$

● 限制式:

$$\begin{aligned} (a.) & \sum_r \sum_{k=1}^K X_{iprk} \leq CTX_{ipt} \quad \forall i \in V^M, p, t = 1 \sim T \\ (b.) & \sum_r \sum_{g,j \geq LT_{gp}} \sum_{k=1}^K S_{gipr(t-LT_{gp})k} \leq CTP_{ipt} \quad \forall i \in V^P, (g,i) \in L, p, t = 1 \sim T \\ (c.) & \sum_r \sum_{k=1}^K S_{ijprk} \leq CTS_{ijpt} \quad \forall (i,j) \in L, p, t = 1 \sim T \\ (d.) & \sum_{i \in V^P} \sum_{r=1}^T \sum_{k=1}^K S_{i,END,pr(t-LT_{END,p})k} = DEM_{rp} \quad \forall p, r \\ (e.) & I_{gpr,(t-1)k} + X_{gprtk} - \sum_{(g,j) \in L} S_{giprk} = I_{gprk} \\ & \quad \forall g \in V^M, p = Prd_g, r, t = 1 \sim T, k = 1 \sim K \\ (f.) & I_{gpr,(t-1)k} + \sum_{(i,g) \in L, t \geq LT_{gp}} S_{igpr,(t-LT_{gp})k} - \sum_{q \in Par(p)} X_{gqrk} \times USE_{pq} = I_{gprk} \\ & \quad \forall g \in V^M, p \in Com_g, r, t = 1 \sim T, k = 1 \sim K \\ (g.) & I_{gpr,(t-1)k} + \sum_{(i,g) \in L, t \geq LT_{gp}} S_{igpr,(t-LT_{gp})k} - \sum_{(g,j) \in L} S_{jiprk} = I_{gprk} \\ & \quad \forall g \in V^P, p, r, t = 1 \sim T, k = 1 \sim K \\ (h.) & \sum_{r=1}^T \sum_{k=1}^K X_{iqrk} = DEM_{rp} \times RAT_{pq} \quad \forall i \in V^M, q, p, r \\ (i.) & I_{gpr,0k} = 0 \quad \forall g \in V, p, r, k = 1 \sim K \\ (j.) & I_{END,prk} = 0 \quad \forall p, r, t < DUE_r, k = 1 \sim K \\ (k.) & S_{ijpr,0k} = 0 \quad \forall (i,j) \in L, p, r, k = 1 \sim K \\ (l.) & X_{iprk} \geq 0 \quad \forall i \in V^M, p, r, t = 1 \sim T, k = 1 \sim K \\ (m.) & I_{iprk} \geq 0 \quad \forall i \in V, p, r, t = 1 \sim T, k = 1 \sim K \\ (n.) & S_{ijprk} \geq 0 \quad \forall (i,j) \in L, p, r, t = 1 \sim T, k = 1 \sim K \\ (o.) & \sum_r X_{iprk} \leq CTX_{ipt} \times NK_{ik} \quad \forall i \in V^M, p, t = 1 \sim T, k = 1 \sim K \\ (p.) & \sum_r \sum_{g,j \geq LT_{gp}} \sum_{k=1}^K S_{gipr,(t-LT_{gp})k} \leq CTP_{ipt} \quad \forall i \in V^P, (g,i) \in L, p, t = 1 \sim T \end{aligned}$$

目標函式包括兩部分, 第一部份先最小化所有訂單的延遲成本; 第二部份在最小化所有生產處理變動成本、運輸變動成本與存貨持有成本總和。

限制式中(a)、(b)與(c)代表各生產節點、配送節點與連結的產能限制。(d)代表訂單的需求滿足限制。(e)代表各生產節點所生產產品或半成品的存貨平衡限制。(f)代表各生產節點原物料的存貨平衡限制。(g)代表各配送節點所處理產品的存貨平衡限制。限制式(h)代表各節點的生產量對每筆訂單來說必恰好用以滿足訂單要求。(i)代表各節點在第 1 期期初存貨初始化。(j)代表各訂單在其交期前不能將最終產品運到供應鏈終點, 強制交期前運至零售商的最終產品發生存貨成本。(k)代表在第 1 期期初各節點間運輸量初始化。(l)、(m)與(n)代表變數為非負值的值域限制。(o)表示生產樹上各生產節點的生產上限, 不在生產樹上的節點對此生產樹則不能有生產量; (p)表示生產樹上各配送節點的處理上限, 不在生產樹上的配送節點對此生產樹則不能有處理量。

● 線性規劃模型的限制

透過線性規劃模型來模式化、解決這個問題, 會具有以下幾個缺點:

1. 線性規劃模型複雜解題時間長

本研究要解決的問題, 包括了完整的供應鏈網路架構、產品結構、多張訂單、多個規劃時距, 因此問題的複雜度, 會與供應鏈網路節點、連結數、生產樹個數、產品結構節點、層數與訂單數相關。線性規劃模型中的變數或限制式, 必須針對網路節點或連結、產品、訂單、時距與生產樹等五個維度分別考量定義。當同時多個維度增加元件, 則與這些維度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。解題的時間也會隨模型的複雜度而成長, 因此解題的時間也會快速成長。

2. 線性規劃模型無解時, 無法知道那些訂單或限制式造成無解

以線性規劃模型解問題, 最大的缺點就是不保證有解。當問題無解時, 我們希望知道規劃排程的問題已經解決到那個程度, 有

多少訂單已經完成排程而剩餘產能為多少，根據這些資訊進而可以增加某些節點的產能或放棄某些訂單以得到部份的排程計劃。但線性規劃模型無解時並無法指引該如何修正問題，只知道問題無解，不知道那些限制式造成無解，也不知道那幾張訂單的需求造成無解。

四、啟發性演算法

4.1 演算法主要流程

本研究的啟發性演算法，主要包括流程(1)~(5)如下：

- (1) 確認單一功能節點：演算法一開始將檢查供應鏈網路上的各個節點，確認節點為單一功能節點，用以確保能採用徐[1]的演算法來找尋網路上最佳的廠商組合(及最佳生產樹)。
- (2) 產能與成本轉換：本研究考慮了產品結構，在此步驟中將供應鏈網路架構中所有產能限制與單位成本等資訊，根據上下游產品的組成數目比例轉換為以最終產品為基準，使排程時不必再考慮組成個數的問題。
- (3) 網路圖形成本設定：徐[1]演算法只根據網路連結的成本找尋生產樹；因此本步驟將節點的生產或處理單位成本，加到節點後的連結上，使任一連結上的成本包括原本的單位運輸成本與前一節點的單位生產成本或單位處理成本。
- (4) 訂單排序：多張未規劃的訂單必須依據訂單的各戶重要性、訂單需求數量、訂單交期與訂單延遲成本等性質，決定一個規劃排程的順序。
- (5) 訂單規劃排程：訂單排序後，則由具有最高優先順序的訂單開始進行規劃排程。詳細流程見下一子節

4.2 規劃排程演算法

規劃排程的詳細流程主要如圖 4-1：

- (1) 是否有未規劃之訂單：開始先檢查是否有未進行規劃排程的訂單，如果沒有，表示所有訂單都已完成規劃排程，因此演算法結束；反之若有未規劃之訂單，則進行下一步驟。

- (2) 取出一張未規劃訂單：當存在未規劃之訂單時，便依訂單排序取出一張訂單進行接下來的規劃排程。
- (3) 還原網路圖形：每次進行訂單規劃排程時，演算法可能會對供應鏈網路進行架構調整或連結成本修改，因此當進行新的一張訂單排程時，必須還原為原本的供應鏈網路架構與成本。
- (4) 尋找最佳生產樹 S：在產品組成結構與供應鏈網路確定後，採用徐[1]的最短路徑演算法尋找最佳的廠商組合，也就是最佳的生產樹 S。
- (5) 檢驗 S 是否存在：若 S 存在則進行下一步驟檢查 S 各節點的產能與存貨使用狀態，反之若 S 不存在表示目前交期下不存在可用的生產樹，延遲訂單交期一期再重新進行規劃。
- (6) 檢查 S 上各節點產能狀況，並尋找 S 的可用產能(feasible capacity; FECP)：由 S 的終點開始根據連結前置時間往前確認每個節點的產能狀態，並判斷是否要發生到存貨成本，而是否符合目前網路圖形的表示。如果不符合目前網路圖形的表示，則要調整連結成本進行步驟(7)，否則尋找考慮最少存貨使用下 S 的 FECP，並依 FECP 進行網路架構調整或分配產能。本步驟詳細演算法列於步驟(9)說明之後。
- (7) 修改網路圖形使節點存貨持有期數達到 NINV：當一節點實際排程發生存貨成本大於網路圖形上成本表示時，需要修改網路圖形上的成本再重新尋找一次生產樹。本步驟將此節點發生的單位存貨成本乘上實際的期數 NINV(i)更新到生產樹 S 的的連結上，並更新 HINV(i)。
- (8) 分配訂單產能：當生產樹 S 的可用產能確定後，即進行訂單需求的滿足：當 $FECP \geq OQ$ 時，表示有足夠的產能可以滿足全部訂單量，因此 S 上所有節點皆分配 OQ 以滿足訂單量，而訂單也完成規劃排程；反之，S 上所有節點皆分配 FECP，滿足部分訂單量，並更新剩餘未規劃訂單量為 $OQ - FECP$ ，回到步驟(6)繼續剩餘訂單量的規劃排程。
- (9) 調整網路圖形：步驟(6) 檢驗生產樹 S 上各節點，若發現一節點由規劃起始時距到

最後可用時距都已無產能時，即考慮所有可能的存貨使用皆無法使其具有可用產能，此處發生的情境，與謝[5]研究中生產樹節點無可用產能發生的情境相同，因此可以採用謝[5]的方法 Copy Cut & Paste 的演算法來進行網路架構的調整，一方面保存所有節點剩餘產能的使用機會，一方面仍可採用徐[1]的演算法來尋找新的生產樹。

● 步驟(6)詳細演算法：

演算法符號說明

S：步驟(4)找出之最佳生產樹
i, j, N：最佳生產樹 S 上的節點
Cap[i,t]：節點 i 在時距 t 的可用產能
NOW：規劃起始時距
LD(N)：節點 N 的最後可用時距
LDCN(N)：節點 N 在最後可用時距的產能
OT：訂單交貨時距
FT：規劃時距
HINV(N)：目前網路圖形中節點 N 使用的存貨期數(網路一開始，所有節點 HINV(N)=0)
NINV(N)：使節點 N 有可用產能所需存貨期數
FECN(N)：使用最少存貨下節點 N 的可用產能
FECP：最少存貨下生產樹 S 的可用產能

演算法

```
(6.1) Initial j=End, LD(j)=OT, FECP=∞
(6.2) For all link i->j and i≠Start
(6.2.1) If LD(j)-LT(i->j)<NOW Goto (9)
      else LD(i)=LD(j)-LT(i->j)
(6.2.2) If Cap[i,LD(i)]=0
(6.2.2.1) Find a max time-bucket : FT
           form LD(i)-1 to NOW
           which let Cap[i,FT]>0
(6.2.2.2) If FT not exists Goto (9)
           else NINV(i)=LD(i)-FT
(6.2.2.3) If NINV(i)>HINV(i) Goto (7)
(6.2.2.4) LD(i)=FT
(6.2.3) FECN(i)=Cap[i,LD(i)]
(6.2.4) if FECP>FECN(i)
           FECP=FECN(i)
(6.2.5) Let j=i
End For
(6.3) Goto (8)
```

(6.1)初始化演算法由 S 的終點 END 開始，設定 S 的可用產能為無限大、END 的最後可用時距為訂單的交貨時距；(6.2)進入迴圈，由 S 的終點往起點確認每個節點；(6.2.1)根據連結的前置時間，計算連結上游節點的最後可用時距，若最後可用時距已在規劃起始時距之前，表示生產樹 S 無可用產能，進入步驟(9)調整網路架構重新尋找生產樹；

(6.2.2)當節點 i 在最後可用時距產能為零時：(6.2.2.1)由節點 i 最後可用時距往前到規劃起始時距，找到使節點 i 有可用產能的時距為止，找到的時距記為 FT；(6.2.2.2)若 FT 存在，計算 FT 與最後可用時距的差距 NINV(i)，即節點 i 存貨持有的期數，若 FT 不存在，表示節點 i 由最遲生產時距往前到規劃起始時距的各期產能皆為零，則生產樹 S 亦無可用產能，進入步驟(9)調整網路架構重新尋找生產樹；(6.2.2.3)檢查網路圖形中節點 i 存貨持有期數資訊 HINV(i)與 NINV(i)的差異，若 NINV(i)較大則表示實際使用 S 所發生的存貨成本將大於步驟(4)尋找出最佳生產樹時所認定 S 的存貨成本，因此要透過步驟(7)更新生產樹 S 實際排程下存貨持有期數的資訊，再重新尋找生產樹；(6.2.2.4)則更新節點 i 的最後可用時距為 FT，用以檢查上游其他節點的產能資訊；(6.2.3)記錄節點 i 的可用產能；(6.2.4)更新生產樹 S 的可用產能為 S 中所有節點可用產能的最小值；(6.2.5)再往上游節點檢查；(6.3)表示當運算完 S 的所有節點且沒有跳至步驟(7)與步驟(9)，即可進入步驟(8)分配產能。

五、實例分析

本章以兩個實例，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成本與排程時間。其中線性規劃模型以 ILOG 計算，啟發性演算法以 VB6 與 MSSQL 建構。

實例一的產品結構如圖 3-1，供應鏈架構如圖 3-2，處理兩張訂單如表 5-1，啟發性演算法的演算流程如下：

1. 取出訂單 1，需求 400，而供應鏈網路尋找出生產樹 S1 如圖 5-1(a)，單位成本 64，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，還原供應鏈網路圖形，最佳生產樹尋找出如圖 5-1(a)相同生產樹 S1，檢驗生產樹上各節點，發現要使 S1 有產能，V3 要發生一期的存貨成本 NINV(V3)=1，但網路圖形上 HINV(V3)=0，修改網路成本使 V3->M12 連結成本為 9+7=16。
3. 再尋找最佳生產樹得到 S2 如圖 5-1(b)，

單位成本 70，可用產能為 300；可分配產能 300，訂單 2 剩餘訂單量 $600-300=300$ 。

4. 再檢查 S2，檢查到節點 D2 即發現要發生一期的存貨成本 $NINV(D2)=1$ ，但網路圖形上 $HINV(D2)=0$ ，修改網路成本使 $D2 \rightarrow R2$ 連結成本為 $7+7=14$ 。
5. 再尋找最佳生產樹得到 S1 如圖 5-1(c)，單位成本 71，可用產能為 100；可分配產能 100，訂單 2 剩餘訂單量 $300-100=200$ 。
6. 再檢查 S1，檢查到節點 R1 即發現要發生一期的存貨成本 $NINV(R1)=1$ ，但網路圖形上 $HINV(R1)=0$ ，修改網路成本使 $R1 \rightarrow END$ 連結成本為 $5+7=12$ 。
7. 再尋找最佳生產樹得到 S2，但節點 V2 其最後可用時距已在規劃起始時距前，即 S1 無產能，用 Copy cut and Paste 調整網路架構如圖 5-1(d)。
8. 去掉調整後無效節點連結後，尋找最佳生產樹得到 S1 如圖 5-1(e)，但同樣地節點 V2 其最後可用時距已在規劃起始時距前，再用 Copy cut and Paste 調整網路圖形，如圖 5-1(f)。
9. 去掉調整後無效節點連結後剩餘圖形已無法找到生產樹，因而延遲訂單 2 交期為 9，還原供應鏈網路圖形，尋找出最佳生產樹尋找出如圖 5-1(a) 相同生產樹，單位成本 64，可用產能為 400；可分配產能 200，完成訂單 2 排程。
10. 無未規劃訂單，完成訂單規劃排程。

啟發性演算法的排程結果如表 5-2，總成本 66500，訂單 2 有 200 單位的延遲。而透過線性規劃模式求得的最佳解，經過整理後如表 5-3，最小化延遲為 200 單位，不含延遲部分的總成本為 66500。由實例一結果可發現啟發性演算法的總成本與線性規劃模式的結果相同，達到最佳解。

實例二的產品結構與供應鏈架構與實例一相同，處理二十張訂單如表 5-4，啟發性演算法的排程結果如表 5-5，總成本 1960700，所有訂單有 8600 單位的延遲。而透過線性規劃模式求得的最佳解，最小化延遲為 8600 單位，不含延遲部分的總成本為 1960700。由實例二結果可發現啟發性演算法的總成本與線性規劃模式的結果亦相同，達到最佳解。而在解題時間上，啟發性演算法需要 103.797

秒，線性規劃模式則需要 264 秒(包括求最小化延遲的 218.51 秒與最小化總成本的 45.49 秒)，啟發性演算法明顯地較快速。

六、結論

本研究提出一啟發性演算法，在單一最終產品的產品結構關係與完整供應鏈網路架構中，考量生產、運輸、存貨與訂單延遲成本，於短時間內有效率地解決多目標最佳化的主規劃排程問題。

未來研究方向將考慮加入其他的排程目標，例如釋放供應鏈網路節點的產能限制，允許使用超額產能，而加入最小化超額產能的使用量的目標；另外也將討論多目標間的組合方式，除了本研究以不同優先權滿足各個目標之外，多個目標也可透過單位轉換與重要性權數加總為相同基準的單一總目標。

七、參考文獻

- [1] 徐昆羿, 供應鏈網路之轉換—以最短路徑演算法解決廠商組合問題, 台灣大學資訊管理研究所碩士論文, 民國 89 年。
- [2] 高郁凱, 以連續時間模式解決主規劃排程問題, 台灣大學資訊管理研究所碩士論文, 民國 91 年。
- [3] 陳書儀, 考慮產品結構樹於供應鏈網路流量管理之研究, 台灣大學資訊管理研究所博士論文, 民國 89 年。
- [4] 蔡與哲, 運用先進排程之概念建立多廠生產的資源分配與生產規劃模式, 台灣大學商學研究所碩士論文, 民國 89 年。
- [5] 謝志欣, 供應鏈管理之主規劃排程演算法—以最短路徑演算法滿足所有訂單需求, 台灣大學資訊管理研究所碩士論文, 民國 90 年。
- [6] Arntzen, B. C., G. G. Brown, T. P. Harrison, and L. L. Trafton, "Global Supply Chain Management at DEC," *Interface*, Vol.25, No. 1, 1995, pp.69—92.
- [7] Cohen, M. A. and H. L. Lee, "Strategic Analysis of Integrated Production-Distribution Systems: Models and Methods," *Operations Research*, Vol.36, No.2, 1988, pp.216—228.
- [8] Min, H., and G. Zhou, "Supply chain modeling: past, present and future,"

Computers & Industrial Engineering, Vol.43, 2002, pp.231—249.

- [9] Simchi-Levi, D., P. Kaminsky, and E. Simchi-Levi, “Designing and Managing the Supply Chain : Concepts, Strategies, and Case Studies,” USA, The McGraw-Hill Company, 2000.
- [10] Stadtler, H. and C. Kilger, “Supply Chain Management and Advance Planning : Concepts, Models, Software and Case Studies,” Springer-Verlag Berlin Heidelberg, 2000.

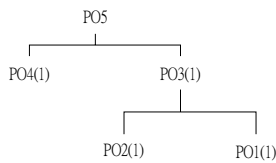


圖 3-1 : PO5 產品組成結構圖 (來源 : [5])

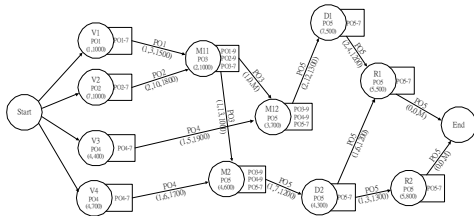


圖 3-2 : 供應鍊網路架構 (來源 : [5]修改)

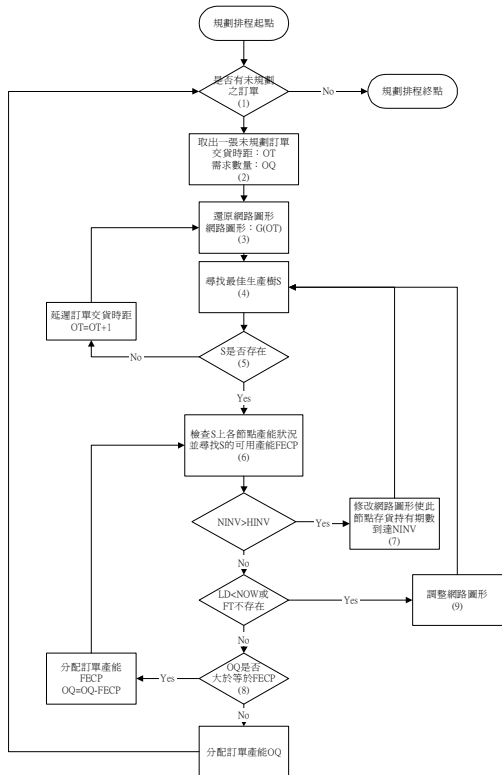


圖 4-1 : 規劃排程詳細流程 (來源 : 本研究)

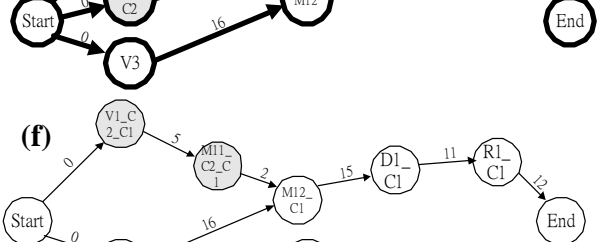
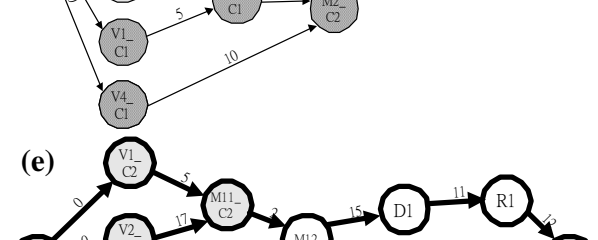
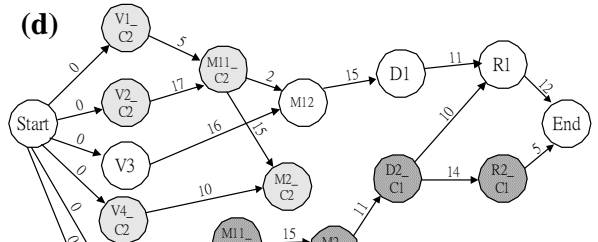
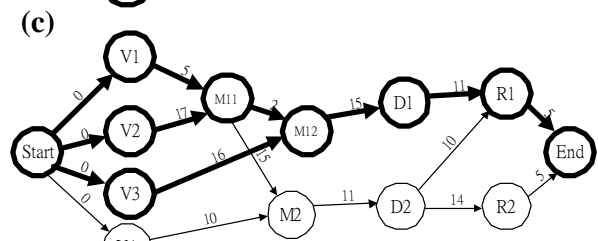
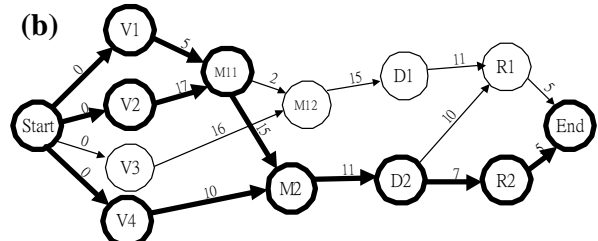
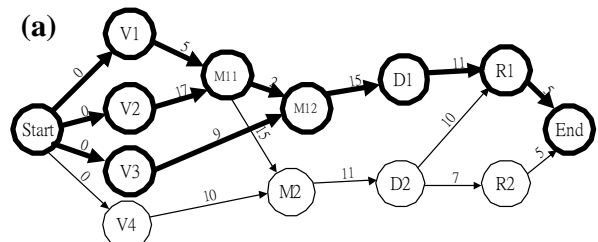


圖 5-1 演算法運作(來源 : 本研究)

表 5-1 實例一訂單狀況

| 訂單編號 | 需求數量 | 交貨時距 | 延遲成本 | 排序結果 |
|------|------|------|------|------|
| 1 | 400 | 8 | 2 | 1 |
| 2 | 600 | 8 | 2 | 2 |

表 5-2 實例一演算法結果

| 訂單編號 | 生產樹單位成本 | 處理數量 | 交貨時距 |
|------|---------|------|------|
| 1 | 64 | 400 | 8 |
| 2 | 70 | 300 | 8 |
| 2 | 71 | 100 | 8 |
| 2 | 64 | 200 | 9 |

表 5-3 實例一線性規劃模型結果

| 訂單編號 | 生產樹 | 處理數量 | 交貨時距 |
|------|-----|------|------|
| 1 | PT1 | 300 | 8 |
| 1 | PT3 | 100 | 8 |
| 2 | PT1 | 400 | 8 |
| 2 | PT1 | 200 | 9 |

表 5-5 實例二演算法結果

| 訂單 | 生產樹成本 | 數量 | 交貨時距 | 訂單 | 生產樹成本 | 數量 | 交貨時距 |
|----|-------|-----|------|----|-------|-----|------|
| 1 | 64 | 400 | 8 | 10 | 64 | 400 | 26 |
| 1 | 70 | 300 | 8 | 10 | 70 | 300 | 26 |
| 1 | 71 | 100 | 8 | 10 | 77 | 300 | 26 |
| 1 | 64 | 400 | 9 | 10 | 64 | 400 | 27 |
| 1 | 70 | 300 | 9 | 11 | 64 | 400 | 28 |
| 2 | 64 | 400 | 10 | 11 | 70 | 300 | 28 |
| 2 | 70 | 300 | 10 | 11 | 77 | 300 | 28 |
| 2 | 85 | 100 | 10 | 11 | 64 | 400 | 29 |
| 2 | 85 | 100 | 10 | 12 | 64 | 400 | 30 |
| 2 | 64 | 400 | 11 | 12 | 70 | 300 | 30 |
| 2 | 70 | 200 | 11 | 12 | 77 | 300 | 30 |
| 3 | 64 | 400 | 12 | 12 | 64 | 400 | 31 |
| 3 | 70 | 300 | 12 | 13 | 64 | 400 | 32 |
| 3 | 77 | 100 | 12 | 13 | 70 | 300 | 32 |
| 3 | 99 | 100 | 12 | 13 | 77 | 300 | 32 |
| 3 | 106 | 100 | 12 | 13 | 64 | 400 | 33 |
| 3 | 64 | 400 | 13 | 14 | 64 | 400 | 34 |
| 3 | 70 | 100 | 13 | 14 | 70 | 300 | 34 |
| 4 | 64 | 400 | 14 | 14 | 77 | 300 | 34 |
| 4 | 70 | 300 | 14 | 14 | 64 | 400 | 35 |
| 4 | 77 | 200 | 14 | 15 | 64 | 400 | 36 |
| 4 | 120 | 100 | 14 | 15 | 70 | 300 | 36 |
| 4 | 120 | 100 | 14 | 15 | 77 | 300 | 36 |
| 4 | 64 | 400 | 15 | 15 | 64 | 400 | 37 |
| 5 | 64 | 400 | 16 | 16 | 64 | 400 | 38 |
| 5 | 70 | 300 | 16 | 16 | 70 | 300 | 38 |
| 5 | 77 | 300 | 16 | 16 | 77 | 300 | 38 |
| 5 | 134 | 100 | 16 | 16 | 64 | 400 | 39 |
| 5 | 64 | 400 | 17 | 17 | 64 | 400 | 40 |
| 6 | 64 | 400 | 18 | 17 | 70 | 300 | 40 |
| 6 | 70 | 300 | 18 | 17 | 77 | 300 | 40 |
| 6 | 77 | 300 | 18 | 17 | 64 | 400 | 41 |
| 6 | 64 | 400 | 19 | 18 | 64 | 400 | 42 |
| 7 | 64 | 400 | 20 | 18 | 70 | 300 | 42 |
| 7 | 70 | 300 | 20 | 18 | 77 | 300 | 42 |
| 7 | 77 | 300 | 20 | 18 | 64 | 400 | 43 |
| 7 | 64 | 400 | 21 | 19 | 64 | 400 | 44 |
| 8 | 64 | 400 | 22 | 19 | 70 | 300 | 44 |
| 8 | 70 | 300 | 22 | 19 | 77 | 300 | 44 |
| 8 | 77 | 300 | 22 | 19 | 64 | 400 | 45 |
| 8 | 64 | 400 | 23 | 20 | 64 | 400 | 46 |
| 9 | 64 | 400 | 24 | 20 | 70 | 300 | 46 |
| 9 | 70 | 300 | 24 | 20 | 77 | 300 | 46 |
| 9 | 77 | 300 | 24 | 20 | 64 | 400 | 47 |
| 9 | 64 | 400 | 25 | | | | |