

東 海 大 學

工業工程與經營資訊研究所

碩士論文

題目：多廠區生產系統之訂單分配模式
—以液晶顯示器產業為例

研 究 生：王世欽

指導教授：王立志 博士

中 華 民 國 九 十 一 年 六 月

An Order Allocation Model in a Multiple Site LCD
Manufacturing Industry

by

Shih-Chin Wang

Advisor: Dr. Li-Chih Wang

A Thesis

Submitted to the Institute of Industrial Engineering
and Enterprise Information
at Tunghai University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Industrial Engineering

June 2002

Taichung, Taiwan, Republic of China

多廠區生產系統之訂單分配模式

—以液晶顯示器產業為例

研究生：王世欽

指導教授：王立志 博士

東海大學工業工程與經營資訊研究所

摘要

面對全球市場的不景氣，高科技產業面臨到整個供應鏈上產銷失調的窘境，所以導入供應鏈規劃的資訊系統已經刻不容緩。由於不同工廠之間，其生產流程的相互依賴性是很複雜的，過去解決多廠區訂單分配問題，最常用的方法為線性規劃及啟發式(heuristics)演算法。若使用線性規劃來解決此類問題，一旦限制條件過多且變數過於複雜，將很難有效益的獲得合理解的規劃；若利用簡單的啟發式法則，往往又無法同時考量企業的資源限制及解決成本最佳化的問題。有鑑於此，本論文將提出以產銷平衡(supply-demand balancing)為考量基礎之訂單分配最佳化的決策模式，供業界做為參考，進而減少銷售與生產之間的失調。研究目的可歸納如下：

- (1) 提出應用於台灣產業之多廠區生產系統下產銷平衡的規劃邏輯。
- (2) 針對多廠區生產環境的特性，發展產銷平衡的訂單分配模式。
- (3) 實作出多廠區生產系統下產銷平衡訂單分配系統。

整體而言，本論文運用了基因演算法之最佳化搜尋的特性，同時考量限制條件滿足技術利用縮小搜尋解空間(solution space)，進一步發展出能快速求解且兼具有最佳化搜尋能力的訂單分配演算法。最後，我們亦將應用本論文所發展的產銷平衡訂單分配模式於台灣LCD產業，以驗證並評估其效益。

關鍵字：供應鏈規劃、多廠區產銷平衡、基因演算法、限制條件滿足技術

An Order Allocation Model in a Multiple Site LCD Manufacturing Industry

Student: Shih-Chin Wang

Advisor: Dr. Li-Chih Wang

Institute of Industrial Engineering and Enterprise Information
Tunghai University

Abstract

Due to the trend of global marketing and competition, the implementation of supply chain planning is timing and necessary. For an enterprise with multiple manufacturing factories, the production flow among different factories are highly dependent and complicated, mathematical planning and heuristics approaches are usually employed. However, it is very difficult to effectively obtain a feasible supply plan once there are too many constraints and the relationships are too complicated. If simple heuristic rules are employed, many enterprise resource constraints and cost issues cannot be concerned. Therefore, the objective of this research is to develop an order allocation decision model, to maintain the supply (i.e., material and capacity) and demand (customer and forecast orders) balance in a multi-site manufacturing environment. In other words, the objectives of this research are:

- (1) To develop a supply and demand balancing process/logic which will consider the characteristics of Taiwan's industry with multiple manufacturing factories.
- (2) To develop an order allocation decision model which can maintain the supply (i.e., material and capacity) and demand (customer and forecast orders) balance in a multi-site manufacturing environment.
- (3) To develop the order allocation system based on the model proposed in this research.

In summary, this research will first survey the literature of supply chain planning and multi-site production systems, then develop a supply

and demand balancing process/logic. Consequently, an order allocation decision model which takes into account important enterprise resource constraints and cost issues will be developed. In this research, a constraint-based genetic algorithm for the order allocation model is also developed to efficiently and effectively allocate orders in a multi-site manufacturing environment. Finally, the proposed order allocation model will be applied in a Taiwan's LCD manufacturer to illustrate the applicability and evaluate the effectiveness of the proposed order allocation model developed in this research.

Key Words: Supply Chain Planning, Multi-site Supply and Demand Balancing, Genetic Algorithm (GA), Constraint Satisfaction Technique (CST)

致謝

二年半的碩士生涯轉眼間就過去了！這段求學的日子裡莫過於感謝指導教授—王立志老師的磨練及苛求，使得我更加茁壯及成熟。口試期間更是感謝王偉華老師、袁明鑑老師、周忠信老師及育仁學長的建議及指導，讓本論文改進了不少的缺點。

想起住在斗六的時光，與明峰、志合、辰仰、敏樟、良彬、在明、文偉、文財、重光等學長們及明達、育煒一起學習成長的一點一滴，不知不覺中也跨出了碩士生涯的第一步。在接下來的時間裡，元銘、銘榮等畢業學長們，和聖倫、仲為、志傑、哲源等同窗好友們及學弟妹們，有你們陪伴著我，讓生活增色不少。當然，其他的朋友們也少不了對妳們的感謝。

最後，在此將本論文獻給各位，因為有你們在我的身邊，才有它的出現！

王世欽 謹誌於
東海大學工業工程與經營資訊研究所
企業資源整合系統研究室
中華民國九十一年六月

目錄

摘要.....	I
ABSTRACT.....	II
致謝.....	IV
目錄.....	V
表目錄	VII
圖目錄	IX
第一章 緒論	1
1.1. 研究背景與動機.....	1
1.2. 研究目的與範圍.....	2
1.3. 研究方法與步驟.....	5
1.4. 論文架構	6
第二章 文獻探討	8
2.1. 多廠區生產系統之探討.....	8
2.1.1. 多廠區生產系統的特性.....	8
2.1.2. 多廠區生產系統的架構.....	12
2.1.3. 多廠區生產規劃的方式.....	15
2.2. 商用軟體的作法.....	24
2.2.1. <i>Manugistics CBMP</i>	24
2.2.2. <i>Adexa SCP</i>	27
2.3. 限制條件滿足問題與基因演算法之探討.....	30
2.3.1. 限制條件滿足問題.....	30
2.3.2. 基因演算法之相關探討.....	33
2.4. 本章結論	47
第三章 多廠區生產規劃系統的架構與訂單分配模式.....	48

3.1. 多廠區生產規劃系統的架構.....	48
3.2. 多廠區產銷平衡的規劃邏輯.....	59
3.3. 產銷平衡的訂單分配模式之建立.....	64
3.3.1. 假設條件說明.....	64
3.3.2. 符號說明.....	65
3.3.3. 目標函式說明.....	67
3.3.4. 限制條件說明.....	67
第四章 以限制條件為基礎之基因演算法的設計與應用	74
4.1. 染色體的表示法.....	74
4.2. 適應性函式說明.....	75
4.3. 基因運算子的設計原則.....	75
4.4. 以限制條件為基礎之基因演算法的運作流程.....	85
第五章 液晶顯示器產業的範例驗證與評估	88
5.1. 範例情節概述.....	88
5.2. 實驗評估與分析.....	97
5.2.1. 訂單分配模式之可行性.....	97
5.2.2. 基因演算法之求解能力.....	99
第六章 結論與建議	104
6.1. 結論	104
6.2. 未來發展方向.....	104
參考文獻	106
附錄 A.....	111
附錄 B.....	114
附錄 C.....	116
簡歷.....	127

表目錄

表 2.1 全域、區域與運輸排程的建模內容	10
表 2.2 預知式與反應式排程的優缺點比較表	11
表 2.3 供應鏈模型的成本項目之相關研究	22
表 2.4 不同產業的多廠區生產規劃比較表	23
表 2.5 基原字串對照表	44
表 3.1 積壓訂單與填補產能訂單間的挪移	71
表 4.1 染色體的表示法	74
表 5.1 各月彙總的需求表	90
表 5.2 各產品群組的製造及儲存成本	91
表 5.3 各廠區間的運輸成本	91
表 5.4 基因演算法之參數設定表	92
表 5.5 製造訂單的分配表	93
表 5.6 運輸訂單的分配表	94
表 5.7 存貨單的分配表	95
表 5.8 訂單開立表	96
表 5.9 供需平衡的挪移表	96
表 5.10 線性規劃之參數設定表	98
表 5.11 線性模式的實驗數據	98
表 5.12 非線性模式的實驗數據	99
表 5.13 50 次實驗進入收斂狀態的初始代數	99
表 5.14 24 次實驗的最小成本（軟、硬性限制條件）	100
表 5.15 求解方法的比較表	102
表 5.16 24 次實驗的最小成本（硬性限制條件）	102
表 5.17 T 檢定的報表	103
表附錄 C.1 基本輸入資料	116
表附錄 C.2 製造訂單的分配表（線性規劃）	119

表附錄 C.3 運輸訂單的分配表 (線性規劃)	120
表附錄 C.4 存貨單的分配表 (線性規劃)	121
表附錄 C.5 訂單開立表 (線性規劃)	122
表附錄 C.6 供需平衡的挪移表 (線性規劃)	122
表附錄 C.7 製造訂單的分配表 (啟發式法則)	123
表附錄 C.8 運輸訂單的分配表 (啟發式法則)	124
表附錄 C.9 存貨單的分配表 (啟發式法則)	125
表附錄 C.10 訂單開立表 (啟發式法則)	126

圖目錄

圖 1.1 薄膜電晶體液晶顯示器產業的製造流程概念圖	5
圖 1.2 研究方法與進行步驟	7
圖 2.1 全域與區域的排程[36].....	10
圖 2.2 多廠區生產系統的架構[36].....	13
圖 2.3 多廠區生產規劃與控制系統	13
圖 2.4 訂單為主的運輸排程之啟發式法則	17
圖 2.5 MANUGISTICS CBMP 的資料結構之概念圖[25]	25
圖 2.6 ADEXA SCP 的資料結構之概念圖	27
圖 2.7 ADEXA SCP 的產銷平衡規劃方式	28
圖 2.8 ADEXA SCP 之網路排程的排程邏輯	29
圖 2.9 基因演算法之相關研究（一）	34
圖 2.10 基因演算法之相關研究（二）	35
圖 2.11 基因演算法流程圖.....	36
圖 2.12 基因編碼及對應關係	37
圖 2.13 簡單型基因演算法的運作概念圖	40
圖 2.14 穩態型基因演算法的運作概念圖	40
圖 2.15 各種不同的突變運算子運作方式	42
圖 2.16 各種不同的交配運算子運作方式	43
圖 3.1 一般性多廠區生產規劃系統的基本模組架構	48
圖 3.2 訂單分配模組的子模組架構	50
圖 3.3 多廠區生產系統的建模示意圖	51
圖 3.4 製造流程的建模範例	53
圖 3.5 運輸流程的建模範例	54
圖 3.6 採購流程的建模範例	56
圖 3.7 儲存流程的建模範例	57
圖 3.8 以流程導向為建模方式的供應鏈示意圖	59

圖 3.9 多廠區生產系統下產銷平衡的規劃邏輯	60
圖 3.10 供給數量限制的示意圖	69
圖 3.11 積壓訂單與填補產能訂單的示意圖	70
圖 3.12 供給與需求平衡的示意圖	71
圖 3.13 物料投入與產出關係的示意圖	72
圖 3.14 多廠區生產系統的網路結構圖	73
圖 4.1 互換突變運算子的運算邏輯	77
圖 4.2 互換突變運算子的運算程序	78
圖 4.3 均勻交配運算子的運算邏輯	80
圖 4.4 均勻交配運算子的運算程序	81
圖 4.5 排序複製運算子的運算邏輯	83
圖 4.6 排序複製運算子的運算程序	84
圖 4.7 疊置運算子的運算程序	85
圖 4.8 簡單型基因演算法的運作邏輯	86
圖 5.1 薄膜電晶體液晶顯示器的多廠區生產系統	89
圖 5.2 多廠區生產系統的範例示意圖	90
圖 5.3 實驗結果之區域圖	92
圖 5.4 最小成本的直方圖	100
圖 5.5 各期存貨量之折線圖	101
圖附錄 A.1 一般性多廠區生產規劃流程圖	111
圖附錄 B.1 基因演算法之類別圖	114
圖附錄 B.2 訂單分配模式的類別圖	115

第一章 緒論

1.1. 研究背景與動機

面對全球市場的不景氣，高科技產業面臨到整個供應鏈上產銷失調的現象。以筆記型電腦產業為例，2000年就發生整個供需失調的現象，幾家筆記型電腦大廠，都接到了亮麗的訂單預測，但是從2000年年初的液晶顯示（Liquid Crystal Display；LCD）面板缺貨，到年中的被動原件缺貨到年尾的驅動輸入電路（Input Circuit；IC）缺貨，很多沒有做好準備的大廠面臨了有訂單卻做不出來的窘境，這些訂單被迫轉移到有材料的工廠，原來與國外大廠的獨家關係也被迫改變，到了2000年底，有些公司勉強繳出了預期的成績單，大部分的公司卻表現得不盡理想。企業內外的供應鏈規劃變得更加重要。供應鏈規劃的意義就是達到『產銷的平衡機制』，企業希望能對未來的需求做更精確的掌握，藉此擬定出企業的運作方向與目標，這些資訊，對內而言是公司產品線佈局的主要參考依據，對外是上游的原物料供應商的備貨計劃，以期能避免在需求突然產生時後面補給無力的狀況發生[48][49]。

多廠區生產系統的訂單分配問題，可視為不同工廠間的生產流程分派問題，其主要特性分為[36]：

- (1) 不同工廠之間，其生產流程的相互依賴性是很複雜的。
 - a. A工廠所製造的產品是B工廠所需要的。
 - b. 相同的品項可能被不同工廠所製造（可能造成不同的成本）。
 - c. 不同工廠之間的零件運輸問題。
- (2) 全域排程需要的資訊類型是以一般性資料來取代精確資料，例如：
 - a. 以機器群組來替代單一機器的產能資訊。
 - b. 中間或最終產品所估計的製造時間。
- (3) 單一工廠之既存排程系統的整合。

(4) 企業下的所有工廠，其分散的排程活動間合作之必須性。

(5) 單一工廠的實際狀態之不確定性。

為了協調廠區間的運輸，多廠區生產規劃通常處理一個可共享的庫存。因此，企業藉由數個具有互補或替代生產能力的廠區，去規劃每個廠區間的互動合作。綜合上述，在多廠區生產系統的環境中，產品的生產流程是需要使用跨廠區間的資源，亦即產品的生產流程是存在替代性。解決多廠區生產的問題，是要改善不同廠區之間在生產規劃與控制層面的合作問題。

因此，本論文預定要解決多廠區生產規劃的問題可以設定為決定「規劃週期中，單一廠區內需要生產及不同廠區間需要運輸的品項數量。」

1.2. 研究目的與範圍

有鑑於此，本論文提出以產銷平衡 (supply-demand balancing) 為考量基礎之訂單分配的決策模式，供台灣液晶顯示器產業做為參考，進而減少銷售與生產之間的失調問題。其研究目的可歸納如下：

(1) 提出應用於台灣產業之多廠區生產系統下產銷平衡的規劃邏輯。

(2) 針對多廠區生產環境的特性，發展產銷平衡的訂單分配模式。

(3) 實作出多廠區生產系統下產銷平衡訂單分配系統。

多廠區訂單分配問題，一般多為批量決策問題 (lot sizing problem)，其包含訂單的切割 (產品種類、數量)、生產、運送等行為。Timpe 與 Kallrath[39]以一個具有數個生產工廠與銷售點的化學公司為研究對象，以各項生產因子 (如生產數量、工廠產能、物料流量、存貨、運送等) 為限制條件，提出一混合整數線性規劃 (Mixed-Integer Linear Programming; MILP)，期望在公司或一最大的目標下，求得各個工廠最適宜的生產數量。然而，該研究利用混合整數線性規劃構建模式，會產生求解費時的疑慮，而每個規劃週期只能進行一次生產

轉換 (order change) , 也是這個方法的一大限制。

在處理多廠訂單分配問題上, 曾氏[55]也進行了此方面的研究, 其依據客戶訂單資訊 (產品別、數量、交期) 與各工廠生產特性資訊 (製造成本、設置成本、產能), 建構一混合整數線性規劃, 藉由此模式的求解, 找出滿足公司利潤最大的目標下, 最佳訂單分派方案。而在模式的求解上, 由於此類訂單分派問題屬於非完全多項式求解 (non polynomial complete), 問題複雜度相當高, 因此作者採用模擬退火法 (Simulated Annealing ; SA) 的啟發式演算法求算近似解, 以克服模式求解費時及陷入局部最佳解的盲點。

Leachman[23]已為整個半導體製造業中生產規劃的問題提供了完整而詳盡的介紹。對於半導體業之大量生產、產品種類繁多及顧客層龐大的特性, 其利用線性規劃 (Linear Programming ; LP) 來將半導體製造的生產計畫模型化, 使企業獲得更大的競爭力。針對晶圓製造流程的產能利用, 其提出了產能負荷模組 (capacitated loading module), 根據各種產品的不同需求, 以線性規劃的方式, 來分配工作站機台的產能, 並決定各生產計畫期每種產品的投料量, 藉由這些不同產品組合之生產與銷售以達成利潤最佳化。因此運用此模組可求得使單一晶圓廠利潤最大的投料計畫, 但對一個多廠區的製造公司而言, 其無法在符合現行制度下解決各廠間產能調度的需求。

由於台灣液晶顯示器產業具有以下特殊性, 造成國外[21]、內研究尚未有對台灣液晶顯示器產業之特性提出以產銷平衡為考量基礎的訂單分配模式:

(1) 薄膜電晶體製造廠的列陣 (array) 製程類似半導體製程, 單元 (cell) 製造廠具有光電產業的特性, 模組 (module) 組裝廠與一般組裝產業相近, 前後段製程特性同如小型的上下游供應鏈體系 (如圖 1.1 所示), 各廠區的生產目標皆不相同, 必須同時滿足各廠區間的不同目標:

- a. 薄膜電晶體製造廠的目標: 因鍍模、曝光、顯影及蝕刻等四段製程有回流生產與共用機台的特性, 生產目標是讓各

設備對各產品的生產能平穩化，同時在滿足最終產品需求的前提下達到資源使用率的最大化。

- b. 單元製造廠的目標：考慮彩色濾光片、薄膜電晶體製造廠的供貨狀況以及滿足最終產品需求的前提下，提高各設備的資源使用率。
 - c. 模組組裝廠的目標：配置各等級的產品給顧客，儘可能地滿足顧客的需求。
 - d. 彩色濾光片製造廠的目標：因彩色濾光片的製造流程屬於單一生產線的生產方式，在滿足單元製造廠需求的前提下，儘量併批生產、減少換線次數，冀能提高資源使用率及良率。
- (2) 薄膜電晶體液晶顯示器產業的設備昂貴，而且針對需求客製化程度不高，屬於計劃式生產 (Make-To-Stock ; MTS)，但是因製程技術的不斷進步，良率及分級率都會隨著時間而改變，使得傳統物料需求規劃 (Materials Requirement Planning ; MRP) 的規劃邏輯無法滿足該產業之需求。
- (3) 為了達到各廠區的生產目標及考慮各製程的產品分級，在供給規劃階段，必須各自開立各製程的製令，最後再以存貨來滿足下游的需求。

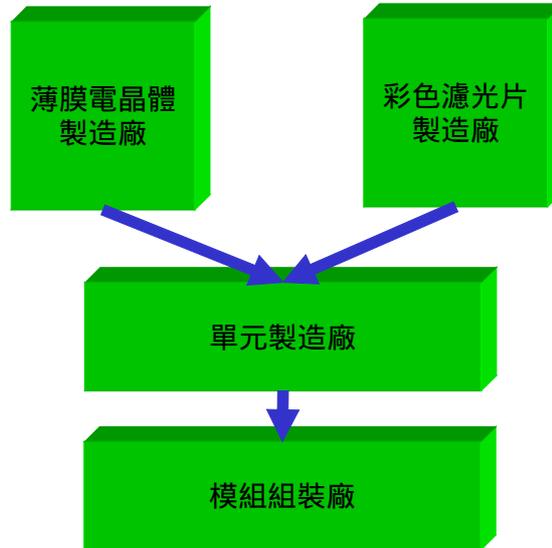


圖 1.1 薄膜電晶體液晶顯示器產業的製造流程概念圖

1.3. 研究方法與步驟

整體而言，本論文首先將探討供應鏈規劃及多廠區生產系統之相關文獻，並參考目前已有 i2[20]、Manugistics[25]和 Adexa[2]等商用先進規劃與排程 (Advanced Planning and Scheduling ; APS) 軟體及相關研究[43]。它們使用限制理論或混合整數線性規劃等技術協助企業進行多廠區生產規劃及物料規劃之功能。歸納出在多廠區生產環境下，同步考量企業的資源限制及成本最佳化的產銷平衡規劃邏輯，進而提出訂單分配的決策模式。為有效求解多廠區訂單分配的問題，本論文將運用基因演算法之最佳化搜尋的特性，同時考量限制條件滿足技術利用縮小搜尋解空間 (solution space)，進而能快速求得一可行解且兼具有最佳化搜尋能力的演算法。

最後，將應用本論文所發展的產銷平衡訂單分配模式於台灣液晶顯示器產業，以驗證並評估其效益。

1.4. 論文架構

本論文共分為五個階段進行，如圖 1.2 所示。

第一階段是文獻探討，本階段主要分為三部份：一為收集供應鏈規劃及多廠區生產規劃或排程之相關學術研究；二為探討目前商用軟體針對供應鏈規劃方面的相關作法，藉由學術及實務兩方面的研究，進一步設計出多廠區生產系統下訂單分配模式；三為探討限制條件滿足問題與基因演算法之相關學術研究，希望藉由基因演算法的特性來求解訂單分配模式。

第二階段是建立規劃邏輯，結合學術及實務的作法，整理出多廠區生產規劃系統之一般性規劃流程。藉由此核心流程進一步推演出下一階段的數學模式。

第三階段是建立數學模式，在考量企業的資源限制與規劃目標之下，決定訂單配置方式，以降低企業整體成本。以限制條件為基礎（constraint-based）的基因演算法做為本論文之核心引擎，解決供應鏈上產銷平衡的最佳化問題。

第四階段是實作模組，以第二階段所推演出的規劃邏輯構建目標函式與限制條件，及以限制條件為基礎的基因演算法做為求解工具，完成多廠區生產系統之產銷平衡訂單分配模組。

第五階段是模組評估，藉由探討台灣的產業特性訂定出績效衡量指標，藉以評估本模組。並著手進行模組的反覆修正。

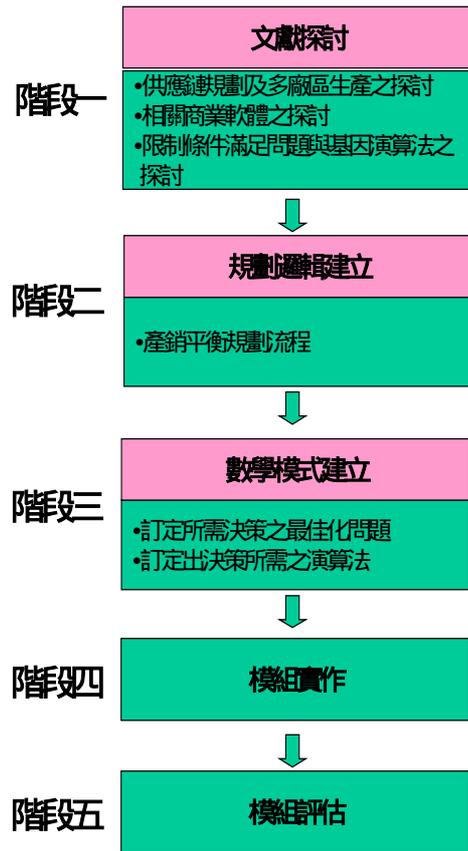


圖 1.2 研究方法與進行步驟

第二章 文獻探討

2.1. 多廠區生產系統之探討

本節分別對多廠區生產系統的特性、架構與規劃方式等三方面探討。因為多廠區生產系統的環境等同於供應鏈規劃 (Supply Chain Planning ; SCP) 的生產環境，所以本節也特別對於供應鏈規劃做進一步地探討。

2.1.1. 多廠區生產系統的特性

一般而言，生產規劃與控制的資訊包括訂單資訊 (顧客資料、顧客訂單、預測資料)，產品結構的資料，生產途程，採購零件和賣出物料的供應商資訊，現有存貨的詳細資料，在製品的資料，資源的資料等等。對於多廠區的生產規劃與控制系統，額外添加如供應鏈方面的資料，供應商與外包商的資料[42]。

Sabri 及 Beamon[34]定義供應鏈規劃是為控制存貨、採購及配銷的方法 (methods)、資源設施 (facilities)、供應商 (supplies)、顧客 (customers)、產品 (products)。為了分析供應鏈，該研究從供應鏈網路及其每層次的生產 / 配銷規劃進行分類討論[10]：

- 供應商 (supplier stage)
- 工廠 (plant stage)
 - 運輸網路 (transformation network)
 - 運輸 / 分配決策 (transformation / allocation decision)
 - 存貨
- 配銷 (distribution stage)
 - 配銷網路 (distribution network)
 - 位置 / 分配決策 (location/allocation decision)
 - 存貨

Thierry 等[38]認為多廠區的生產系統中，產品的製造流程是需要

使用不同廠區內的資源。換句話說，單一產品的生產是具有替代性的，比如相同企業下的二個廠區，對某個生產流程可能具有相同製程能力或是具有互補能力。

解決多廠區生產規劃的問題，是藉由改善相同企業下不同生產單位間生產規劃與控制的合作關係。因此，這樣的問題主要是要決定不同生產單位需要生產的品項數量與各個生產單位間的運輸數量。它具有以下的基本資料：

- 相同的企業下有不同的生產單位及其產能效益；
- 用於生產零件的物料清單；
- 生產不同組件與最終料件需要的資源數量；
- 生產單位間的運輸時間。

及限制條件：

- 產能限制；
- 存貨與生產及運輸的平衡式；
- 生產單位間生產與需求的限制；
- 需求與運輸的限制。

Sauer 等[36]則認為多廠區生產的訂單分配問題，可視為不同工廠間的生產流程分派問題，其主要特性分為：

- (1) 不同工廠之間，其生產流程的相互依賴性是很複雜的。
 - a. A 工廠所製造的產品是 B 工廠所需要的。
 - b. 相同的品項可能被不同工廠所製造（可能造成不同的成本）。
 - c. 不同工廠之間的零件運輸問題。
- (2) 全域排程需要的資訊類型是以一般性資料來取代精確資料，例如：
 - a. 以機器群組來替代單一機器的產能資訊。
 - b. 中間或最終產品所估計的製造時間。
- (3) 單一工廠之既存排程系統的整合。
- (4) 企業下的所有工廠，其分散的排程活動間合作之必須性。
- (5) 單一工廠的實際狀態之不確定性。

所以，多廠區生產系統的目標就是減少分散式排程之複雜度，且同時改善排程的品質。

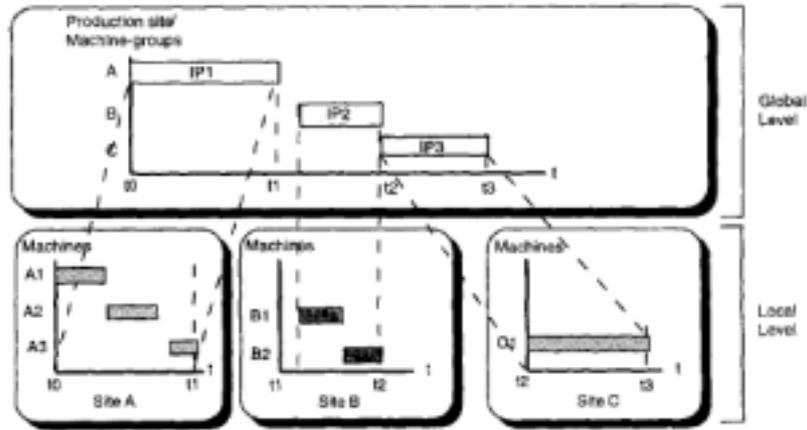


圖 2.1 全域與區域的排程[36]

此外，Sauer 及 Appelrath[35]將多廠區生產系統的排程問題分為全域（global）排程、區域（local）排程與運輸排程（如表 2.1）。全域層次的主要任務是產生每個位置（location）中需要被製造的中間產品（intermediate product）之需求量（如圖 2.1 所示）及不影響其他廠區的同時，提供足夠的彈性使得區域排程（指的是單一位置）能夠反應區域性的干擾事件。它可以藉由加入緩衝時間的啟發式法則及模糊技術來最佳化機器群組的平均負荷。

表 2.1 全域、區域與運輸排程的建模內容

	區域排程	運輸排程	全域排程
資源	機器	運輸車輛	機器群組
產品	由數個作業所生產的中間產品	使用特定運輸車輛的中間產品	由數個中間產品所組成的最終產品

訂單	中間產品的內部 訂單	中間產品的內部 訂單	最終產品的外部 訂單
硬性限制條件	根據生產需求(順序限制), 排定所有 有訂單	根據技術需求(車輛種類、運輸產 能), 排定所有訂 單	根據生產需求(順序、產能限制), 排定所有訂單
軟性限制條件	最佳化機器使用 率, 準時交貨, 最 小化在製品成本	準時交貨 (meet due dates), 最佳 化車輛使用率, 最 小化成本	準時交貨, 最小化 運輸時間 / 成 本, 平衡生產負 荷, 減少存貨成本

全域排程根據規劃特性又可以細分為預知式 (proactive) 與反應式 (reactive), 其優缺點整理如表 2.2 :

表 2.2 預知式與反應式排程的優缺點比較表

	全域的預知式排程	全域的反應式排程
優點	<ul style="list-style-type: none"> • 整體最佳化, 做多目標決策分析。 • 將區域性排程所發生的干擾視為限制條件, 一次做好長期規劃。 	<ul style="list-style-type: none"> • 有效地運用區域性排程, 做較精確的目標規劃。 • 當區域性排程發生干擾, 只需做部份規劃, 有利於即時回應資訊。
缺點	<ul style="list-style-type: none"> • 無法正確反應區域性排程所發生的干擾, 可能積壓多餘的產能或庫存。 • 當區域性排程發生干擾, 需要重新做整體最佳化。(牽一髮則動全身) 	<ul style="list-style-type: none"> • 一旦多個區域性排程發生變異而其相依性太高時, 規劃執行時間將會很長, 或需要很複雜的演算法來規劃。

此外，陳氏[47]指出當市場需求不能被延宕，而供應商又面臨供給不足的窘境時，供應商必須抉擇對該訂單接受與否。若不接受該訂單，則僅面臨的是損失短缺成本的情境，對正常的產能及其他訂單並無影響。但若供應商選擇接受該訂單，則供應商必須面臨產能調整或重新分配現有之訂單等現實狀況。產能調整實屬不易，而重新分配訂單卻是可行的解決之道。事實上，業界面臨的難題也多半是如何重新分配訂單使得供給之間均得最大利益。

為了協調廠區間的運輸，多廠區生產規劃通常處理一個可共享的庫存。因此，企業藉由數個具有互補或替代生產能力的廠區，去規劃每個廠區間的互動合作。綜合上述，在多廠區生產系統的環境中，產品的生產流程是需要使用跨廠區間的資源，亦即產品的生產流程是存在替代性。解決多廠區生產的問題，是要改善不同廠區之間在生產規劃與控制層面的合作問題。

因此，本論文預定要解決多廠區生產規劃的問題可以設定為決定『規劃週期中，單一廠區內需要生產及不同廠區間需要運輸的品項數量。』

2.1.2. 多廠區生產系統的架構

Sauer 等[36]將多廠區生產系統的規劃問題稱為 MUST (MULTI-SiTe) 問題，而提出解決此問題的一般性架構如圖 2.2 所示。

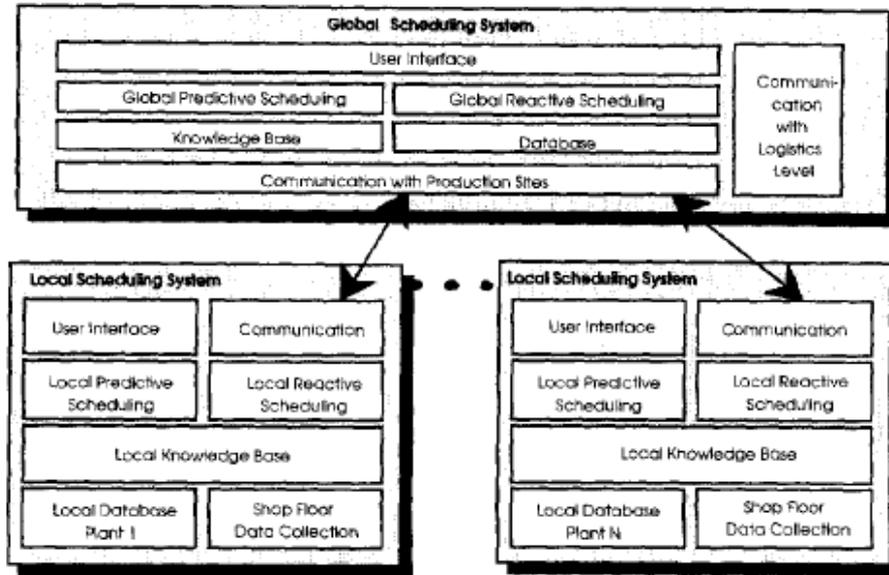


圖 2.2 多廠區生產系統的架構[36]

Thierry 等[38]也提出由中央規劃控制系統（centralized multi-site planning and control）統籌進行企業的整體規劃後，再將規劃結果下放至各種生產單位執行，如圖 2.3 所示。

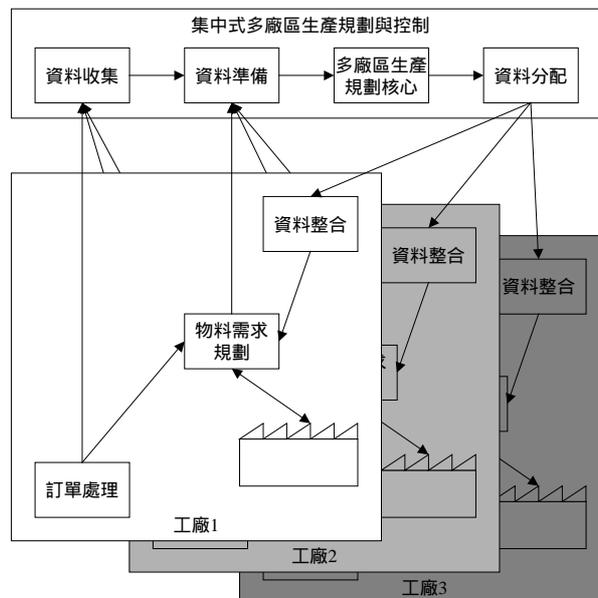


圖 2.3 多廠區生產規劃與控制系統

此外，郭氏[44]基於平衡負荷的理念，以關鍵資源為監控重點，提出晶圓製造廠多廠間的訂單抵換機制，主要分為三個模組：

(1) 產能推估模組—

這個模組中，首先透過各廠已知的生產規劃資訊及顧客換單資訊，逐一檢測插入訂單於各廠的達交可行性。接著基於平衡關鍵資源產能供需面的理念，針對可以滿足插 / 換單交期之各廠，推估、比較已規劃工單與插入工單對系統關鍵資源的產能供需狀況，以搜尋出初步的候選工單。

(2) 訂單分派模組—

獲取初步候選工單後，在訂單分派模組口中，首先透過篩選階段，分析候選工單與插入工單在作業策略面及產能面的異同，以進行候選工單的篩選，提升抵換效率。其次經由排序階段，依據通過篩選機制的候選工單所能提供之瓶頸資源產能量、投料時點先後等準則，進行一合理的候選工單抵換優先權排序，以協助挑選出最適抵換方案，並分派插入工單至生產系統。

(3) 抵換評估模組—

承續訂單分派模組搜尋出最適抵換方案，首先因應各廠訂單異動的結果進行重排程。其次比較訂單抵換前後，系統各項績效指標的變動情形，如關鍵機種利用率、系統彈性、工單延誤資訊等，以確保系統變動狀況為一可接受的範圍。另外，亦可透過此相關資訊的提供，協助接單人員決策是否允諾顧客的換單要求或做為其索價的依據。

因此，中央規劃的多廠區生產規劃系統之架構，一般以傳遞「訂單資訊」的方式做為單廠區生產排程系統的溝通協調機制。大致而言，「訂單資訊」除了包含基本的「顧客資料」及「產品資料」外，最重要的資訊是多廠區生產系統所彙總的需求，經由規劃後所分配給各個廠區的訂單「數量」及其「交期」資訊。換句話說，多廠區生產規劃系統的功能模組主要考量各種生產因子及限制，再決定訂單數量

及其交期，進而達成多廠區生產系統的生產目標。第 3.3 節就是根據上述的文獻探討，進一步提出一般性的多廠區生產規劃系統之架構。

2.1.3. 多廠區生產規劃的方式

多廠區訂單分配問題，一般多為批量決策問題 (lot sizing problem)，其包含訂單的切割 (產品種類、數量)、生產、運送等行為。Timpe 及 Kallrath[39]以一個具有數個生產工廠與銷售點的化學公司為研究對象，以各項生產因子 (如生產數量、工廠產能、物料流量、存貨、運送等) 為限制條件，提出一混合整數線性規劃 (Mixed-Integer Linear Programming ; MILP)，期望在公司的目標下，求得各個工廠最適宜的生產數量。然而，該研究利用混合整數線性規劃構建模式，會產生求解費時的疑慮，而每個規劃週期只能進行一次生產轉換 (order change)，也是這個方法的一大限制。

蔡氏[43]也以作業研究中的混合整數線性規劃為基礎，來解決多廠生產下的資源分配與生產規劃，並透過數學之運算，尋求以成本極小化為目標函式的最佳解答；該模式主要是應用在生產運籌管理中的「主排程規劃」(Master Planning ; MP) 與「工廠規劃」(Factory Planning ; FP) 這兩個層次，在模式的設計上，以電子組裝廠商的生產程序、及其在生產排程上所卻解決之問題，做為模式規劃的基礎。在模式中，同步考量生產成本、物料採購成本、物料存貨成本與工廠開工成本等總成本最小化，及因應實狀況的限制條件，如下：

- 需求量限制式；
- 產能限制式；
- 物料平衡限制式；
- 物料存貨限制式。

與可供彈性選擇的限制條件，如下：

- 物料採購限制式；
- 生產線開工限制式。

謝氏[46]同樣指出過去解決供應鏈問題，最常用的方法為混合整數線性規劃模式，若使用線性規劃模型，因變數相當複雜、限制式過多，常有無解的情形。該研究提出另一種解決問題模式，以網路流量管理的啟發式（Heuristic）演算法，解決供應鏈的廠商指派與訂單規劃排程問題。

該研究的方法主要分為四大步驟，首先將具有不同生產程序之節點分離，執行產能初始化，第二步驟則是將所有產能轉換為以最終產品為產能單位。第三步驟，根據使用者需求選擇適合訂單排序方式與參數，訂單排序後便可執行該演算法核心步驟，每筆訂單依序規劃排程。

在對每一張訂單規劃排程時，找出最小成本之廠商組合，然後尋找與安排適當生產量，當此廠商組合無法滿足需求時，調整供應鏈網路圖形以尋找次佳之廠商組合，不斷重複上述步驟直到訂單需求滿足或供應鏈已無任何產能幫助生產為止。當交貨時距限制下需求仍未滿足，延遲一個交貨時距，重複上述之方法，直到訂單需求完全滿足為止。

Sauer 及 Appelrath[36]採用階層式的規劃方式外，同時將模糊理論的概念（fuzzy concept）及啟發式演算法應用於排程問題當中，期望藉由加入模糊理論的概念，使生產管理人員在利用不明確的資料進行整體排程規劃（global level scheduling）時，同時能兼顧到各個工廠間（local level scheduling）排程計畫的穩健性與協調性。

該研究處理模式是結合啟發式的排程策略、問題的分解及模糊資料和法則的處理。訂單為主的問題分解過程如下：系統粗略評估排定的訂單；看起來“較困難完成”的訂單優先被製造，以使得目標可被達成。因此，訂單會被特定的評估計劃所分類。這使得中間產品的訂單能適時地被分派至適當的機器群組。這些法則主要應用於最佳化平衡工作負荷（load balancing），或避免瓶頸資源（avoid bottleneck resources）。其求解目標為：最佳化機器使用率（optimizing machine utilization.）與所有訂單準時達交（meeting all due dates.）

而模糊資料和法則的處理可分為以下三大步驟：

- (1) 排序 (sequence)：根據法則來區分訂單的重要性；
- (2) 排程 (schedule)：根據由法則 (考量軟、硬性限制條件) 所決定訂單次序中的位置，而每個訂單會評價所有可能被分派的機器群組。位置評價愈大表示愈適合的機器群組將被排定；
- (3) 計算物料需求 (material demand)：彙總所有產品的資料來計算物料需求。

此外，在 Sauer[35]等的後續研究中，也將產品的運輸問題整合於排程規劃過程中，其方法在於藉由專家系統、知識庫的導入，將運輸排程問題轉換為生產排程問題進行求解，圖 2.4 為其求解過程。

```
BEGIN
WHILE transportation orders to schedule
  select order
  select possible transportation vehicle
  select time interval
  IF possible THEN schedule it
  ELSE solve_conflict.
END WHILE
optimize schedule
END
```

圖 2.4 訂單為主的運輸排程之啟發式法則

Thierry 等[38]將生產過程中順序相依的機台整置時間 (setup time) 問題加入排程的考量。該研究所提出的方法，是藉由時距 (time bucket) 大小的設定，決定各時距內的生產單位，以處理機台整置問題。而這個方法，從另一個角度來看，其實是類似於批量決策問題 (lot sizing)。此外，在決定時距時，採用解決限制滿足問題 (Constraint Satisfaction Problem; CSP) 的方法，考量了系統狀態平衡式、產能、交期與機台設置等限制式，來縮小可行解的空間，以增進求解效率。但是，由於該研究的研究對象為汽車組裝業，為一流線型生產 (flow

shop) 形式，因此可以利用此法來解決機台整置問題。

Dasci 及 Verter[9]提出供應鏈的網路結構可以定義為生產-配銷系統，以此為基礎提出結合離散與連續型的數學模式，運用離散型混合整數線性規劃來解決整個生產-配銷網路上廠區間的替代性問題。也就是在滿足顧客需求的條件下，決定由那一個廠生產及是否增設新廠？其目標是考量設新廠的固定成本、產能消耗和運輸成本等總成本最小化，其中運輸成本主要包含工廠位置間的距離、車輛運輸量的大小及送貨排程等因素。

在處理多廠訂單分配問題上，曾氏[55]也進行了此方面的研究，其依據客戶訂單資訊（產品別、數量、交期）與各工廠生產特性資訊（製造成本、設置成本、產能），構建一混合整數規劃模式，藉由此模式的求解，找出滿足公司利潤最大的目標下，最佳訂單分派方案。而在模式的求解上，由於此類訂單分派問題屬於非完全多項式求解（non polynomial complete），問題複雜度相當高，因此作者採用模擬退火法（Simulation Annealing；SA）的啟發式法則求算近似解，以克服模式求解費時及陷入局部最佳解的盲點。不過，這也代表在求解過程中，各項參數的設定，攸關最終解的品質。

Sue[37]提出一供應鏈上的配銷網路，應用於美國聯合包裹運輸公司（United Parcel Service of America；UPS）、美國運通（American Express）等流通業。應用基因演算法及其結合禁忌搜尋法（tabu search）的演算法來解決流通業上的運輸問題，其目標函式是主要考量總運輸成本最小化及以下的限制條件：

- 轉運中心（hub）與運輸通路（spoke）的限制式；
- 轉運中心之間的限制式。

Leachman[23]為整個半導體製造業中生產規劃的問題提供了完整而詳盡的介紹。對於半導體業之大量生產、產品種類繁多及顧客層龐大的特性，其利用線性規劃（Linear Programming；LP）來將半導體製造的生產計畫模型化，使企業獲得更大的競爭力。針對晶圓製造流程的產能利用，其提出了產能負荷模組（capacitated loading

module)，根據各種產品的不同需求，以線性規劃的方式，來分配工作站機台的產能，並決定各生產計畫期每種產品的投料量，藉由這些不同產品組合之生產與銷售以達成利潤最佳化。因此運用此模組可求得使單一晶圓廠利潤最大的投料計畫，但對一個多廠區的製造公司而言，其無法在符合現行制度下解決各廠間產能調度的需求。

陳氏[45]採用 Leachman 所提出的線性規劃生產計劃模型為工具，以求解各單一晶圓廠在其產品需求與產能限制下之利潤最大的生產計劃。而在產能調度部份，以一次調整一對可以相互支援的資源項目組合為原則，逐步進行產能支援作業。修訂選取每次進行產能調度的支援標的之原則，以提高獲得最佳產能支援方案的機率。或更進一步，研究建立可獲得最佳產能支援方案的產能支援方法。

該研究方法中主要包括求解二個決策的方法。首先為求解在特定供需條件下的生產計劃，再者為尋找與調度各晶圓廠間對應的產能剩餘與不足資源項目的產能支援方案。所要調度及充分利用的資源即為各晶圓廠的工作站產能。

Pirkul 及 Jayaraman[32]認為配銷系統一般會發生的情況：「藉由一些生產工廠供給 (supply) 倉儲中心多種產品，來轉換不同產品的顧客需求量。」該研究根據有限產能之工廠與倉儲中心所形成的供應鏈管理，提出一個混合整數線性規劃的模式，並以啟發式法則來解決此模式。

此模式的目標函式是最小化工廠及倉儲中心的固定成本，加上從工廠運輸產品至倉儲中心與倉儲中心運輸產品至顧客的變動成本。

其考量的限制條件包含：

- 需求滿足的限制式；
- 倉儲中心的運輸量限制；
- 從工廠運輸至倉儲中心的可供給量與所有顧客需求之平衡；
- 工廠產能的限制。

Guinet[15]指出數個生產單位稱為多廠區。在最低的成本下，以供給來滿足不確定需求。這些成本因素包括廠區內資源（處理成本、

持有成本、整置成本), 顧客方面(運輸成本、延遲成本), 控制生產系統所帶來的管理成本(製程能力, 強韌性, 彈性等等)。

該研究提出二階段的生產管理方法來控制多廠區生產系統。它除了最佳化上述成本外, 更考慮資源產能的限制條件。

多廠區生產系統的目標是回答二個相關的問題:「誰要生產?」與「何時生產及生產那一個零件?」

其考量的限制條件包含:

- 每個需求(特定週期內特定產品)都被滿足;
- 每個週期, 每個廠區內的有限處理及儲存產能;
- 每個需求至少被分配給一個(特定週期內特定廠區的)資源。
- 每個廠區在某週期內至多滿足 N 個需求量;
- 運輸上的分配問題;
- 考量整置時間;
- 物料清單(Bill of Material; BOM)上的產品結構關係。

該研究的多廠區生產規劃之啟發式法則是依據每個週期及每個廠區來規劃產品的處理。首先, 由各廠區執行生產規劃。第二步, 將每個短期規劃問題各別交給多工作站的啟發式排程法來執行, 以便最小化訂單最大延遲時間。為了同步化廠區間的排程, 選擇以提升瓶頸運作(shifting bottleneck)的方法來規劃。第三步, 依照各廠區的生產力排序, 以「供應商-顧客」的關係來連結整個排程問題。也就是, 以訂單交期與開立時間來遞推排程。

Ross[33]提出主要以資源分配為主的績效規劃(performance planning)來發展以利潤最大化為目標的供應網路模式, 且應用於一上下游供應網路的石油產業, 並以線性規劃軟體 - LINGO 來解此模式。目標函式考量預測及顧客訂單的利潤, 配銷中心的固定作業成本, 配銷中心的運輸車輛分派之變動作業成本, 向供應商採購的成本(包括單位換算與預測成本), 地區間的平均配送成本(包括直接人工、油料與運輸的成本)。限制條件如下;

- 顧客需求與運輸車輛分派的對應式;

- 運輸車輛的總產能不得超過配銷中心的儲存產能；
- 運輸車輛的配送量上下限；
- 每輛車輛的累積運輸公里數之上限限制；
- 每輛車輛的地區性最低配送量限制。

Sabri 及 Beamon[34]發展一整合性多目標的供應鏈模式，同步規劃供應鏈上的策略及作業面。多目標的決策過程中，分析包含成本、顧客服務水準（滿足率）及彈性（達交量）等的衡量指標。進一步地，此模式藉由分解生產、配銷及不確定性需求等問題，為整個供應鏈網路提供一個多目標的衡量指標。策略面最佳化模式的主要目標是決定更有成本效益的資源設施（工廠與配銷中心）的位置，供應鏈上的產品流（flow of goods）與由配銷中心至顧客間的指派工作；作業面最佳化模式的主要目標是決定每個儲存單位的安全存貨，產品的補貨與裝配之批量大小及頻率，運輸與生產的前置時間，及顧客服務水準。

策略面模式分為四個子層次：(1) 供應商 (supplier / vendor)；(2) 生產工廠；(3) 配銷中心；(4) 顧客。第一個目標函式是最小化固定與變動成本，包括

- (1) 原物料採購成本和供應商至工廠間的運輸成本；
- (2) 工廠作業上的固定與變動成本；
- (3) 配銷中心的存貨與持有變動成本和工廠至配銷中心間的運輸成本；
- (4) 配銷中心至顧客的運輸成本。

第二個目標函式表示工廠與配銷中心的彈性。

其限制條件如下：

- 原物料的供給限制；
- 生產工廠的產能限制；
- 供給與需求的關係式。

作業面模式是根據策略面模式的輸出（決策變數），及顧客需求、最小服務及彈性水準、成本與前置時間、物料清單及不確定的變動成本等基本資料，來決定存貨變數（批量大小、再訂購點及安全存貨）

的最佳化。它分為三個子層次：(1) 供應商；(2) 工廠；(3) 配銷中心。

供應商層次是控制原物料的採購成本，包括持有存貨水準（平均存貨水準加上安全存貨）、再訂購點及變動前置時間等限制考量。工廠層次中再細分為：生產模式是最小化整置成本、處理成本及在製品持有成本等總生產成本；最終產品儲存模式是考量最終產品的儲存持有成本、由工廠送至配銷中心的運輸持有成本及整置成本。

配銷中心層次是控制總配銷成本，包括產品持有成本、再訂購成本與延遲訂單的成本。

不同的供應鏈模型的確會採用不同的成本細目，表 2.3 列舉相關領域之學者所採用哪些成本項目為考量因素。

表 2.3 供應鏈模型的成本項目之相關研究

	訂單 成本	運輸 成本	銷售量	短缺 成本	持有 成本	原料 成本
Banerjee[3]	*			*	*	*
Gurnani[16]				*	*	*
Abad[1]	*	*	*		*	*
Metters[26]				*	*	*
Chiang[7]	*		*	*		*

綜合上述，不同產業的多廠區生產特性各不相同，其考量的限制條件及目標函式也不盡相同，但是對於生產因子而言，仍可大致分類如表 2.4 所示。

表 2.4 不同產業的多廠區生產規劃比較表

產業別		流通業	電子組裝業	汽車組裝業	半導體 製造業	石化業
目標 函式	最大化	x	x	x	• 利潤(產品 組合及銷售)	• 利潤(預測 及顧客訂 單)
	最小化	• 設廠成本 • 生產成本 • 運輸成本	x	• 生產成本	• 生產成本 • 存貨成本	• 生產成本 • 採購成本 • 存貨成本 • 運輸成本
限制 條件	產能 限制	• 運輸距離 • 送貨產能 • 運輸通路	• 工廠產能	• 工廠產能 • 機台設置	• 工廠產能 • 工作站產 能	• 工廠產能 • 車輛產能
	物料 限制	• 工廠產能	• 物料平衡 • 物料存貨		• 物料存貨	• 物料存貨 • 物料運送
	供給 與 需求 平衡	• 滿足需求 • 供需平衡	x	• 狀態平衡	• 供需平衡 • 需求分級	• 運輸需求
	數量 限制	• 運輸數量	• 需求數量	x	• 分量限制 • 總量限制	• 生產數量 • 物料流量 • 配送數量
	時間 限制	x	x	• 訂單交期	x	x

<p style="text-align: center;">規 劃 方 法</p>	<ul style="list-style-type: none"> ● 混合整數 線性規劃 ● 基因演算 法結合禁 忌搜尋法 ● 啟發式法 則 	<ul style="list-style-type: none"> ● 混合整數 線性規劃 	<ul style="list-style-type: none"> ● 限制條件 滿足技術 	<ul style="list-style-type: none"> ● 線性規劃 ● 啟發式法 則 	<ul style="list-style-type: none"> ● 混合整數 線性規劃
--	---	---	---	--	---

其他多廠區生產規劃方法尚有模擬退火法、模擬理論等技術，而多廠區生產規劃方法可以大致二類：(1) 以訂單分配為主；(2) 以資源分配為主的規劃方法。接著，第 2.2 節將介紹一般商用軟體的多廠區生產規劃方法及所考量的限制條件。

2.2. 商用軟體的作法

本小節將以二套目前最廣受歡迎的先進規劃與排程軟體來探討一般商用軟體解決供應鏈規劃（或稱多廠區生產規劃）的方法。

2.2.1. Manugistics CBMP

CBMP 是 Manugistics 的一個模組，其功能主要是產生滿足需求的生產計劃。它同步考量物料供給（material availability）、配送產能（distribution capacity）及生產產能（production capacity）等限制條件。它可以根據公司的企業策略（business strategy）設定限制條件，例如顧客訂單中計算安全存貨及消耗預測訂單的法則。主要考量的限制條件如下：

- 物料供給（material availability）；
- 生產產能（production capacity）；
- 進料存貨能力（in handling capacity）；
- 出貨存貨能力（out handling capacity）；

- 運輸通路產能 (lane capacity)

CBMP 也包含即時可允諾交貨數量查詢 (Real-Time Available-To-Promise ; RTATP+) 的功能 , 其考量關鍵資源及物料的供給能力來決定品項 (item) 的達交數量。其資料結構如圖 2.5 所示。

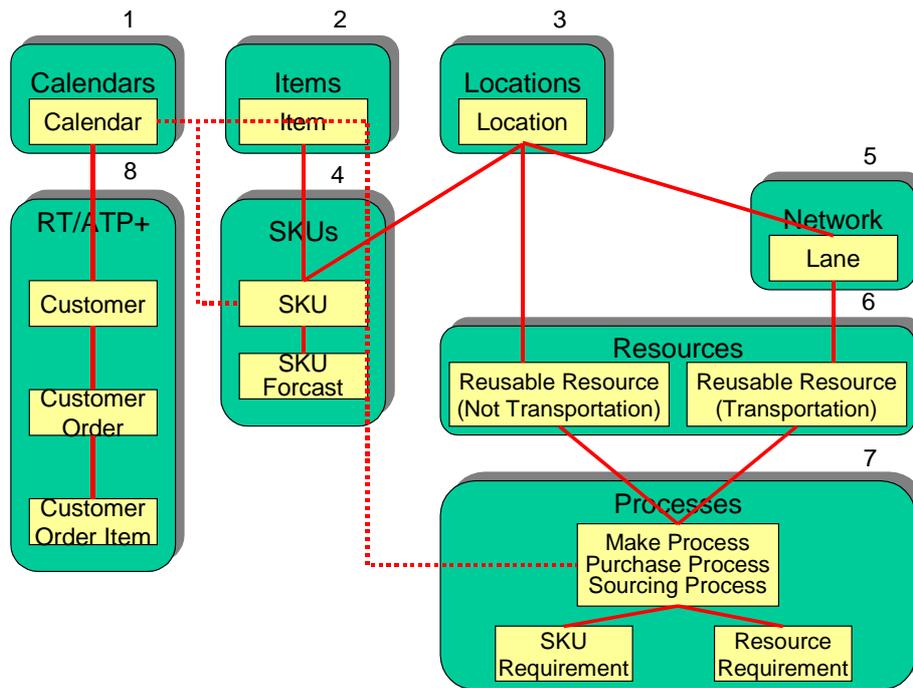


圖 2.5 Manugistics CBMP 的資料結構之概念圖[25]

CBMP 藉由設定優先度模式 (priority scheme) 來決定那個訂單能最先被滿足。其優先度模式主要考量以下因素的權重：

- 收益 (revenue) ;
- 邊際利潤 (margin) ;
- 品項 (item) ;
- 顧客 (customer) ;
- 安全存貨 (safety stock) ;
- 預測 (forecast) ;
- 交期 (need date) 。

而決定訂單的排序方式如下：

- 現有品項 (item) 需求中，優先度較高較優先處理；
- 新的品項需求能以現有多筆品項需求同時滿足；
- 根據顧客與品項之優先度的排序結果來重新規劃 (re-plan)；
- 如果優先度較高的訂單在重新規劃的過程中仍然延遲 (late) 或未被滿足 (unmet)，則系統會發出例外訊息；；

滿足需求的方式，首先以現有的存貨來滿足其儲存單位的需求。如果無法滿足的話，它將會指定製造 (make)，外包 (sourcing) 及採購 (purchase) 流程來要求生產 (producing) 運輸 (transporting) 及購買 (buying) 該儲存單位。CBMP 會優先考量優先度最高的流程，當流程能完全或部份滿足需求訂單 (顧客訂單、儲存單位的預測訂單及安全存貨單) 時，它將根據流程來建立供給訂單 (製造訂單、外包訂單或採購訂單)。它會一直規劃至需求被滿足，或是該儲存單位的流程都被指定過了。在建立供給訂單的同時，系統會要求設定物料的供給狀況及生產和運輸的產能狀況之限制條件。它也能設定預測調整及安全存貨計算的法則。

2.2.2. Adexa SCP

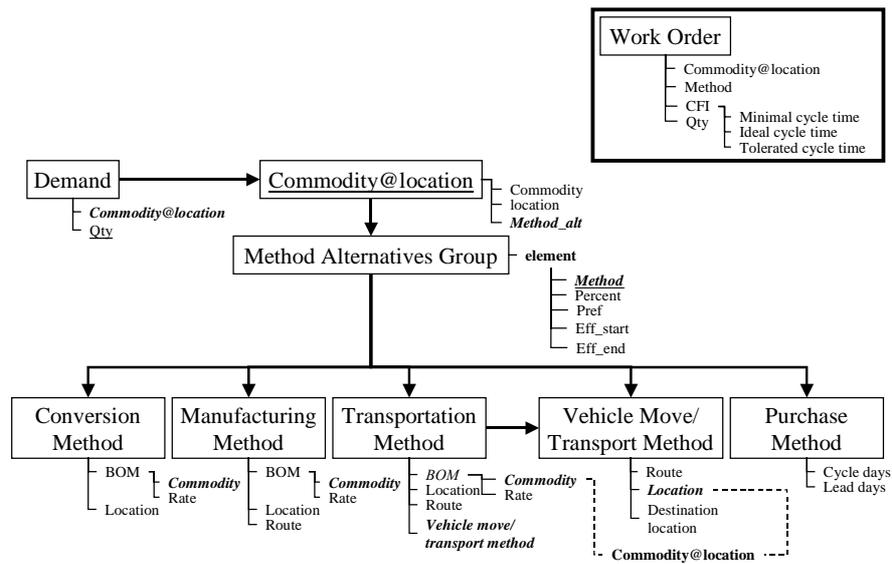


圖 2.6 Adexa SCP 的資料結構之概念圖

Adexa SCP[2]的規劃方式分為兩大部分：(1) 產銷平衡 (balancing) 規劃；(2) 供應網路排程 (scheduling)。

(1) 產銷平衡規劃方式可分兩種：

- a. 線性規劃 - 其中包含了 Inner , CPLEX , XPRESS , External 四種法則可供選擇。
- b. 啟發式法則 (heuristic algorithm) - 根據優先度 (preference) 及外包比率 (sourcing rate) , 並考慮等級需求 (grading requirement) 及工單大小限制 (work order sizing constraint) 做規劃。

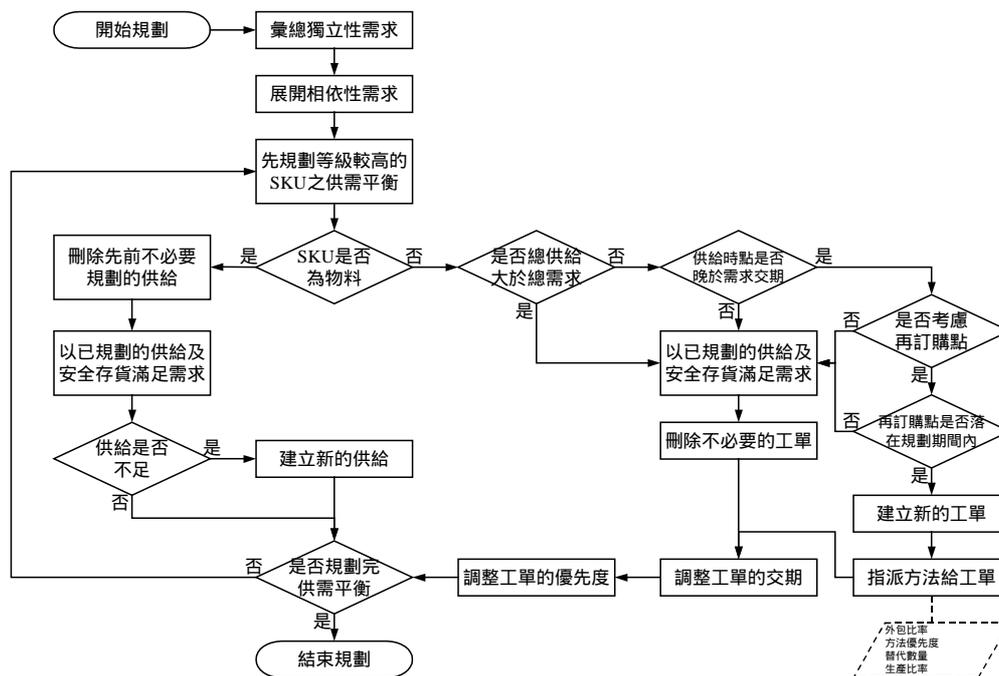


圖 2.7 Adexa SCP 的產銷平衡規劃方式
- 啟發式法則

圖 2.7 的規劃邏輯主要是針對每個儲存單位 (Stock Keeping Unit ; SKU) 以現有的供給存貨來滿足需求。若是供給過多，則是刪除多餘的工單 (work order) ; 若是供給不足，則是建立新的工單。依圖 2.6 的資料結構，此一階段同時決定每個工單的生產方法(method)

(2) 供應網路排程方式

圖 2.8 是根據產銷平衡規劃方式所規劃的工單需求，以產品生產週期 (cycle time) 為目標來決定各個工單的開工及完工時間。結合 Adexa SCP 的規劃方式是以供應鏈網路 (supply chain network) 中關鍵要徑 (critical path) 上的工單先行規劃，再規劃非關鍵要徑上的工單，若是在非關鍵要徑上的工單無法決定其合理的開工及完工時間，則將此工單列入關鍵要徑上的工單，再重新規劃。決定各個工單的開工及完工時間之規劃邏輯，以前推 (forward) 方式由上游的工單以

最早開工時間為基準，規劃至下游的工單。若無法得到合理的解，則再以後推 (backward) 方式由下游的工單以交期為基準，規劃至下游的工單。若無法再行得到合理解，則以前推 (forward) 方式規劃至有合理解為止。

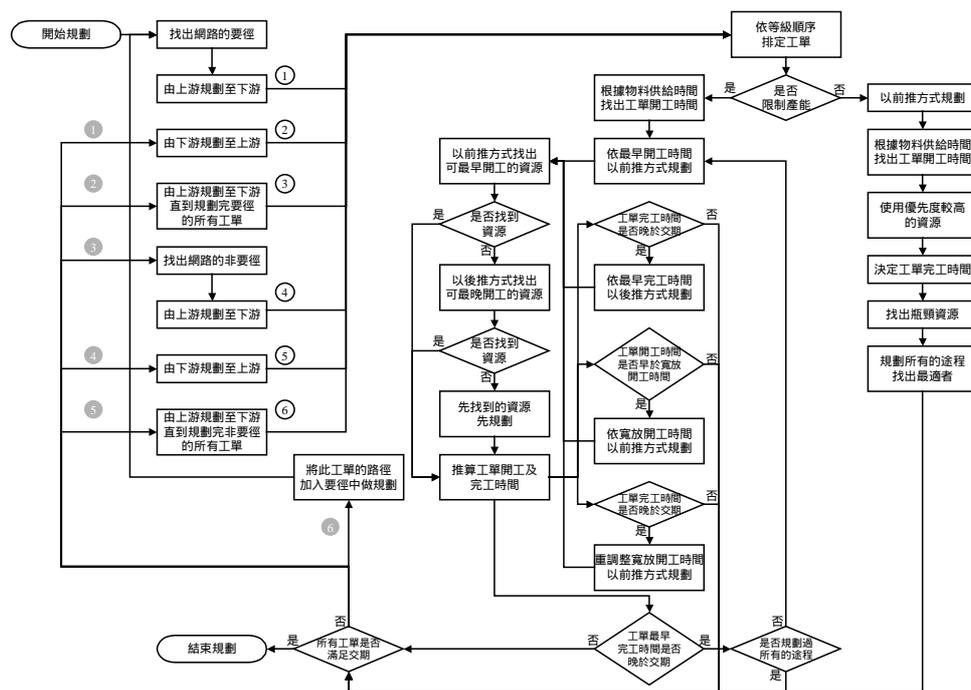


圖 2.8 Adexa SCP 之網路排程的排程邏輯

Adexa SCP 的產銷平衡規劃方式是在考量供應鏈上的成本最小化之前提下，決定每個儲存單位 (commodity@location) 的生產方式 (method) 之外包/生產比率 (sourcing rate/percent)。決定外包/生產的比率所考量的成本如下：

- 違反安全存貨的成本是每單位低於存貨水準的成本。
- 持有成本是每單位可存貨空間所造成的成本。
- 存貨成本是位置上所存貨的成本。
- 單位成本是物料被生產的成本。

- 低於使用率的成本是資源沒有滿載之情況下所造成的處罰成本。
- 收入是賣給顧客的品項價格。
- 延遲成本是顧客訂單交期延遲所造成收入損失的成本。
- 卸貨成本是將物料送至需要的位置的總成本。

2.3. 限制條件滿足問題與基因演算法之探討

2.3.1. 限制條件滿足問題

2.3.1.1. 限制條件滿足問題之簡介

在 Tsang[40]中所提出的定義：「基本上，一個限制條件滿足問題是由一組有限個數的變數所組成，每一個變數都伴隨著一個有限範圍的值域以及一組限制條件，這些限制條件限制著變數可以同時被指定的值（value）。因此限制條件滿足的主要工作，就是為每一變數指定一個合適的值，使其符合所有與其相關的限制條件」。所以，只要問題滿足以上的敘述，就稱此問題為限制條件滿足問題。

數學模式（如作業研究）與圖形表示（如網路）都是常用限制條件滿足問題表達的方式，選擇哪一種表示法端看所遇到問題之特性來決定。在此採用一個較一般的方式來表示一個限制條件滿足問題：

$$CSP (Z,D,C)$$

其中

Z ：為一組有限個數的變數 $\{x_1, x_2, \dots, x_n\}$ ；

D ：為一個有限範圍的值域 $\{D_1, D_2, \dots, D_n\}$ ， D_k 表示第 k 個變數 x_k 的值域；

C ：為是一組任意變數的組合，代表的是變數與變數間或變數本身的限制條件。如 C_{x_1, x_2, x_3} 代表的是 x_1, x_2, x_3 之間的限制條件。假設 x 限制於 a, b, c 三個值那麼 C_x 可以記為

$C_x = \{\langle x, a \rangle, \langle x, b \rangle, \langle x, c \rangle\}$ 。其中 $\langle x, a \rangle$ 表示 x 可以設定為 a ，

這樣一個變數與一個值的組合，稱為一個複合符號
(compound label)

而限制條件滿足的主要工作，就是必須找到一組解：

$\{ \langle x_1, v_1 \rangle, \langle x_2, v_2 \rangle, \dots, \langle x_n, v_n \rangle \}$ ，使其符合所有的限制條件 C ，其中
 V ：為指定給變數的值 $\{v_1, v_2, \dots, v_n\}$ ， v_i 為第 i 個變數 x_i 的指定值。

2.3.1.2. 限制條件滿足技術

用來輔助或進行求解限制條件滿足問題的演算法，稱為限制條件滿足技術 (Constraint Satisfaction Technique ; CST)，依各演算法的特性，可區分為下列六類：

(1) 搜尋演算法 (searching algorithms)

限制條件滿足問題中一個簡單的搜尋方法可分為三個選擇點：

(a) 尋找一變數，(b) 指派一個滿足限制條件的值，(c) 變數中所有的值皆不能滿足任何限制條件，則回溯 (backtrack) 至先前所進行搜尋的變數，指派尚未指派的值繼續搜尋。例如：chronological backtracking、backjumping、benchmarking 與 forward checking 等演算法。

(2) 排序演算法 (ordering algorithms)

可分為變數指派排序 (variable ordering) 和值域指派排序 (value ordering)。所謂變數指派排序為進行求解的過程中，尋找變數的順序。值域指派則是尋找到一變數後，針對此變數的值域，找尋適當的指派值，而指派值的順序稱為值域指派排序。好的變數指派和值域指派的排序，會減少搜尋過程中回溯的數目，可幫助求解時的效率。而在前瞻式 (lookahead) 的演算法中，排序的好壞會影響刪除搜尋空間的多寡。此類方法有 failed-first principle、minimal width ordering 等演算法。

(3) 一致性檢驗演算法 (consistency checking algorithms)

一致性檢驗演算法主要是藉由對限制條件的初步檢查，將搜尋空間中不合理的值域刪除，以減少搜尋的空間，增加搜尋時的效率。此類演算法具有下列三點好處：(a) 減少搜尋的空間，增加搜尋時的效率，(b) 避免搜尋時重複搜尋無效的狀態空間，(c) 提早察覺出無合理解 (insoluble) 的問題。而所謂的 k -consistency，表示在 n 個變數中，任意 k 個變數之組合，都必須滿足相關的限制條件， k 愈高表示減少的搜尋空間愈多，搜尋的效率愈好 (若 $k = n$ ，則搜尋過程不會發生回溯現象)。若為 n -consistency，則表示 n 個變數中的值域都能滿足所有的限制條件，亦即此限制條件滿足問題已找出所有的可行解，但是相對所花費的時間與費用愈高。因此在使用此技術時，必須在一致性檢驗所花費的時間與搜尋到解答之品質中取得平衡。

(4) 圖形結構驅動演算法 (structure-driven algorithms)

限制網路 (constraint network) 為限制條件滿足問題的圖形化表示方法，藉由限制網路的局部特徵可用來導引求解限制條件滿足問題的策略，而此方法可支援一致性檢驗與搜尋演算法，以應用於特定的問題。此類演算法有 tree search algorithm、tree clustering method 及 graph-based backjumping 等。

(5) 隨機搜尋演算法 (stochastic search algorithms)

利用一般搜尋演算法進行求解限制條件滿足的問題。例如：登山法 (hill climbing)、禁忌搜尋法、模擬退火法與基因演算法等。

(6) 其他演算法

針對用來解決限制條件滿足問題的演算法卻不屬於上述各類的演算法皆可歸納為此類，例如解答合成法 (solution synthesis) 或是上述各類演算法混合 (hybrid) 而成的方法等。

另一方面，Tsang[40]曾針對限制條件滿足技術做過完整的調查，並指出應用限制條件滿足技術時，利用個別限制條件滿足問題的特徵 (feature) 選擇適當的限制條件滿足技術，可增加求解的效率。相關的限制條件滿足問題特徵可分為六項：(a) 解答數量的要求，(b) 問題的大小 (size)，(c) 變數與限制條件的種類，(d) 限制圖形的特

徵，(e) 問題的緊密程度 (tightness)，(f) 解答的品質。

2.3.2. 基因演算法之相關探討

基因演算法是結合最適殘留者與隨機資訊交換的一種創新的搜尋法。它是期望由過往的資訊中，藉著新搜尋點之開發來改善績效。

目前的搜尋法主要分為三類：積分式，列舉式和隨機式。

積分式方法又可分為二種：間接和直接。間接方法是尋找區域極值，用來解決目標函式是收斂至零的非線性方程式之集合。直接方法是尋找區域最佳解，藉由區域性地移動同一個方向漸近來運作，簡單的代表就像是爬山法。二種方法都尋找區域解，最佳解也是搜尋所在點的鄰域最佳值。

列舉式方法則考量很多形狀及大小，它定義一個有限或可數的無限搜尋空間，一次把這些空間中的每一點帶入目標函式得其值。這個方法的缺點是效率太差。

隨機式方法需要長時間的執行，所以效率上並不期望比列舉式方法來得好。比較受歡迎的技術有基因演算法及模擬退火法。

基因演算法不同於一般最佳化和搜尋的程序有四方面：

- (1) 基因演算法的運作是藉由參數集合的編碼，而不是參數本身。
- (2) 基因演算法是搜尋一個族群，而非單點。
- (3) 基因演算法是使用目標函式的資訊，不是額外的知識。
- (4) 基因演算法使用機率性法則，而非決定性法則。

大部分最佳化方法的點對點運作方式是很危險的，因為有可能會找錯峰點，特別是有多峰點的搜尋空間。相反地，基因演算法提供了能同時平行搜尋多個峰點的機制，因此，使得找錯峰點的機率降低了。

直接使用編碼、搜尋一個族群、不需要額外知識及隨機運算子，這些是使得基因演算法強韌的原因。

自 Holland 於 1975 年提出基因演算法後，許多學者紛紛投入此方面的研究，如圖 2.9、圖 2.10 所列的文獻。而這許多的應用經驗，

顯示基因演算法是在解決最佳問題上為一極可行之方法，尤其適合於多重限制性最佳化之問題，如排程 規劃以及資源配置等方面之問題

最佳化問題		
年	作者	標題
1996	Yokota	Study on solving system reliability optimization problems with interval data by genetic algorithms.
1996	Liu	Application of multiobjective genetic algorithm algorithms to control system design.
1996	Michael	Form-finding analysis of vibrating towered shells of revolution as an inverse eigenproblem and as a genetic algorithm optimization problem.
1996	Murata	Genetic algorithms for multiobjective optimization.
1997	Canploat	Optimization of seasonal irrigation scheduling by genetic algorithm.
1998	Filho	Optimizing hydrocarbon field development using a genetic algorithm-based approach.
1998	Matthew	Applications and limitations of genetic algorithms for the optimization of multivariate calibration techniques.
1998	Ikonen	Genetic algorithm for a three-dimensional non-convex bin packing problem.
1999	Zhou	Study on constrained spanning tree problems with genetic algorithms.
規劃與排程		
年	作者	標題
1996	Edirisinghe	Optimization of radiotherapy planning using genetic algorithms.
1996	Moridawa	Study on scheduling using genetic algorithms.
1996	Wall	Genetic algorithm for resource-constrained scheduling.
1997	Al-harkan	On merging sequencing and scheduling theory with genetic algorithms to solve stochastic job shop problems.
1997	Cheng	Study on genetic algorithm-based optimal scheduling techniques.
1997	Hocaoglu	Multipath planning using evolutionary computation with specification.
1997	Lin	Genetic algorithm-based scheduling system for dynamic job shop scheduling problem.
1997	Wang	Genetic algorithm-based approach to subtask matching and scheduling in heterogeneous computing environments and a comparative study of parallel genetic algorithms.
1999	Li	Study on hybridized genetic algorithms for production distribution planning problems.

圖 2.9 基因演算法之相關研究 (一)

基本理論

年	作者	標題
1996	Crawford	Role of recombination in genetic algorithms for fixed-length subset problems.
1996	Eberlein	GA heuristic generically has hyperbolic fixed points.
1996	Hightower	Computational aspects of antibody gene families.
1996	Kargupta	Search, polynomial complexity, and the fast messy genetic algorithm.
1996	Merkle	Analysis of linkage-friendly genetic algorithms.
1996	Picardo	Framework for the analysis of evolutionary algorithms.
1996	Sato	TA proposal and analysis on the alternation of generation models of genetic algorithms.
1996	Stumpf	Studies on enhanced operator-oriented genetic algorithms.
1996	Wang	Matrix genome encoding for genetic algorithms.
1996	Wong	Performance analysis for genetic algorithms.
1996	Xie	Properties and characteristics of genetic algorithms as a problem-solving method.
1997	Harik	Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms.
1997	Kaiser	Dynamic load distributions for adaptive computations on MIMD machines using hybrid genetic algorithms.
1997	Potter	Design and analysis of a computational model of cooperative coevolution.
1997	Rosin	Coevolutionary search among adversaries.
1998	Rasheed	GADO: a genetic algorithm for continuous design optimization.
1998	Sarma	Analysis of decentralized and spatially distributed genetic algorithms.
1998	Spears	Role of imitation and recombination in evolutionary algorithms.
1998	Voicu	Multiresolution aspects in genetic algorithms.

圖 2.10 基因演算法之相關研究 (二)

2.3.2.1. 基因演算法之簡介

基因演算法又稱遺傳演算法係由 John Holland 於 1975 年首度發表，發展基因演算法的動機，在於他們希望發展出具有強大能力且能被廣泛應用的演算法。藉由基因組合交換的方式，能在解答空間中不斷產生出新的點，以改善系統績效。其目的在於將自然界 (natural system) 生態的演進過程以較為抽象化的方式解釋，以及設計並建立一個融合自然生態特性的人工計算系統 (computing system)。

圖 2.11 是基因演算法演算過程。其中包含幾項重要的步驟，分別為：(1) 定義表示法；(2) 根據問題的特性，定義適應函式 (fitness function)；(3) 設計基本運算子 (operators)，包含複製 (reproduction) 交配 (crossover) 突變 (mutation) 等三個基因運作流程；(5) 決定參數，如交配率、突變率，以及終止條件。分述如下：

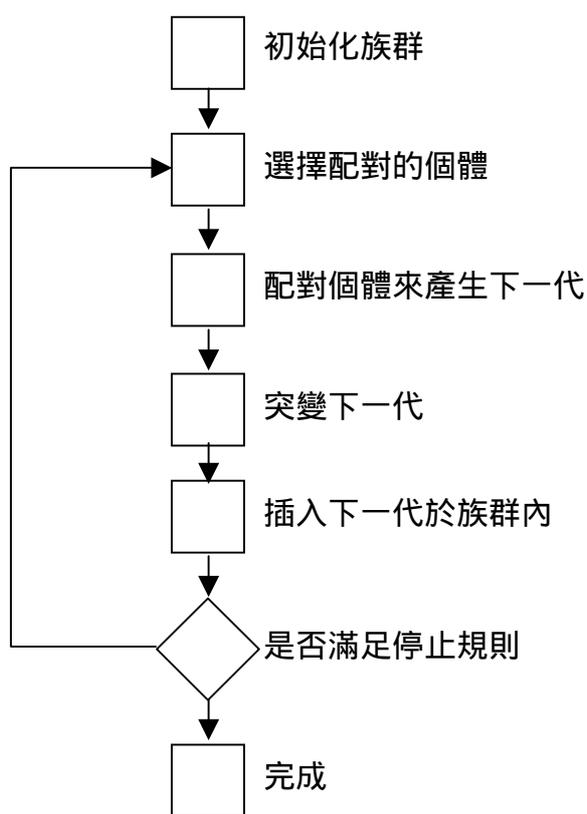


圖 2.11 基因演算法流程圖

(1) 定義表示法

根據問題來使用適當的資料結構。定義一個適用的表示法是使用基因演算法的藝術之一，要完全反應出問題解的特性。表示法除了要表現出問題的合理解外，所有可能表現出不合理解的表示法要儘量避免，因為愈能表現出不合理解的表示法愈有可能讓目標函式值變得不合理。表示法應該不會包含超過表現出問題解的額外資訊，雖然它的優點是包含一些額外的基因使搜尋空間變大，但是阻礙基因演算法的效益。

a. 編碼 (coding)

編碼在基因演算法的過程中，是最重要也是最困難的一環，編碼的選擇將會影響基因演算法搜尋的結果。為了避免產生無效的變數值，首先要先確認每個變數的合理值域，再將每個變數以固定長度的

字串加以編碼。編碼方式可分為二進位（又稱間接編碼）編碼方式（binary coding）與直接編碼（direct coding），主要的限制是解答空間要能夠以離散型態（discrete value）表達。基因演算法染色體編碼有以下幾點關鍵因子（key issue）：落在 Feasible area 的染色體稱為合理解，落在求解空間的染色體稱為合法解。染色體由編碼空間對照回求解空間，有三種對照關係 1 對 1、n 對 1 及 1 對 n 對應關係。其中以 1 對 1 為最佳的編碼，而 1 對 n 則為最不希望產生的編碼方式。如圖 2.12 所示。

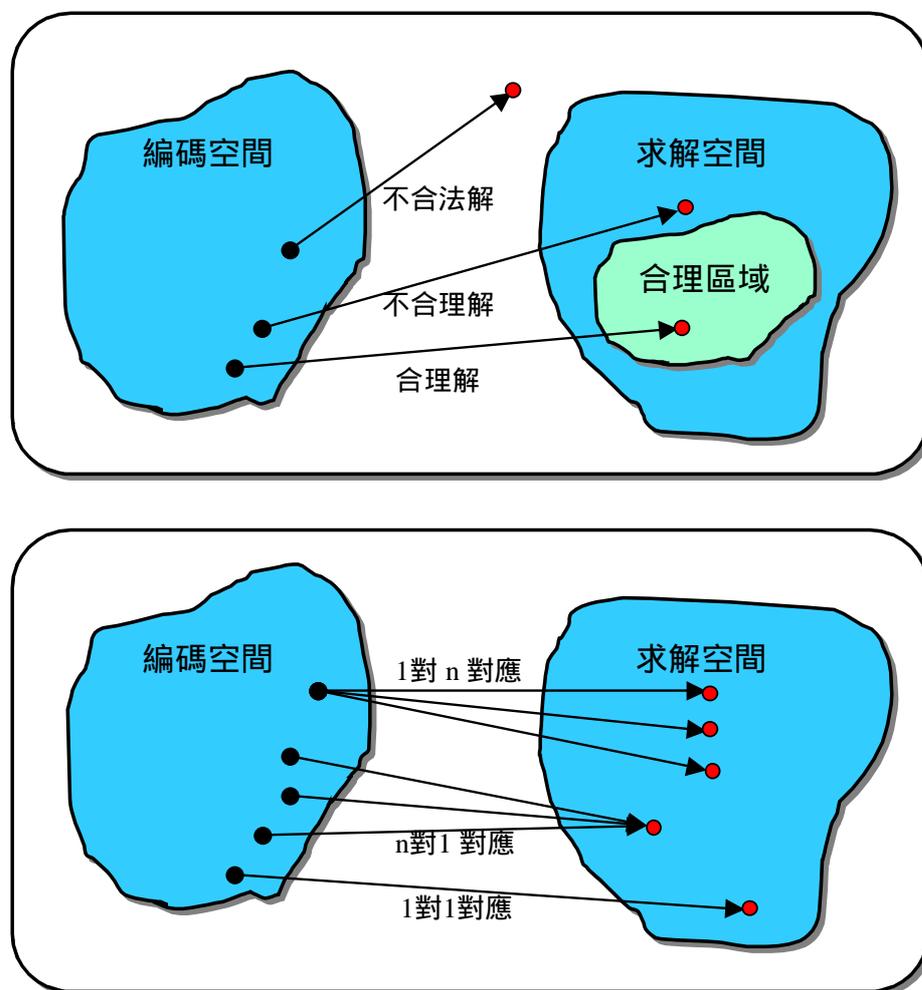


圖 2.12 基因編碼及對應關係

b. 產生位元字串 (bit string)

將每個變數，依照編碼方式所產生的字串，依序連接之後，變成為一個完整的位元字串，在基因演算法中稱為「染色體」

(chromosome) 或稱為「個體」(individual)，每條染色體便代表解答空間的一個點。基因演算法便是模擬物種進化的過程，保留適應能力高的基因，淘汰適應能力差的基因，使得整體的績效會隨著一代一代的演化而改善。

(2) 定義適應函式 (fitness function)

基因演算法最吸引人的地方莫過於它是一種漸近式的搜尋法，因為它不需要編輯不同的方程式或是一種平滑的搜尋空間。基因演算法需要有一個單一個體跟其它個體比較的衡量指標，目標函式就是提供這樣的衡量標準。最重要的是適應和目標值是不同的。目標值是由目標函式得來的，它是單純衡量一個基因的績效；適應值是基因演算法來決定個體繁衍下一代之適應能力的一個可能轉換比率。它是由目標值之線性排列而獲得的。

根據不同的問題，發展出不同的適應函式，適應函式是表達問題的績效函式，在基因演算法中，適應函式的目的便是判斷某一組變數值之集合的適應程度，最終的目的就是要找到使適應值最高的那組變數值。

(3) 設計基本運算子 (operators)

每條基因有四個主要的運算子：初始化、複製、突變和交配。根據那些運算子可以修正初始的族群來針對問題的表示法，或是族群進化時的演化部份，定義複製、突變和交配運算子。

初始化運算子決定基因如何被初始化，它不止實際建立新的基因組，而且填塞所有演化解的原始基因；複製運算子類似生物的無性生殖。根據每個個體的適應函式值高低，決定該個體被複製的機率；突變運算子定義每條基因的突變程序。一般而言，突變具有探索和開發的功用，它會引入新的基因來修正已存在的基因；交配運算子定義由二個母基因來產生一個子基因的程序，不像突變運算子，它影響到多

個基因。

除了四個主要的運算子之外，每條基因也都包含一個目標函式和比較測定式。目標函式是用來評估基因的，比較測定式（距離函式）是用來決定基因之間的差異性。每個基因演算法需要一個被定義好的目標函式—這是讓基因演算法決定那一個個體比其它的來得好。某些基因演算法則需要一個比較測定式。

一般而言，基因演算法不需要知道資料結構的運作內容，就能夠知道為了建立族群如何複製基因組，執行初始化基因組，透過基因組來產生子代，和突變基因組。

（4）決定參數

基因演算法的參數設定將會影響，演算收斂的速度。過早收斂影響解的品質，太慢收斂則耗費過多資源。因此，遺傳演算法在正常執行下，群體演化過程將趨向於全域最佳解。如何判斷系統是否已經收斂、何時該停止演化，或目標是否達成，常用的方式有下列幾項：

- a. 演化代數：直接設定演化幾個代數即停止演算，至於應該幾代才適合需視問題複雜度及資料量而定。
- b. 演化時間：設定演化時間，時間到達後即停止運算。
- c. 當解經幾個世代的演化而無改變時，即可視為找到最佳解。
- d. 當最佳適合度函式值的變動量誤差持續數代在某個百分比內時，即可視為找到最佳解。

2.3.2.2. 基因演算法之類型

2.3.2.2.1. 簡單型（simple）基因演算法

建立一個簡單型基因演算法時必須指定一個個體或是族群，如圖 2.13 所示。新的基因演算法會複製所指定的個體來產生它自己的族群。建立後或是在演化的過程中，可以改變大部份基因演算法的行為。簡單型基因演算法藉由複製基因或是指定族群的方式來建立最初

的族群。每一代，演算法會由前一代選擇個體來建立新的族群，然後配對產生新的子代。這個過程會持續至達到停止規則為止。

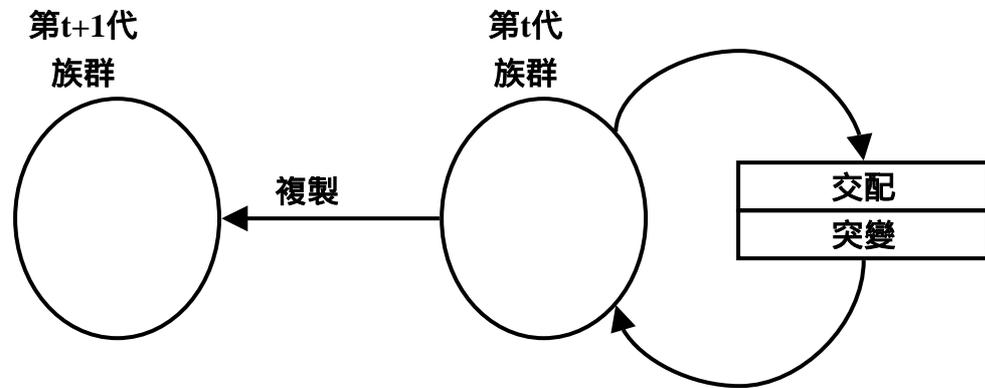


圖 2.13 簡單型基因演算法的運作概念圖

2.3.2.2.2. 穩態型 (steady-state) 基因演算法

它用使用者指定的重疊數之重疊性族群，藉由複製基因或是指定族群的方式來建立族群，如圖 2.14 所示。每一代，演算法會建立一個暫時的族群，把它加入先前的族群，然後移去最壞的個體來保持族群的原有個數。

若指定一個替代比率，然後每一代族群就會根據這個比率被替代。另外一種是指定每代的替代個體數。

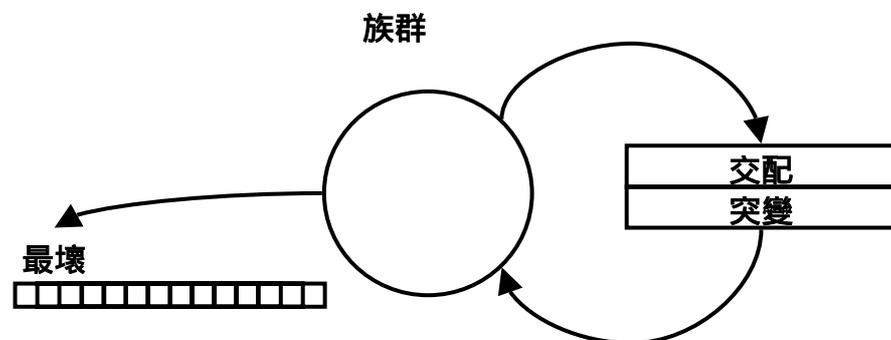


圖 2.14 穩態型基因演算法的運作概念圖

2.3.2.2.3. 內婚型 (deme) 基因演算法

這個基因演算法有多重獨立的族群，它是藉由複製基因或是指定族群的方式來建立族群組。每一個族群演化都是一個穩態型基因演算法，但是每一代中的個體是由一個族群移到另一個族群。這種遷移的演算法是確定性一步步交換地；每個族群移轉固定個數的最好個體到它的鄰域。主要的族群是更新每個族群中每一代的最好個體。

2.3.2.3. 基因演算法之基本運算子

基因演算法主要包含了三個運算子，分別為複製 (reproduction)、交配 (crossover) 及突變 (mutation)。以下分別說明之：

2.3.2.3.1. 複製運算子

根據每個個體的適應性函式值高低，來決定該個體被複製的機率。當適應性函式值越高的個體，其複製機率將相對提高，反之則越低。所以在下一子代的族群中，適應性函式值較低的個體將逐漸減少，而取而代之的則是適應性函式值較高的個體。

基因演算法常見的複製機率策略有下述三種[13]：(a) 等機率法：每個個體被複製的機率均相等。(b) 比率法：根據個體適應性函式值佔總個體函式值和的比例，決定其複製機率。(c) 排序法：個體被複製的機率決定其適應性函式的排序值。

2.3.2.3.2. 突變運算子

突變的過程是將交配後產生的子代，根據預設的突變機率進行突變，其主要的目的是避免過度提早收斂，且可以開發新的搜尋區域，避免收斂於局部最佳解。圖 2.15 顯示較為人熟知的的突變運算子運作方式。

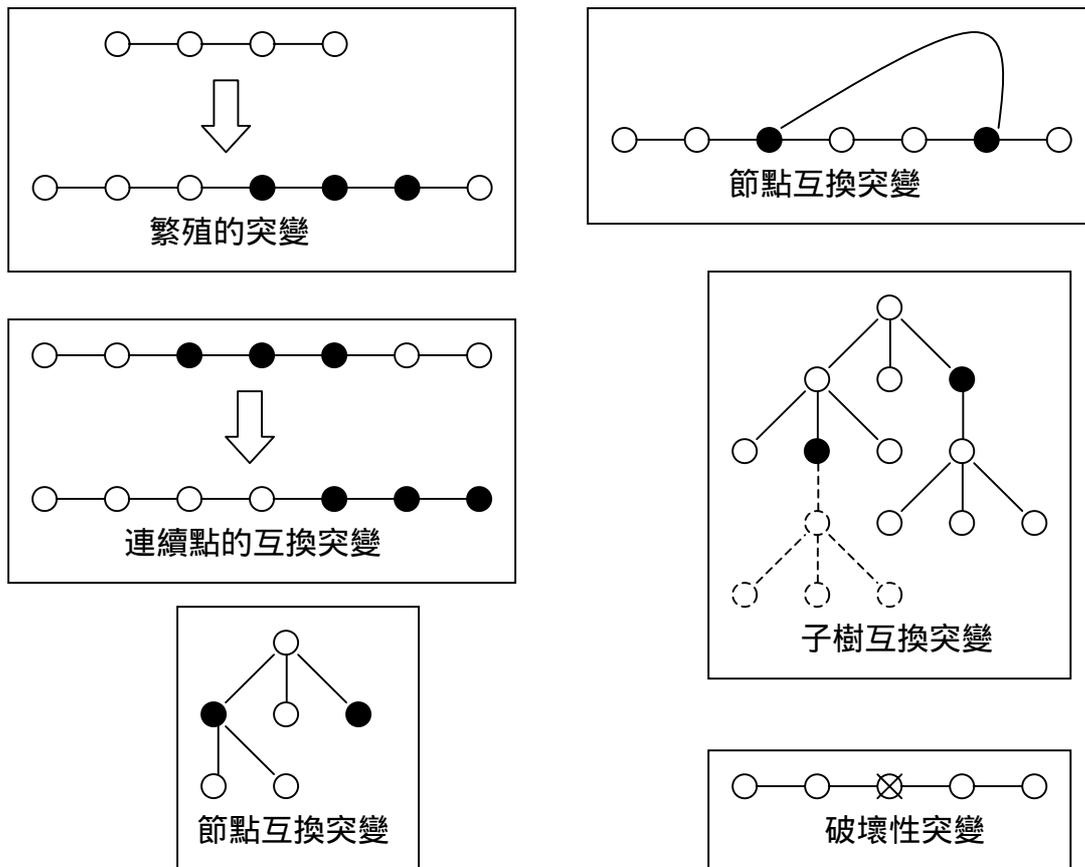


圖 2.15 各種不同的突變運算子運作方式

2.3.2.3.3. 交配運算子

交配運算子在基因演算法中可算是最重要的運算子，因為透過交配運算子可以將基因字串(gene strings)重新排列組合以產生不同型態的染色體(chromosomes)。圖 2.16 顯示較為人熟知的交配運算子運作方式。

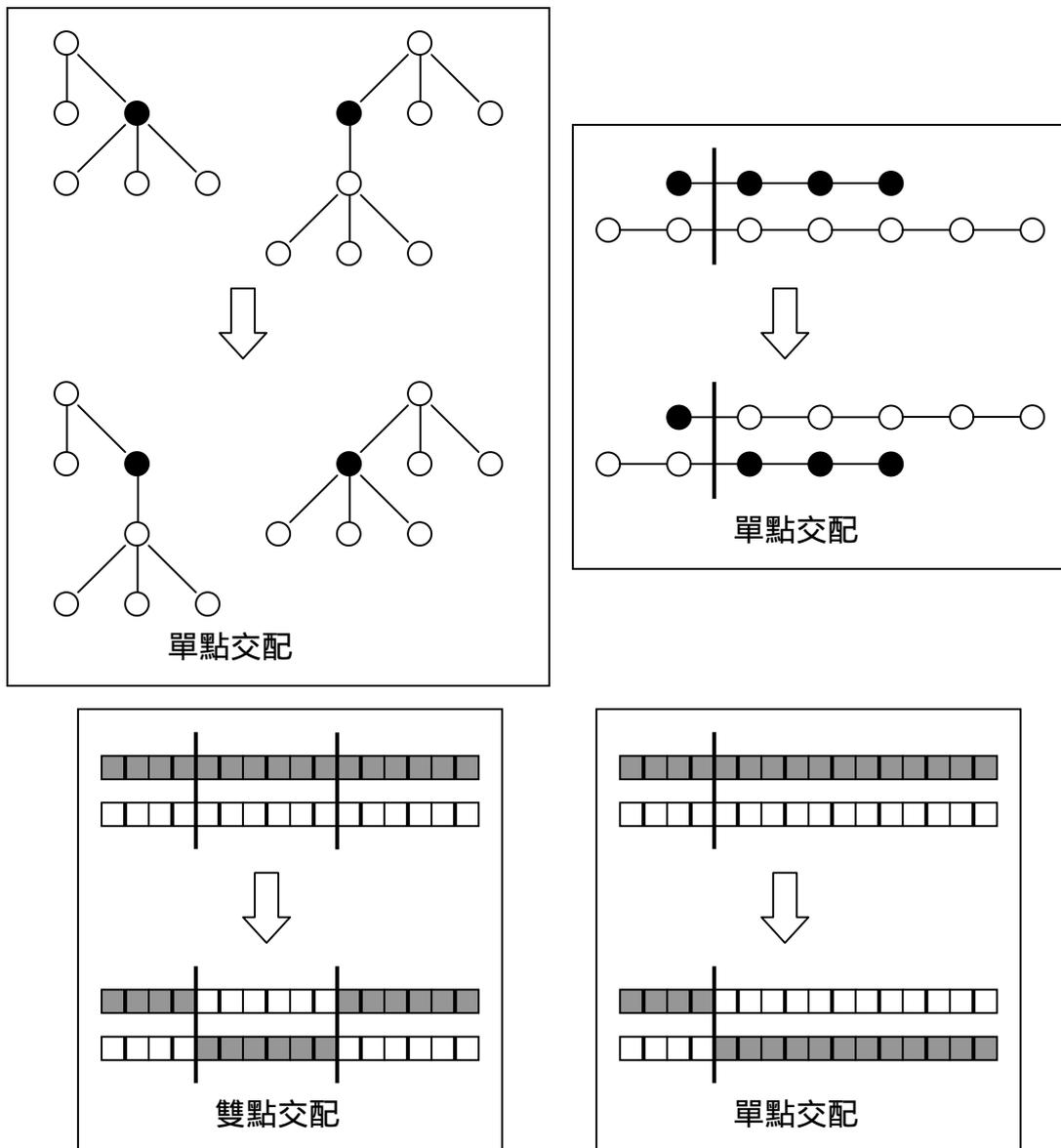


圖 2.16 各種不同的交配運算子運作方式

2.3.2.4. 基因演算法的運作理論

基因演算法運用許多隨機計算，決定初始群組、選擇方法、複製、交配及突變，這些運算為基因演算法的基本運作程序。但這樣的過程如何順利運作，Holland 於 1975 年提出了基因運作的基本理論 (the fundamental theorem)，稱之為基原理論 (Schema Theorem ; ST)，奠

定了基因演算法的理論基礎。

組成字串的基本要素稱之為基原 (schemata)，二位元編碼由 (0,1,*) 三個字元所組成。如 $H_1 = (0****1)$ 、 $H_2 = (**10**)$ 、 $H_3 = (0*1*01)$ 稱為基原，其中「*」表示不在乎其值為何，可表示為 0 或 1。假若在二元字串中有 n 個「*」，表示此基原將會有 2^n 個基因字串表示之，如有 2 個「*」，則此基原有 2^2 個基因字串表示，如表 2.5 所示。基原有兩個基原參數—基原秩序 (schema order) 及基原長度 (schema length)。

表 2.5 基原字串對照表

基原	1	*	0	0	*	1
字串 1	1	0	0	0	0	1
字串 2	1	0	0	0	1	1
字串 3	1	1	0	0	0	1
字串 4	1	1	0	0	1	1

基原秩序：基原 H 在染色體字串中的個數，以 $o(H)$ 表示。如：基原 $H_1 = (1*0**1)$ ，則 $o(H) = 3$ ，或基原 $H_2 = (1**0**)$ ，則 $o(H) = 2$ 。

基原長度：基原 H 在染色體中的最大長度，以 $\delta(H)$ 表示。如：基原 $H = (1*0*1*)$ ，則 $\delta(H) = 5 - 1 = 4$ ，或基原 $H = (1*****)$ ，則 $\delta(H) = 1 - 1 = 0$ 。

以基原來說明複製、交配及突變，三個運算子如何提高個體的適存度達到演化的目的[27]。

(1) 複製對適應值的影響

在第 t 代族群中以基原 H 所對應的字串數 m ，以 $m(H, t)$ 表示，

則下一代 $t+1$ 族群中基原 H 所對應的字串數 m 為：

$$m(H, t+1) = m(H, t) * \frac{f(H)}{\bar{f}} \dots\dots\dots (公式 2.1)$$

其中 \bar{f} 為第 t 代族群中所有字串的平均適應函式值；

$f(H)$ 為第 t 代包含基原 H 的字串平均適應函式值。

令 $f(H) = \bar{f} + c\bar{f}$ 代入 (公式 2.1) 得

$$m(H, t+1) = m(H, t) * \frac{\bar{f} + c\bar{f}}{\bar{f}} = m(H, t)(1+c) \dots\dots\dots (公式 2.2)$$

其中 c 為常數。當 t 從第 0 代開始到第 t 代時，族群中包含基原 H 的字串各數為：

$$m(H, t) = m(H, 0)(1+c)^t \dots\dots\dots (公式 2.3)$$

由 $f(H) = \bar{f} + c\bar{f}$ 可知，當 $f(H)$ 大於族群平均適應函式值時，常數 c 值為正值，所以 $m(H, t)$ 將成指數遞增。反之，當 $f(H)$ 小於族群平均適應函式值時，常數 c 值為負值，則 $m(H, t)$ 將成指數遞減。可得知適應值較高的基原 H 會被保留且越來越多，適應值低的會逐漸減少，最後被淘汰。

(2) 交配對應值的影響

以單點交配說明，基原為 H 因交配而被破壞而消失的機率為：

$$P_c * \left(\frac{\delta(H)}{l-1}\right) \dots\dots\dots (公式 2.4)$$

其中 P_c 為交配率， $\delta(H)$ 為基原 H 的最大長度； l 為基因字串的總長度。

若考慮基因的存活率 P_s 則可表示為：

$$P_s \geq 1 - P_c * \left(\frac{\delta(H)}{l-1}\right) \dots\dots\dots (公式 2.5)$$

算式右邊表示基原 H 交配後存活的機率，而 P_s 為交配後基因存活的機率。將上式帶入 (公式 2.1) 得：

$$m(H, t+1) \geq m(H, t) * \frac{f(H)}{\bar{f}} (1 - P_c * \left(\frac{\delta(H)}{l-1}\right)) \dots\dots (公式 2.6)$$

由上式可得知基原 H 會因交配而較上一代增加。

(3) 突變對適應值的影響

每一基因突變機率為 P_m ，所以不突變的機率為 $(1-P_m)$ 。基原 H 不因突變而被破壞而消失的存活率為 $P_s = (1-P_m)^{o(H)}$ ，經由二項式展開因 P_m 的值遠小於 1，簡化得到 $P_s = 1 - o(H)P_m$ ，將 P_s 帶入 (公式 2.6) 可得：

$$\begin{aligned} m(H, t+1) &\geq m(H, t) * \frac{f(H)}{f} (1 - P_c * (\frac{\delta(H)}{l-1})) (1 - o(H)P_m) \\ &\geq m(H, t) * \frac{f(H)}{f} (1 - P_c * (\frac{\delta(H)}{l-1}) - o(H)P_m - P_c P_m o(H) * (\frac{\delta(H)}{l-1})) \end{aligned}$$

因 $P_c P_m o(H) * (\frac{\delta(H)}{l-1})$ 微小，忽略可得下式：

$$m(H, t+1) \geq m(H, t) * \frac{f(H)}{f} (1 - P_c * (\frac{\delta(H)}{l-1}) - o(H)P_m) \dots \text{(公式 2.7)}$$

由上式可知，若基原的 $o(H)$ 值小、 $\delta(H)$ 值小、基原平均適應函式值高，則隨著運算次數增加，而能提高下一代適應函式值，達到演化目的。

2.3.2.5. 基因演算法處理限制條件滿足問題

基因演算法是一項可靠，且穩定的廣域搜尋技術，但美中不足的是不具備處理限制條件問題的能力，因此必須具備額外的機制才能解決限制條件滿足問題。Gen[13]將這些方法歸納為下列四種策略：

(1) 拒絕策略 (reject strategy)

一旦基因演算法所產生的個體值落入不可行區域內，即直接捨棄該個體。

(2) 修正策略 (repairing strategy)

一旦基因演算法所產生個體違反限制條件，即進入基因修正程序，修正該基因為可行解。如 Liepins 等[24]求解集合涵蓋 (set covering) 及銷售人員旅行 (traveling salesman) 問題；Michalewicz 等[28]所提出的 GenoCop 以及 Paredis[29][30][31]所提出的 Genetic

State Space Search (GSSS) 法等。

(3) 改變基因演算策略 (modifying genetic operator strategy)

根據問題的特性，對於染色體及基本運算子加以修正，使的演算法在演算過程中，避免出現違反限制條件之結果。Falkenauer[11]根據此原則，設計染色體表達方式並修正基本基因運算子來求解工作站排程 (job shop scheduling) 問題。

(4) 懲罰策略 (penalty strategy)

在原目標函式中，增加一懲罰項。懲罰值的大小視其個體違反限制式的程度而言。當違反的限制條件越多，相對的懲罰值將越大；反之則愈小。如此一來，便可將有限制式問題轉換成無限制式問題，方便基因演算法求解。有關基因演算法的運用處罰函式求解限制性問題，Coit[8]針對可靠度最佳化的問題，提出結合處罰函式及基因演算法的概念，而該方法可以有效且迅速的找出可行的最佳解。

2.4. 本章結論

為了協調廠區間的運輸，多廠區生產規劃通常處理一個可共享的庫存。因此，企業藉由數個具有互補或替代生產能力的廠區，去規劃每個廠區間的互動合作。總結第 2.1 節的文獻探討，在多廠區生產系統的環境中，產品的生產流程是需要使用跨廠區間的資源，亦即產品的生產流程是存在替代性。解決多廠區生產的問題，是要改善不同廠區之間在生產規劃與控制層面的合作問題。

歸納第 2.1 及 2.2 節的相關研究，本論文針對一般性多廠區生產系統提出產銷平衡的規劃邏輯，進而發展產銷平衡的訂單分配模式。同時，對於第 2.2 節商用軟體的作法彙整出一般性供應鏈規劃的邏輯流程圖，並整理於附錄 A。最後，以限制條件為基礎的基因演算法求解於台灣液晶顯示器產業的多廠區生產環境下之訂單分配問題。

第三章 多廠區生產規劃系統的架構與訂單分配模式

3.1. 多廠區生產規劃系統的架構

總結第二章文獻探討之內容，本論文歸納出同一企業內的一般性多廠區生產規劃系統可分為三大模組：(1) 訂單進入模組；(2) 訂單分配模組；(3) 訂單開立模組，如圖 3.1 所示。

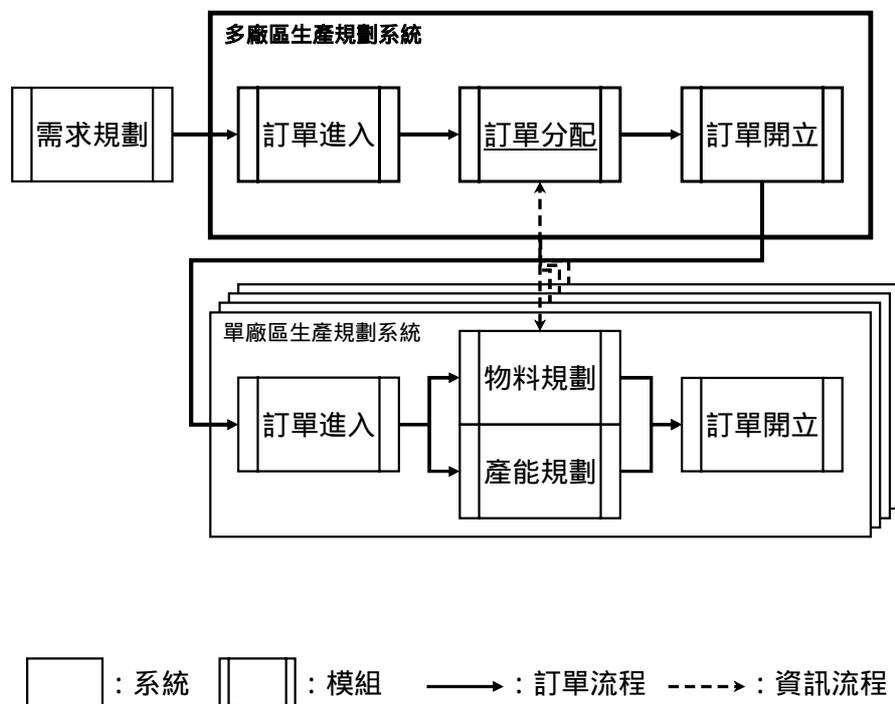


圖 3.1 一般性多廠區生產規劃系統的基本模組架構

「需求規劃」模組彙總整個供應鏈（來自於外部企業或顧客）的需求訂單（Demand Order；DO），需求來源包括顧客及預測需求，將這些訂單資訊輸入至多廠區生產規劃系統的「訂單進入」模組。「訂單進入」模組負責收集需求訂單相關的生產資料傳送給「訂單分配」模組執行訂單分配的功能，包括物料清單（Bill Of Materials；BOM）

與途程資料 (routing date)、資源清單 (Bill Of Resources ; BOR) 與產能負荷 (capacity loading) 狀況、及訂單資料 (交期、數量、需求來源等) 與成本資訊等。

「訂單分配」模組為多廠區生產規劃系統的核心功能，主要是依「訂單進入」模組所輸入的生產資料，將需求訂單分配給各個工廠，其輸出的資訊是各個工廠所需要生產及運輸的訂單數量與交期。接著，「訂單開立」模組負責把訂單分配的結果，即各個品項 (原物料、中間產品和最終產品) 的需求訂單開立至各個工廠，進入單廠區生產規劃系統的「訂單進入」模組執行各個工廠的基本資料輸入功能。

其中，「訂單分配」模組進行訂單分配的過程中，可能與單廠區生產規劃系統的「物料規劃」與「產能規劃」模組進行資訊傳達，以增進多廠區生產規劃系統的規劃結果之精確性與可行性。一般而言，多廠區生產規劃系統與單廠區生產規劃系統的互動關係分為三大類：

(1) 由上而下 (top down) 規劃方式—

由多廠區生產規劃系統規劃完的計畫，單廠區生產規劃系統完全接收規劃結果，並不立即回饋給多廠區生產規劃系統；

(2) 由下而上 (button up) 規劃方式—

先由單廠區生產規劃系統規劃後，再由多廠區生產系統規劃，並將結果交給單廠區生產規劃系統調整；

(3) 混合 (mix) 規劃方式—

先由多廠區生產規劃系統規劃後，再交由單廠區生產規劃系統規劃決定單廠區的主生產排程，接續回饋給多廠區生產規劃系統決定最後的輸出結果。

「訂單分配」模組又可細分為「產銷平衡」與「網路排程 (network scheduling)」子模組，如圖 3.2 所示。「產銷平衡」子模組主要是根據各個工廠的剩餘產能狀況及物料供給能力，同時依物料清單將最終產品的需求訂單 (外部需求) 展開各個品項的需求訂單 (內部需求)，且決定各個工廠所配置的需求訂單之數量，使得整個多廠區生產系統的供給與需求達成平衡狀態；「網路排程」子模組則承接「產銷平衡」

子模組的規劃結果，依各個工廠的生產目標與預計生產的訂單數量，決定所有訂單的預計完工時間，即內部需求訂單的交期，也就是最佳化各個工廠的生產目標以決定內部需求訂單的交期。

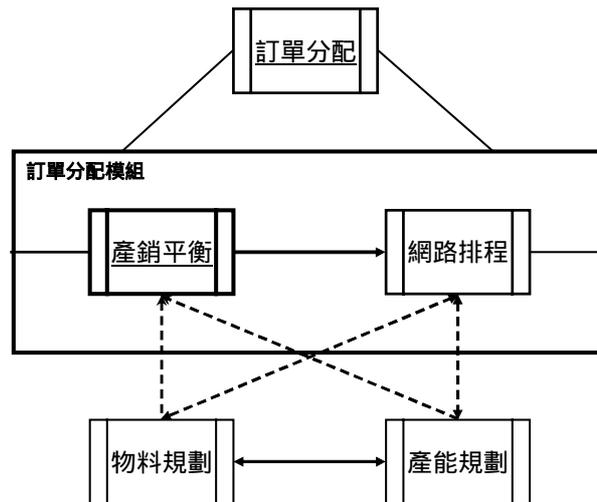


圖 3.2 訂單分配模組的子模組架構

本論文的目的就是實作多廠區生產規劃系統的「訂單分配」模組中「產銷平衡」子模組，為了建構多廠區生產系統（或稱為供應鏈網路），同時也提出以流程導向的建模（modeling）方式。

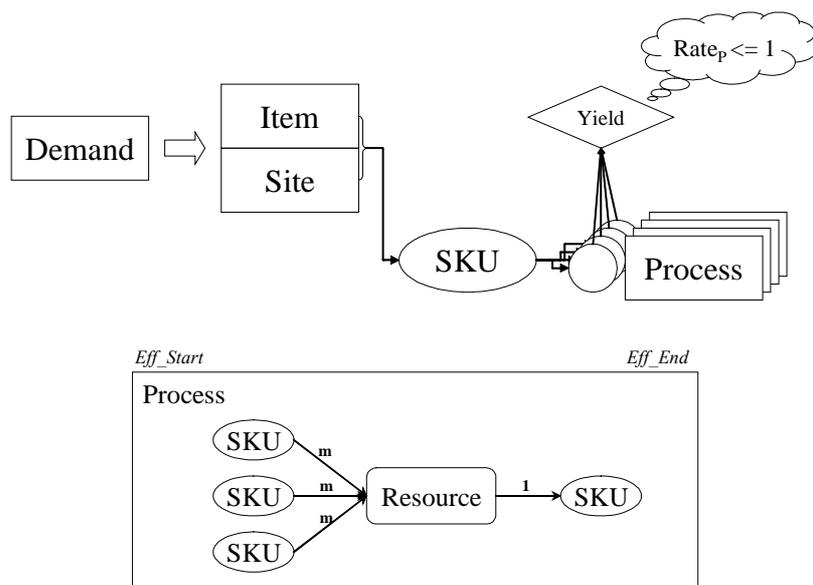


圖 3.3 多廠區生產系統的建模示意圖

圖 3.3 表示內部需求 (demand) 訂單的分配問題，主要決定哪個品項 (item) 在哪個廠區 (site) 製造或運輸多少數量？換句話說，訂單分配是對特定存貨 (inventory) 中的品項執行規劃或控制，這些品項稱為「儲存單元 (Stock Keeping Unit ; SKU)」，一般以 item@site 或 item@location (location 表示位置，本論文稱為廠區) 表示。每個儲存單元都是經由多個製造、運輸及儲存流程所規劃。以製造流程為例，不同 m 個儲存單元使用生產資源 (resource) 能夠製造成某一個儲存單元。

本論文將生產資源分為製造資源、運輸資源、採購資源及儲存資源等四類。以下針對不同的生產流程之建模方式及成本計算方式做一簡介。

(1) 製造流程的建模方式

製造流程主要是建構一儲存單元如何被製造出來，其構成元素包括輸入儲存單位 (相同廠區內的不同品項，如原物料或中間產品) 使用的製造資源 (機器設備或人力資源等) 及輸出儲存單位 (被製造

出來的品項，如中間產品或最終產品）。圖 3.4 是二個不同製造流程的建模範例。以製造 SKU1 為例，Process1 與 Process2 分別為替代流程，皆可製造出 SKU1。Process1 使用 CCR1 的製造資源將 m1 個 SKU2、m2 個 SKU3 與 m3 個 SKU4 製造成一個 SKU1；Process2 使用 CCR2 的製造資源將 m4 個 SKU5、m2 個 SKU3 與 m3 個 SKU4 製造成一個 SKU1。若不同流程內所組成的輸入儲存單位皆相同而生產資源不同，則稱此為替代資源的建模方式；若不同流程內所組成的生產資源相同而僅一個輸入儲存單位不同，則稱此為替代品項的建模方式。

因為耗用其它品項及資源產能，使得製造過程中增加品項的製造成本。其成本的計算公式如下：

$$\text{每單位製造成本 (Make_Prc_Cost)} = \text{輸入儲存單位的耗用成本 (SKU_Unit_Cost)} + \text{製造資源的耗用成本 (Make_Unit_Cost)}$$

以 Process2 為例，每單位 SKU1 的製造成本 (SKU1_Unit_Cost_P2)，是製造每單位 SKU1 所需要 m4 個 SKU5、m2 個 SKU3 及 m3 個 SKU4 的耗用成本 (SKU5_Unit_Cost * m4 + SKU3_Unit_Cost * m2 + SKU4 * m3)，加上每單位時間使用製造資源所耗用的成本乘上製造時間 (Capacity_Unit_Cost_CCR2 * Make_Time_P2)，即製造資源的耗用成本。每單位時間使用製造資源所耗用的成本是以每單位時間機器折舊成本或僱用人員薪資估計之。建模的假設情況如下：

- 建構具有替代品項(alternative item) 替代流程(alternative process) 與替代資源 (alternative resource) 等性質的製造流程。
- 該廠為製造工廠。

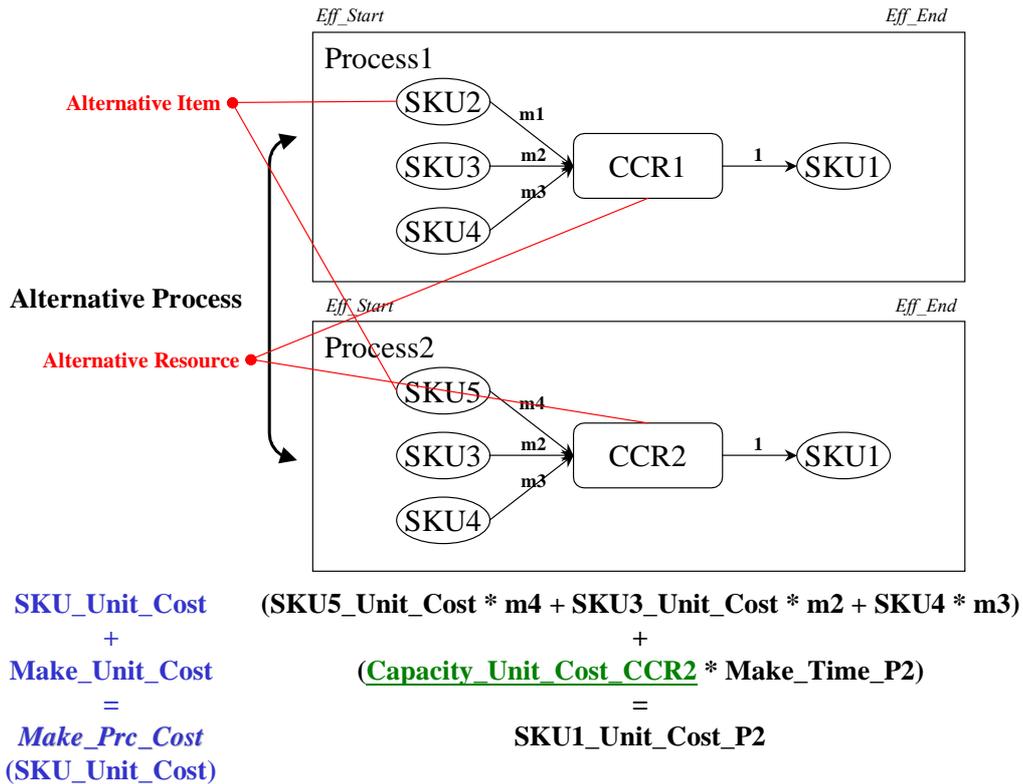


圖 3.4 製造流程的建模範例

(2) 運輸流程的建模方式

運輸流程主要是建構一儲存單位如何被運輸至該廠成為另一個儲存單位，其構成元素包括輸入儲存單位（它廠區內的某個品項，如原物料或中間產品）、使用的運輸資源（運輸設備，如貨車、貨船與飛機等）及輸出儲存單位（被運送至該廠的品項，如中間產品或最終產品）。圖 3.5 是二個不同運輸流程的建模範例。以運輸 SKU1 為例，Process3 與 Process4 分別為替代流程，皆可將 Item1 運輸至 Site1 且 Item1 與 Item3 互為替代品項。Process3 使用 Airline 的運輸資源將 Item3 由 Site2 (SKU6) 運輸至 Site1 替代 Item1 (SKU1)；Process4 使用 Truck 的運輸資源將 Item1 由 Site3 (SKU7) 運輸至 Site1 (SKU1)。

因為耗用其它品項及資源產能，使得運輸過程中增加品項的運輸成本。其成本的計算公式如下：

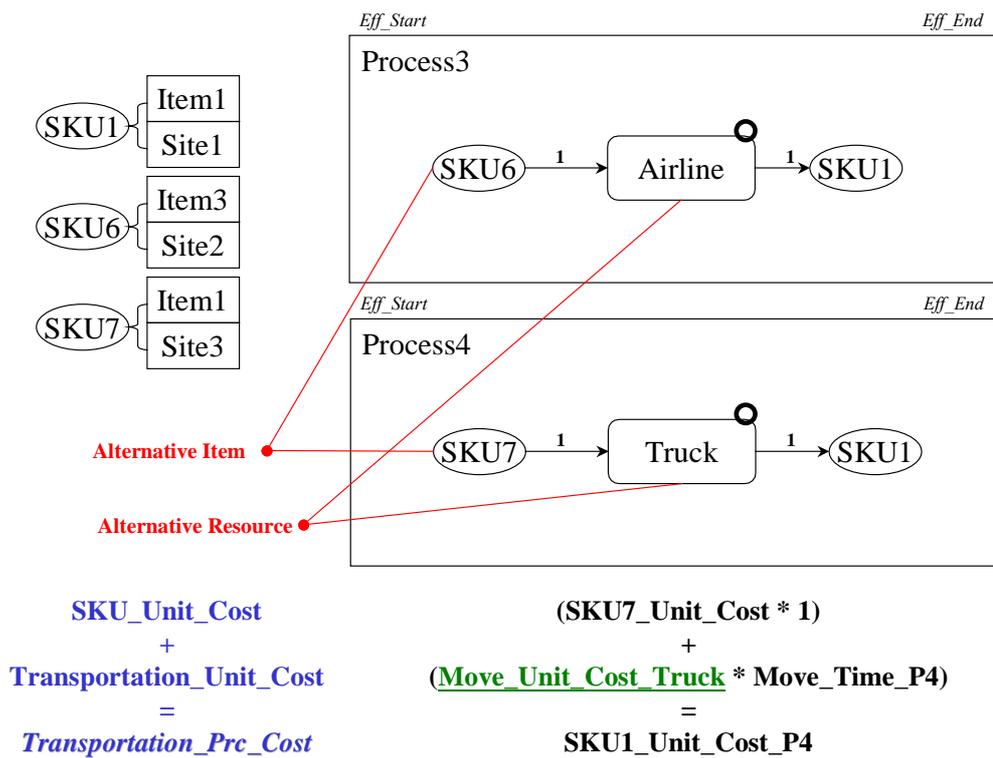


圖 3.5 運輸流程的建模範例

每單位運輸成本 (*Transportation_Prc_Cost*) = 輸入儲存單位的耗用成本 (*SKU_Unit_Cost*) + 運輸資源的耗用成本 (*Transportation_Unit_Cost*)

以 Process4 為例，每單位 SKU1 的運輸成本 (*SKU1_Unit_Cost_P4*)，是一單位 Item1 由 Site3 (SKU7) 運輸至 Site1 (SKU1) 的耗用成本 (*SKU7_Unit_Cost * 1*)，加上每單位時間使用運輸資源所耗用的成本乘上運輸時間 (*Move_Unit_Cost_Truck * Move_Time_P4*)，每單位時間使用運輸資源所耗用的成本是以每單位時間運送費用估計之，如空、海及陸運費用。

建模的假設情況如下：

- 建構廠際間具有替代品項 (alternative item) 替代流程 (alternative

process) 與替代資源 (alternative resource) 等性質的運輸流程。

- 建構廠際間的運送通路。

(3) 採購流程的建模方式

採購流程主要是建構一儲存單位如何採購其原物料至該廠的流程，其構成元素與運輸流程相同，與運輸流程之相異處在於成本計算方式及輸入儲存單位的品項為原物料。圖 3.6 是二個不同採購流程的範例。以採購 SKU2 為例，Process5 與 Process6 分別為替代流程，皆向 Site4 與 Site5 採購原物料 Item2 運輸至 Site1。Process5 使用 Airline 的採購資源向 Site4 採購原物料 Item2 且由 Site4 (SKU8) 運輸至 Site1 (SKU2)；Process6 使用 Truck 的採購資源向 Site5 採購原物料 Item2 由 Site5 (SKU9) 運輸至 Site1 (SKU2)。

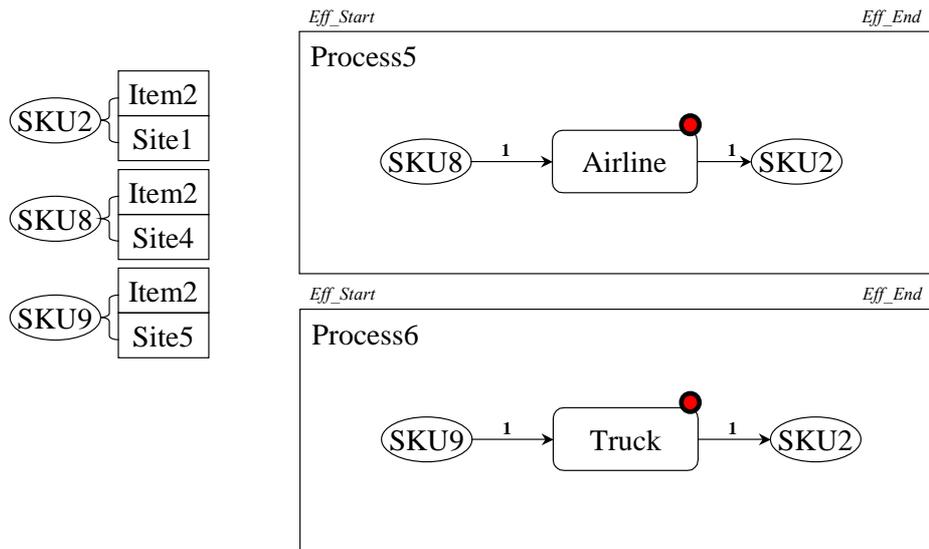
因為耗用其它品項及資源產能，使得採購過程中增加品項的採購成本。其成本的計算公式如下：

$$\begin{aligned} \text{每單位採購成本 (Purchase_Prc_Cost)} &= \text{每單位運輸成本} \\ &+ \text{採購資源的耗用成本} \\ &+ \text{每單位採購成本 (Purchase_Unit_Cost)} \end{aligned}$$

以 Process6 為例，每單位 SKU2 的採購成本 (SKU2_Unit_Cost_P6)，是每單位 Item2 由 Site5 (SKU9) 運輸至 Site1 (SKU2) 的運輸成本 (SKU2_Unit_Cost_P9 * 1)，加上每單位 SKU2 花費於採購作業的成本 (Buy_Unit_Cost_P6)。

建模的假設情況如下：

- 由供應商提供原物料給製造工廠的運送通路；
- 由外部企業所供給的品項 (原物料、中間產品或最終產品)，皆視為採購原物料。



$$\begin{array}{r}
 \text{Transportation_Prc_Cost} \\
 + \\
 \text{Purchase_Unit_Cost} \\
 = \\
 \text{Purchase_Prc_Cost}
 \end{array}
 \qquad
 \begin{array}{r}
 (\text{SKU2_Unit_Cost_P9} * 1) \\
 + \\
 \text{Buy_Unit_Cost_P6} \\
 = \\
 \text{SKU2_Unit_Cost_P6}
 \end{array}$$

圖 3.6 採購流程的建模範例

(4) 儲存 (Storage) 流程的建模方式

儲存流程主要是建構一儲存單位如何管理存貨的方式，其構成元素包括輸入儲存單位（該廠的某個品項，如原物料、中間產品或最終產品）使用的儲存資源（儲存設備，如倉庫等）及輸出儲存單位（該廠的品項）。圖 3.7 是二個不同儲存流程的建模範例。以儲存 SKU1 為例，Process7 與 Process8 分別為替代流程，皆可將 Item1 儲存至 Site1 且 SKU1 與 SKU10 互為替代品項。Process7 將 SKU10 入庫於 Dock 的儲存資源中，取料時以 SKU10 替代 SKU1；Process8 將 SKU1 入庫於 Dock 的儲存資源中，取料時再取出 SKU1。

其成本的計算公式如下：

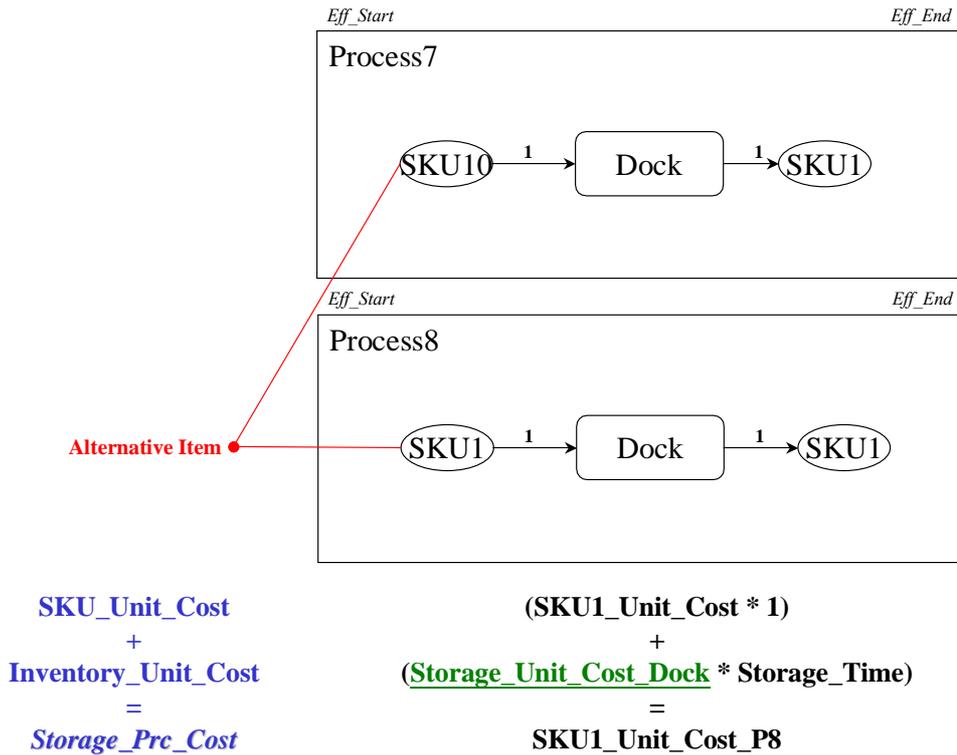


圖 3.7 儲存流程的建模範例

每單位儲存成本 ($Storage_Prc_Cost$) = 輸入儲存單位的耗用成本 (SKU_Unit_Cost) + 儲存資源的耗用成本 ($Inventory_Unit_Cost$)

以 Process8 為例，每單位 SKU1 的儲存成本 ($SKU1_Unit_Cost_P8$)，是一單位 Item1 儲存在 Site1 (SKU1) 的耗用成本 ($SKU1_Unit_Cost * 1$)，加上每單位時間使用儲存資源所耗用的成本乘上儲存時間 ($Storage_Unit_Cost_Dock * Storage_Time$)。每單位時間使用儲存資源所耗用的成本是以每單位時間占用儲存空間的費用估計之，例如租用倉儲中心的某個儲存空間，依占用空間大小計算每單位時間的儲存費用。

建模的假設情況如下：

- 運輸/採購流程之後；
- 製造流程之後；
- 該廠為物流/倉儲中心。

綜合以上四種生產流程的建模及成本計算方式，加總所有流程的單位成本，即可得到每單位儲存單位的單位成本，計算公式如下：

$$\begin{aligned} \text{每單位儲存單位的成本 (SKU_Unit_Cost)} = & \text{每單位製造成本} \\ & (\text{Make_Prc_Cost}) + \text{每單位運輸成本 (Transportation_Prc_Cost)} + \\ & \text{每單位採購成本 (Purchase_Prc_Cost)} + \text{每單位儲存成本} \\ & (\text{Storage_Prc_Cost}) \end{aligned}$$

以 SKU1 為例，每單位 SKU1 的成本 (SKU1_Unit_Cost)，是每單位 SKU1 的製造成本 (SKU1_Unit_Cost_P1 + SKU1_Unit_Cost_P2)，加上每單位 SKU1 的運輸成本 (SKU1_Unit_Cost_P3+ SKU1_Unit_Cost_P4)，每單位 SKU1 的儲存成本 (SKU1_Unit_Cost_P7 + SKU1_Unit_Cost_P8)。由於 SKU1 不是原物料，所以沒有採購成本。

依四種生產流程的特性，即可建構出最基本的供應鏈網路，如圖 3.8 所示。本論文將基本的供應鏈網路分為四個層次：供應商 (vendor / supplier)，製造工廠 (plant / factory / manufacturer)，物流中心 / 倉儲中心 / 轉運站 / 倉庫 (distribution center / warehouse / hub / bank) 與顧客 (customer)。

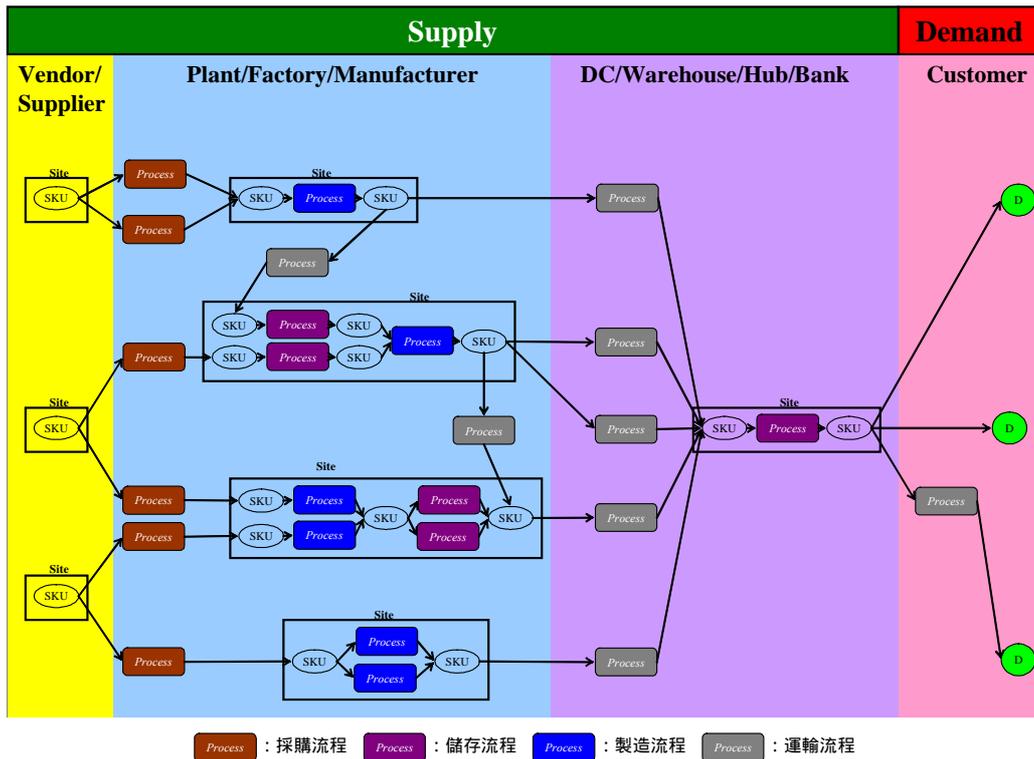


圖 3.8 以流程導向為建模方式的供應鏈示意圖

圖 3.8 的右半邊代表來自於顧客的需求，左半邊代表來自於企業內各個廠區的供給，整體而言，是以產銷平衡為基礎的供應鏈體系，本論文就是在此多廠區生產系統的架構下，來分配不同的生產流程（製造流程、運輸流程、採購流程與儲存流程）之產出或投入的訂單數量。

3.2. 多廠區產銷平衡的規劃邏輯

依流程導向的建模方式，本論文進而提出多廠區生產系統下產銷平衡的規劃邏輯。由最終產品的外部需求訂單展開為中間產品的內部需求訂單，不同工廠間的生產流程分派問題是由製程能力或生產指標來決定的。對於以往企業僅以經驗法則來評斷其上下游工廠（縱向供應鏈），或是具有相同製程能力的工廠（橫向供應鏈），的訂單分配數量（或分配比率）。特別是競爭優勢較強的工廠往往在爭取訂單的能

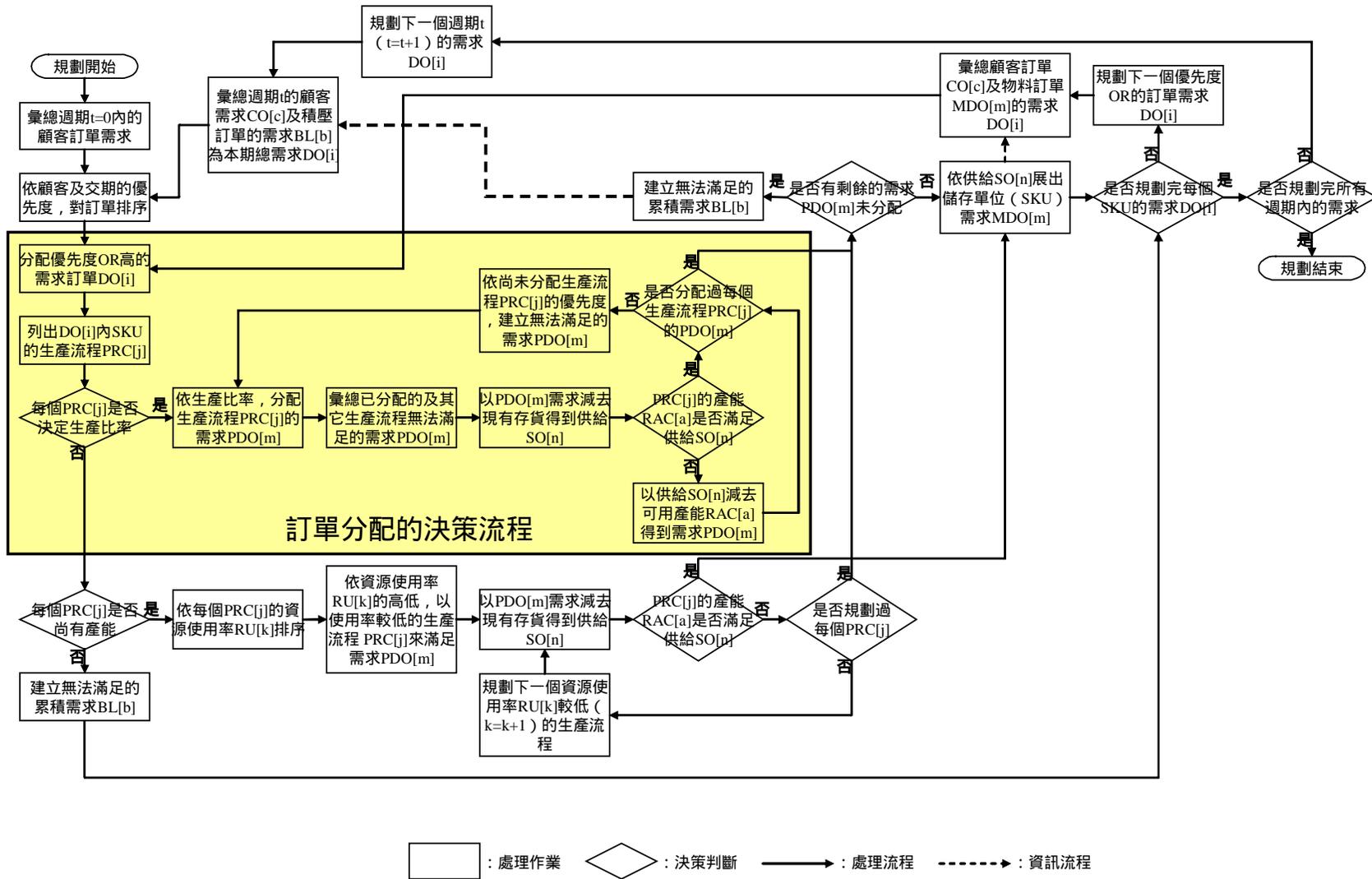


圖 3.9 多廠區生產系統下產銷平衡的規劃邏輯

力較其它工廠來得強勢，但是未必在成本考量上較為優勢。

有鑑於此，圖 3.9 的規劃邏輯主要以啟發性法則來決定不同生產流程間的訂單分配數量，並且假設該規劃邏輯所需要設定的基本參數皆能正確且有效地被取得，如生產流程的訂單分配比率，是根據經驗法則或事前評估成本效益（生產指標）且考量物料及產能狀況所獲得的。每張訂單都是針對廠區內品項（儲存單位）所開立的，所以訂單分配的方式是依此儲存單位的所有生產流程（製造、運輸、採購與儲存）所設定的分配比率配置訂單，進而獲得每個流程所需要製造、運輸、採購與儲存的數量。

以 SKU1 為例，若 SKU1 的生產流程包含製造、運輸與儲存流程（假設 SKU1 不是原物料，所以不考量採購流程），且依經驗法則或生產指標得到（同時考量物料及產能狀況）的分配比率分別為製造 60%，運輸 30%（由它廠運輸至該廠的 SKU1）與儲存 10%（由該廠的倉庫中取料）。假設某規劃週期內，SKU1 需要生產 10000 個，則依照分配比率推算製造流程需要自製 6000 個，運輸流程需要由它廠運送 3000 個，儲存流程需要從倉庫中取料 1000 個。進一步地，製造及運輸流程將推算所需要的輸入品項個數，而輸入品項也由各自的生產流程推算分配數量，依此類推。最後，規劃出預計生產 SKU1（10000 個）時，各個不同儲存單位的生產流程之分配數量為何？

圖 3.9 中的符號表示，與本論文所提出以產銷平衡為基礎的訂單分配模式的符號表示並無相關性，僅以方便為宜，以下說明規劃邏輯的運算步驟說明。

步驟一：彙總週期 $t=0$ 內的顧客需求（外部需求）為需求訂單（ $DO[i]$ ）

步驟二：依顧客及交期的優先度，對訂單排序，優先度較高較優先排。

優先度（OR）的計算方式是（顧客權重×顧客的重要性）+（交期權重×交期與規劃始初時間的差距），且顧客權重+交期權重= 0。

- 步驟三：分配優先度（OR）高的需求訂單（DO[i]）。
- 步驟四：列出（DO[i]）內 SKU 的生產流程（PRC[j]），包含製造、運輸、採購與儲存流程。
- 步驟五：判斷 PRC[j] 是否決定生產比率，生產比率是由生管人員依照經驗法則或評估生產指標所決定的分配比率。如果生管人員能夠決定生產比率，則進行步驟六；反之，則進行步驟十五。
- 步驟六：依生產比率，分配生產流程（PRC[j]）的內部需求（PDO[m]），其內部需求的計算方式是 $PDO[m] = DO[i] \times PRC[j].rate$ 。
- 步驟七：除了步驟六所計算每個生產流程需要生產的需求外，尚有來自於其它相依性需求的物料需求量（MDO[m]），本步驟彙總已分配的及其它生產流程無法滿足的需求（PDO[m]）。
- 步驟八：以內部需求（PDO[m]）減去現有存貨（inventory）得到淨需求量，即該生產流程預計生產的供給量（SO[n]）。
- 步驟九：判斷生產流程 PRC[j] 所耗用資源的可用產能（RAC[a]）是否滿足供給（SO[n]），如果是的話，則進行步驟十；反之，則進行步驟十二。
- 步驟十：判斷是否分配過每個生產流程（PRC[j]）的內部需求（PDO[m]），如果有的話，則進行步驟十三；反之，則進行步驟十一。
- 步驟十一：依尚未分配生產流程（PRC[j]）的優先度，建立無法滿足的需求（PDO[m]， $m = m + 1$ ）。
- 步驟十二：以供給（SO[n]）減去可用產能（RAC[a]）得到未被滿足的需求量（ $PDO[m] = SO[n] - RAC[a]$ ）。
- 步驟十三：判斷是否有剩餘的需求（PDO[m]）未分配，如果有的話，則進行步驟十四；反之，則進行步驟二十二。
- 步驟十四：建立無法滿足的累積需求（BL[b]）。提供積壓訂單的生產資訊給步驟二十九，納入下週期（t+1）的規劃。
- 步驟十五：判斷每個生產流程 PRC[j] 是否尚有剩餘的可用產能，如果有的話，則進行步驟十六；反之，則進行步驟二十三。

- 步驟十六：依每個 $PRC[j]$ 的資源使用率 ($RU[k]$) 排序，資源使用率較低較優先排。由於以訂單為主的「產銷平衡」機制較難達成「提高資源使用率」的生產目標，所以排序法則主要是針對「提高資源使用率」的生產目標而設計的。
- 步驟十七：依資源使用率 ($RU[k]$) 的高低，以使用率較低的生產流程 ($PRC[j]$) 優先滿足需求 ($PDO[m]$)。
- 步驟十八：以內部需求 ($PDO[m]$) 減去現有存貨 (inventory) 得到淨需求量，即該生產流程預計生產的供給量 ($SO[n]$)。
- 步驟十九：判斷生產流程 $PRC[j]$ 所耗用資源的可用產能 ($RAC[a]$) 是否滿足供給 ($SO[n]$)，如果是的話，則進行步驟二十二；反之，則進行步驟二十。
- 步驟二十：判斷是否規劃過每個生產流程 ($PRC[j]$) 的內部需求 ($PDO[m]$)，如果有的話，則進行步驟十三；反之，則進行步驟二十一。
- 步驟二十一：規劃下一個資源使用率 ($RU[k]$) 較低 ($k=k+1$) 的生產流程。
- 步驟二十二：依供給 ($SO[n]$) 的輸出儲存單位 (output SKU) 展開相依性需求之輸入儲存單位 (input SKU) 的物料需求量 ($MDO[m]$)。
- 步驟二十三：建立無法滿足的累積需求 ($BL[b]$)。
- 步驟二十四：判斷是否規劃完每個 SKU 的需求 ($DO[i]$)，如果是的話，則進行步驟二十七；反之，則進行步驟二十五。
- 步驟二十五：規劃下一個優先度 (OR) 的訂單需求 $DO[i]$ ， $i = i + 1$ 。
- 步驟二十六：彙總外部需求之顧客訂單 ($CO[c]$) 及內部需求之物料訂單 ($MDO[m]$) 的需求 ($DO[i]$)。
- 步驟二十七：判斷是否規劃完所有週期內 ($t = 1, 2, \dots, T$) 的需求，如果是的話，則規劃結束；反之，則進行步驟二十八。
- 步驟二十八：規劃下一個週期 t ($t = t + 1$) 的需求 ($DO[i]$)。
- 步驟二十九：彙總週期 t 的外部需求之顧客訂單 ($CO[c]$) 及來自於

步驟十四及二十三的積壓訂單之需求 (BL[b]) 為該期總需求 (DO[i])。

藉由以產銷平衡為基礎的規劃邏輯，能夠決定多廠區的生產流程需要生產之訂單投入或產出量，而圖 3.9 中「訂單分配的決策流程」就是訂單分配的主要決策流程。由於生產流程的訂單分配比率，是根據經驗法則或事前評估成本效益（生產指標）且考量物料及產能狀況所獲得的，而沒有一個有效且精確的參考準則可以提供給生管人員做決策依據。因此，本論文提出一以產銷平衡為基礎的訂單分配模式，用以彌補上述的缺點。

3.3. 產銷平衡的訂單分配模式之建立

3.3.1. 假設條件說明

本論文所提出的訂單分配模式主要應用於「薄膜電晶體液晶顯示器 (Thin Film Transistor-Liquid Crystal Display ; TFT-LCD)」產業的多廠區生產系統，配合第 3.1 節以流程導向的建模方法與成本計算方式建構數學模式。基於台灣液晶顯示器產業的特殊性，與以產銷平衡為基礎的訂單分配模式之應用範圍，提出以下基本假設：

- (1) 假設所有本模式需要計算成本公式的資料來源及其相關的生產資訊，皆能正確且有效地估算。
- (2) 本模式採用中央式規劃方法 (centralized multi-site planning)，應用於同一企業內或生產資訊能夠完全容易取得的多廠區生產系統。
- (3) 由於本模式著重於企業內部的多廠區生產規劃，不考量企業外部的採購問題 (如供應商的選擇)，所以本模式採用採購成本併入運輸成本的方式，簡化訂單分配的複雜度。同時，假設來自於企業外部的原物料、組件等品項是無限量供給的，也就是無缺料的問題。

- (4) 假設本模式以供給驅導 (push) 方式來規劃物料消耗的問題。換句話說, 就是來源廠 (source) 所提供的物料供給必須大於消耗廠 (consumption) 的物料需求, 所以不考量物料的短缺成本與經濟訂購量、經濟製造批量等問題。
- (5) 大部份業界都採用計算安全存貨的方式, 是以採購前置時間乘經濟訂購量, 也就是採購前置時間內的備料數量。換句話說, 其安全存貨乃是固定成本, 所以本模式不考量安全存貨的成本。
- (6) 對於液晶顯示器產業而言, 某規劃週期內未依計劃投料的積壓訂單, 將由生產人員根據實際需求決定是否納入下一規劃週期內做規劃。因此, 本模式不考慮積壓訂單所造成的損失成本。
- (7) 假設多廠區生產環境是屬於穩定式生產的狀態, 所以生產函式為一離散均勻分配 (discrete uniform distribution) 函式。同時, 該環境的產品定位策略 (Product Positioning Strategy; PPS) 是屬於計畫式生產 (Make To Stock; MTS) 的多廠區生產系統。

3.3.2. 符號說明

下標的定義

- t : 規劃週期 ($t = 0, 1, 2, \dots, T$)。
- i : 品項 (泛指物料、零組件或最終產品)。
- p : 產品尺寸 (1: 15 吋, 2: 17 吋)。
- s : 所有工廠 (包括 v : 來源廠, c : 消耗廠)。
- v : 來源 (供應) 廠。
- c : 消耗廠。
- m : 子品項 i 的上一階母品項。
- r : 被挪移訂單之週期 ($r = 0, 1, 2, \dots, T$)。

決策變數的定義

- $u_{isp,t}$: 週期 t 內, 工廠 s 製造出尺寸 p 的品項 i 之投入訂單量。
- $q_{isc,t}$: 週期 t 內, 從工廠 s 運送品項 i 至消耗廠 c 的出貨量。

相關變數的定義

$j_{is,t}$ ：週期 t 末，在工廠 s 之品項 i 的存貨量。

$e_{ip,t}$ ：週期 t 內，製造尺寸 p 的品項 i 之填補產能訂單量。

$b_{ip,t}$ ：週期 t 末，製造尺寸 p 的品項 i 之積壓訂單量。

$f_{rip,t}$ ：週期 r 內，製造尺寸 p 的品項 i 之填補產能訂單是否挪移至週期 t 配置給該期之積壓訂單消耗？（0：否；1：是）

目標函式中常數的定義

$MPC_{isp,t}$ ：週期 t 內，在工廠 s 製造出尺寸 p 的品項 i 之每單位製造成本。

$TPC_{isc,t}$ ：週期 t 內，從工廠 s 運送品項 i 至消耗廠 c 之每單位運輸/採購成本。

$SPC_{is,t}$ ：週期 t 末，在工廠 s 的品項 i 之每單位存貨成本。

常數的定義

$UT_{sc,t}$ ：週期 t 內，從工廠 s 運輸至消耗點 c 的最大運輸量。

$LT_{sc,t}$ ：週期 t 內，從工廠 s 運輸至消耗點 c 的最小運輸量。

$UM_{sp,t}$ ：週期 t 內，工廠 s 製造尺寸 p 的最大製造量（包含製造總量上限與製造分量上限）

$RA_{s,t}$ ：週期 t 內，工廠 s 的總投入量。

$DO_{ip,t}$ ：週期 t 初，消耗廠 c 中品項 i 的需求量。

$RS_{isp,t}$ ：週期 t 內，在工廠 s 製造出尺寸 p 的品項 i 之分級率。

$RL_{is,t}$ ：週期 t 內，在工廠 s 製造出品項 i 之良率。

CT_{is} ：工廠 s 內製造出品項 i 的平均製造週期（天）

MT_{isc} ：從工廠 s 運送品項 i 至消耗廠 c 的平均運輸時間（天）

DT_t ：規劃週期 t 內的總時間單位（天）數。

MM_{im} ：每單位製造品項 m 所需要物料 i 的數量。

IX ：週期 $t = 0$ 末，在工廠 s 之品項 i 的存貨量。

IY ：週期 $t = 0$ 內，工廠 s 製造出尺寸 p 的品項 i 之投入訂單量。

IZ ：週期 $t = 0$ 內，從工廠 s 運送品項 i 至消耗廠 c 的出貨量。

生產函式的定義

$T_1(x)$ ：計算上週期投料於本週期產出量的函式。

$T_2(x)$ ：計算本週期投料於本週期產出量的函式。

$S_1(x)$ ：計算上週期出貨於本週期收貨量的函式。

$S_2(x)$ ：計算本週期出貨於本週期收貨量的函式。

3.3.3. 目標函式說明

$$\begin{aligned} \text{Min.} \quad & \sum_t \sum_i \sum_s \sum_p (MPC_{isp,t} \times u_{isp,t}) + \sum_t \sum_i \sum_s \sum_c (TPC_{isc,t} \times q_{isc,t}) + \\ & \sum_t \sum_i \sum_s (SPC_{is,t} \times j_{is,t}) \end{aligned}$$

本論文所提出之訂單分配模式的目標函式是最小化所有規劃週期內，各廠區內與廠際間的總生產成本（製造、運輸與儲存）。

3.3.4. 限制條件說明

(1) 運輸供給的限制式

$$\sum_i q_{isc,t} \leq UT_{sc,t}, \forall t, s, c \dots\dots\dots (\text{公式 3.1})$$

$$\sum_i q_{isc,t} \geq LT_{sc,t}, \forall t, s, c \dots\dots\dots (\text{公式 3.2})$$

(2) 資源供給的限制式

$$\sum_i \sum_p u_{isp,t} = RA_{s,t}, \forall t, s \dots\dots\dots (\text{公式 3.3})$$

(3) 供給數量的限制式

$$\sum_i u_{isp,t} \leq UM_{sp,t}, \forall t, s, p \dots\dots\dots (\text{公式 3.4})$$

(4) 產銷平衡的限制式

$$DO_{ip,t} = \begin{cases} \sum_s u_{isp,t} - e_{ip,t} & \text{if } DO_{ip,t} \leq \sum_s u_{isp,t} \\ \sum_s u_{isp,t} + b_{ip,t} & \text{if } DO_{ip,t} > \sum_s u_{isp,t} \end{cases}, \forall t, i, p \dots \text{ (公式 3.5)}$$

$$\text{Min.} \left\{ \sum_{x=0}^T (f_{xip,t} \times e_{xip}) - b_{ip,t} \right\} \geq 0 \text{ if } \exists f_{rip,t} = 0, \forall t, i, p \dots \text{ (公式 3.6)}$$

(5) 物料需求的限制式

$$j_{is,t} = j_{is,t-1} + \sum_p T_1(u_{isp,t-1}) + \sum_p T_2(u_{isp,t}) + \sum_v S_1(q_{ivs,t-1}) \\ + \sum_v S_2(q_{ivs,t}) - \sum_c q_{isc,t} - \sum_{\substack{m \text{ s.t. } i \text{ is} \\ \text{a material} \\ \text{of } m}} \sum_p (MM_{im} \times u_{msp,t}), \forall t, i, s \dots \text{ (公式 3.7)}$$

其中

$$T_1(u_{isp,t-1}) = \lfloor (CT_{is} / DT_{t-1}) \times RS_{isp,t-1} \times RL_{is,t-1} \times u_{isp,t-1} \rfloor \\ T_2(u_{isp,t}) = \lfloor ((DT_t - CT_{is}) / DT_t) \times RS_{isp,t} \times RL_{is,t} \times u_{isp,t} \rfloor \\ S_1(q_{ivs,t-1}) = \lfloor (MT_{isc} / DT_{t-1}) \times q_{ivs,t-1} \rfloor \\ S_2(q_{ivs,t}) = \lfloor ((DT_t - MT_{isc}) / DT_t) \times q_{ivs,t} \rfloor$$

(6) 初始值與相關變數的條件式

$$j_{is,0} = IX, \forall i, s \quad q_{isc,0} = IZ, \forall i, s, c \\ u_{isp,0} = IY, \forall i, s, p \quad u_{isp,t} \geq 0, \forall t, i, s, p \\ j_{is,t} \geq 0, \forall t, i, s \quad q_{isc,t} \geq 0, \forall t, i, s, c \\ e_{ip,t} \geq 0, \forall t, i, p \quad \sum_t f_{rip,t} \leq 1, \forall r, i, p \\ b_{ip,t} \geq 0, \forall t, i, p$$

第 1 個運輸供給的限制式 (公式 3.1) 及 (公式 3.2) 是不考慮將運輸作業交由企業外部的物流中心處理的條件下, 同一企業下的運輸資源是有限的, 所以來源廠運輸物料至消耗廠的總運輸量不能夠大於最大運輸量, 且基於經濟批量的原則, 其總運輸量不能夠小於最小運輸量。

對於液晶顯示器產業而言, 購買設備的成本遠大於儲存物料的成本

本，也就是生產目標主要著重於提高產能利用率。因此，為了充分利用產能，除了需要開立填補產能的訂單之外（第 4 個限制式），所以第 2 個資源供給的限制式是限制每個廠區的總投入訂單量必須等於其可投入的訂單總量。

第 3 個供給數量的限制式（公式 3.3）主要為了避免生產現場太集中於重複製造相同品項的訂單，進而延誤製造其它品項的訂單之生產進度，所以液晶顯示器產業通常以天為單位，對生產工廠內某些品項及全部品項的投入量加以限制，此又稱為分量限制及總量限制。如圖 3.10 所示，多廠區生產規劃系統的規劃週期是以月為單位，但是分量限制及總量限制大多針對各個廠區以日為單位的生產規劃。為了有效落實多廠區生產規劃系統所規劃結果，且能確實執行於單廠區生產規劃系統，將以日為單位的分量限制彙總成以月為單位的分量限制，以日為單位的總量限制彙總成以月為單位的總量限制。

製造品項的日總 / 分量限制彙總成月總 / 分量限制

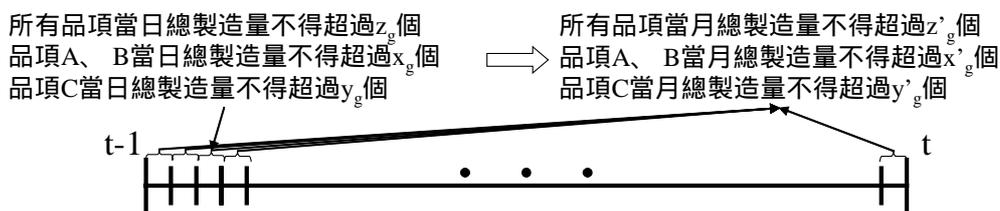


圖 3.10 供給數量限制的示意圖

對於液晶顯示器產業而言，投資於機器設備上的成本遠大於物料儲存的成本，所以「提高資源使用率」為其生產目標之一。因此，若供給大於需求時，為了提高資源使用率，則開立填補產能的訂單；若供給小於需求時，造成預計投料而未投料的情形，則對未投料的訂單數量開立積壓訂單，如圖 3.11 所示。由於生管人員對於每一規劃週期內所開立的積壓訂單將進行進一步地評估，決定是否納入下一個規劃週期內做規劃，所以第 4 個產銷平衡的限制式（第一個公式）僅提供積壓訂單的生產資訊，不將其納入下一個規劃週期內做規劃。為了避免開立過多的填補產能訂單，造成生產現場過多的在製品或成品（end item），基於訂單提前（pull）與延遲（push）生產的概念，本論文進而採用訂單挪移的機制（如圖 3.12 所示）解決此問題（第二個公式），其特點在於將它期末投料的積壓訂單以填補產能訂單消耗（consumption）。優點是減少產銷協調間的衝突發生，同時提供生管人員執行生產規劃的參考依據（如表 3.1），功用是避免以主觀的經驗法則造成供給與需求間的差距波動太大，形成供應鏈上的長鞭效應。

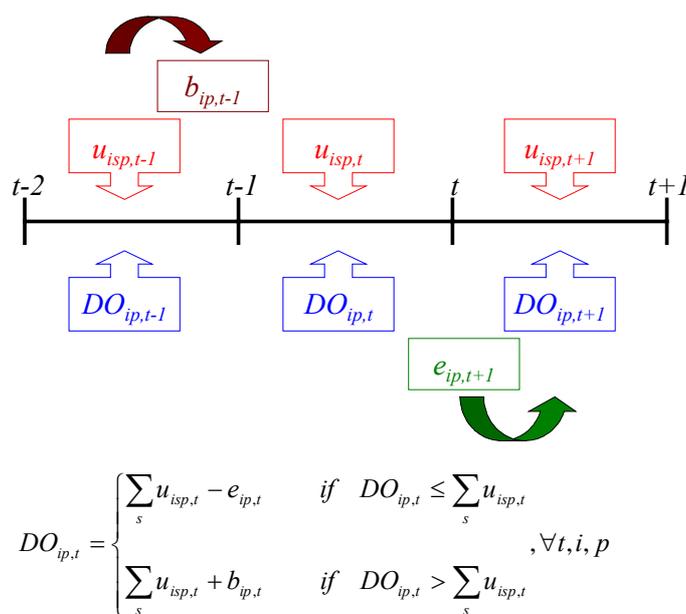
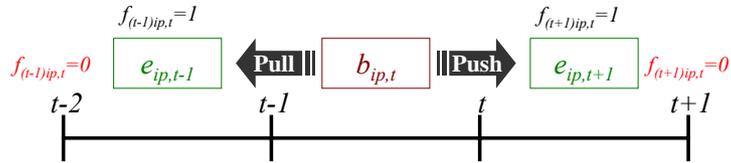


圖 3.11 積壓訂單與填補產能訂單的示意圖



$$\text{Min.} \left\{ \sum_{x=0}^T (f_{xip,t} \times e_{xip}) - b_{ip,t} \right\} \geq 0 \quad \text{if } \exists f_{rip,t} = 0, \forall t, i, p$$

圖 3.12 供給與需求平衡的示意圖

表 3.1 積壓訂單與填補產能訂單間的挪移

	$b_{31,1}$	$b_{32,1}$	$b_{31,2}$	$b_{32,2}$	$b_{31,3}$	$b_{32,3}$
$e_{31,1}$	*	*	$f_{131,2}$	*	$f_{131,3}$	*
$e_{32,1}$	*	*	*	$f_{132,2}$	*	$f_{132,3}$
$e_{31,2}$	$f_{231,1}$	*	*	*	$f_{231,3}$	*
$e_{32,2}$	*	$f_{232,1}$	*	*	*	$f_{232,3}$
$e_{31,3}$	$f_{331,1}$	*	$f_{331,2}$	*	*	*
$e_{32,3}$	*	$f_{332,1}$	*	$f_{332,2}$	*	*

第 5 個物料需求的限制式是計算每一個規劃週期末各個廠區內品項的存貨量，主要考量上一個規劃週期末的存貨，加上本規劃週期內所製造與運輸的產出量，減去本規劃週期內該品項被其它品項所消耗的投入量及運輸至它廠的數量。其中， $T_1(x)$ 表示計算上週期投料於

本週期產出量的函式； $T_2(x)$ 表示計算本週期投料於本週期產出量的函式； $S_1(x)$ 表示計算上週期出貨於本週期收貨量的函式； $S_2(x)$ 表示計算本週期出貨於本週期收貨量的函式。圖 3.13 是物料投入與產出關係的示意圖。由於製造前置時間（平均製造週期）與運輸前置時間（平均運輸時間）的遞延情況，造成上一個規劃週期內所投入的生產或運輸數量，將於該規劃週期內產出或送達該廠區，也就是當投料日或出貨量小於製造或運輸前置時間的話，其產出量會造成次一個規劃週期的存貨量。

圖 3.14 是各個廠區間的運輸關係，以（來源廠 v ，消耗廠 c ）的符號表示由來源廠將物料運送至消耗廠的配送關係。以「薄膜電晶體製造廠（ $s = 1$ ）把薄膜電晶體運送至單元製造廠（ $s = 3$ ）」為例，運輸關係就是以（1, 3）表示。

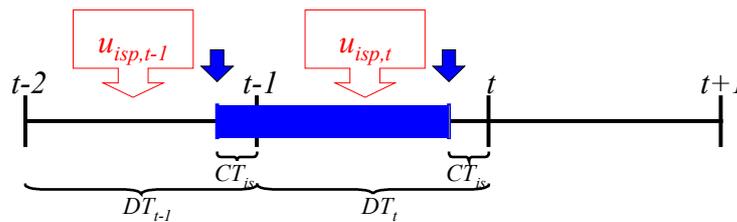


圖 3.13 物料投入與產出關係的示意圖

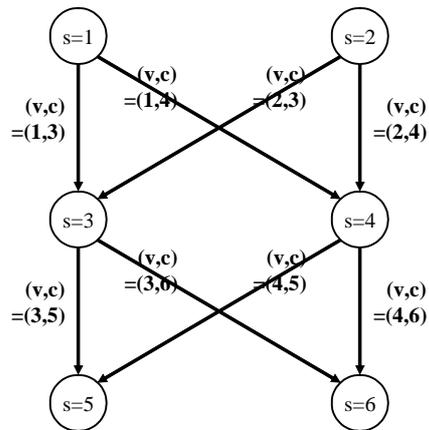


圖 3.14 多廠區生產系統的網路結構圖

第 6 個初始值與相關變數的條件式是每個決策變數的非負整數與相關限制，及初始狀態的設定值。

為了有效求解上述的訂單分配模式，本論文將於第四章運用基因演算法之最佳化搜尋的特性，同時考量限制條件滿足技術利用縮小搜尋解空間 (solution space)，進一步發展出能快速求解且兼具有最佳化搜尋能力的訂單分配演算法。

第四章 以限制條件為基礎之基因演算法的設計與應用

本章介紹以限制條件為基礎之基因演算法的設計與應用。首先，第 4.1 節針對問題的特性，決定染色體的表示法；接著，第 4.2 節說明加入懲罰策略的適應性函式；第 4.3 節分別描述突變、交配、複製與疊置等基因運算子的設計原則；最後，第 4.4 節結合第 4.3 節基因運算子的運算流程，說明以限制條件為基礎之基因演算法的運算流程。

4.1. 染色體的表示法

基因編碼可分為二進位編碼及實數編碼兩類。對二進位編碼而言，當系統參數增多時，編碼及解碼需要執行大量的浮點數運算。在每一代的演化過程，適應函式的計算都需要進行解碼運算，將耗費電腦資源，降低搜尋速度，且染色體長度的長短，影響解準確度的高低，如染色體長度不足，可能導致搜尋不到系統整體最佳解，染色體長度太長，則降低搜尋速度。因此，本論文將採用實數編碼基因演算法 (Real-coding Genetic Algorithm ; RGA) 的方式，可確保解空間的任一狀態點都能對應。

針對本論文所提出的訂單分配模式，主要是求解於 $u_{isp,t}$ 與 $q_{isc,t}$ 二個決策變數，則其它決策變數是此二個決策變數之值與相關限制式所決定的。因此，基因演算法中染色體的表示法，如表 4.1 所示。

表 4.1 染色體的表示法

基因	$u_{isp,t}$	$u_{isp,t}$		$u_{isp,t}$	$q_{isc,t}$	$q_{isc,t}$		$q_{isc,t}$
基因值	X_1	X_2		X_N	Y_1	Y_2		Y_M

接下來各小節將分別介紹以限制條件為基礎的基因演算法 (Constraints-Based Genetic Algorithm ; CBGA) 之設計原則，特別針

對以產銷平衡為基礎的訂單分配模式中，分別對硬性（hard）與軟性（soft）限制條件的處理方式詳加描述。最後，第 4.4 節總結各小節針對基因演算法所具備的運作條件（包含定義適應性函式，突變、交配與複製運算子），描述處理限制條件之基因演算法的整體運作流程及特性。

4.2. 適應性函式說明

本論文對軟性限制條件（soft constraints）的處理方法，採用建構懲罰策略之加法形式（addition form）的評估函式，表示如下：

$$eval(x) = f(x) + p(x)$$

x 表示染色體； $f(x)$ 是問題的目標函式； $p(x)$ 是懲罰函式。而 $p(x)$ 是採用違反限制式的懲罰數量且與問題獨立（problem independent）的懲罰函式。

懲罰策略的優點在於資訊保留與選擇壓力之間取得一個平衡點，也就是化解「保留一些不合理的」與「拒絕一些不合理的」之間的衝突[13]。由於本論文所提出的訂單分配模式是屬於最小化的問題，則

$$\begin{aligned} p(x) &= 0, & \text{if } x \text{ is feasible} \\ p(x) &> 0, & \text{otherwise} \end{aligned}$$

才能避免負的適應值（negative fitness value）。

4.3. 基因運算子的設計原則

本論文對硬性限制條件（hard constraints）的處理方法，應用 Handa[17][18]所提出解決限制條件滿足問題的方法，以基原為基礎運用疊置（superposition）運算子求解，使得整個運作流程能快速收斂。藉此概念，應用疊置運算子結合互換突變、均勻交配與排序複製運算子的規劃邏輯中，用以滿足硬性限制條件的求解過程，期望達成增進

搜尋速度。以下分別說明各個運算子的規劃邏輯。

(1) 互換 (reciprocal exchange) 突變運算子

互換突變運算子是隨機選取二個位置，然後，交換這二個位置上的基因值。其運算步驟說明如下：

步驟一：從族群中選擇一個染色體 $gene[a]$ ，初始值 $a = 0$ 。

步驟二：隨機產生該染色體的突變值 m ， m 值介於包含 0 而不包含 1 的區間。

步驟三：判斷 m 值是否小於突變率（是否進行突變的運算程序），如果是的話，則進行步驟四；反之，則進行步驟六。

步驟四：產生一隨機值設定為第一個互換點的位置 i 。

步驟五：產生一隨機值設定為第二個互換點的位置 i 。

步驟六：判斷族群中每個染色體是否都被選出來（ $a == pop_size?$ ），如果是的話，則規劃結束；反之，則進行步驟七。

步驟七：選擇下一個染色體（ $++a$ ）。

步驟八：判斷第一個與第二個互換點的位置是否重複（ $i == j?$ ），如果是的話，則進行步驟五；反之，則進行步驟九。

步驟九：進行突變程序，互換位置 i, j 上的基因值。

步驟十：從限制條件池（pool）找出所有硬性限制條件的基原。

步驟十一：尋找與基因相關之基原 $hc[k]$ ，初始值 $k = 0$ 。

步驟十二：將突變後的染色體 $gene[a]$ 比對基原 $hc[k]$ 的硬性限制。

步驟十三：判斷染色體 $gene[a]$ 是否滿足硬性限制，如果是的話，則進行步驟十四；反之，則進行步驟十六。

步驟十四：判斷染色體 $gene[a]$ 是否都比對過基原 $hc[k]$ 的硬性限制（ $k == h_cst_size?$ ），如果是的話，則進行步驟六；反之，則進行步驟十五。

步驟十五：選擇下一個基原 $hc[k]$ 的硬性限制（ $++k$ ）。

步驟十六：以硬性限制重新產生合理的隨機值代換非法的基因值。

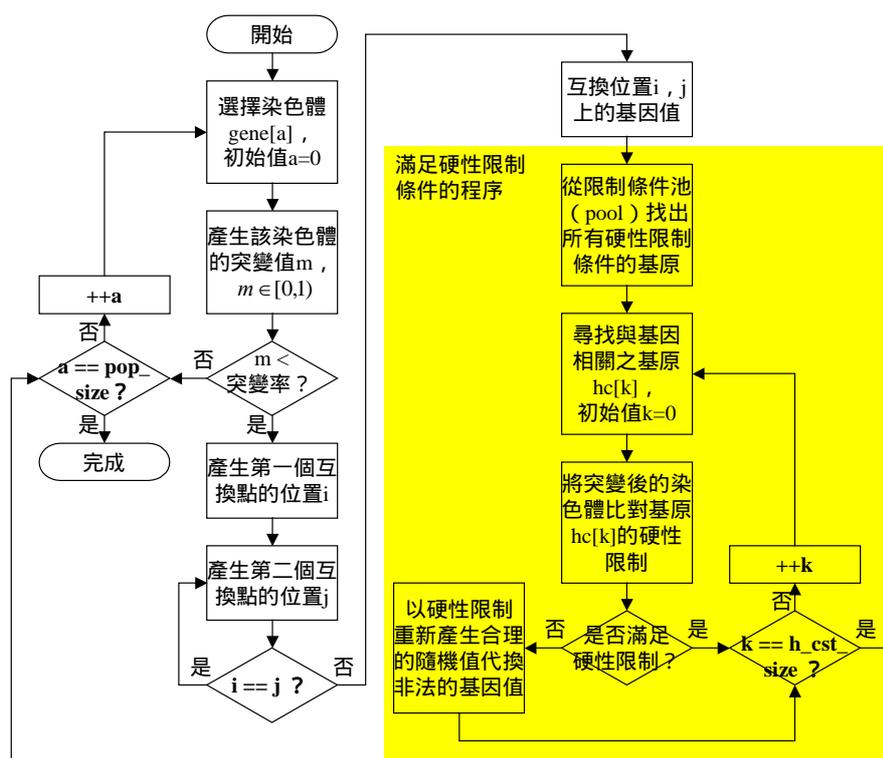


圖 4.1 互換突變運算子的運算邏輯

其運算邏輯及運作範例如圖 4.1 與圖 4.2 所示。假設圖 4.2 中進行突變的染色體隨機選中第 3 個及第 7 個位置（起始位置為 0）的基因進行「互換突變」，將二位置的基因值互換。接著，從限制條件池中找出與第 3 個及第 7 個位置所代表的決策變數相關之硬性限制條件的基原（問號代表基因值是任意值，星號代表基因值是受到硬性限制條件所限制的，可能本身所造成的，或是受到其它基因所影響的），進而與突變後的染色體「比對硬性限制條件的基原」（即進入疊置運算子的運作流程），如果違反硬性限制條件的話，則「修正違反硬性限制條件的基因」對第 7 個基因的限制範圍重新產生基因值。否則，則不進行「修正違反硬性限制條件的基因」的程序。

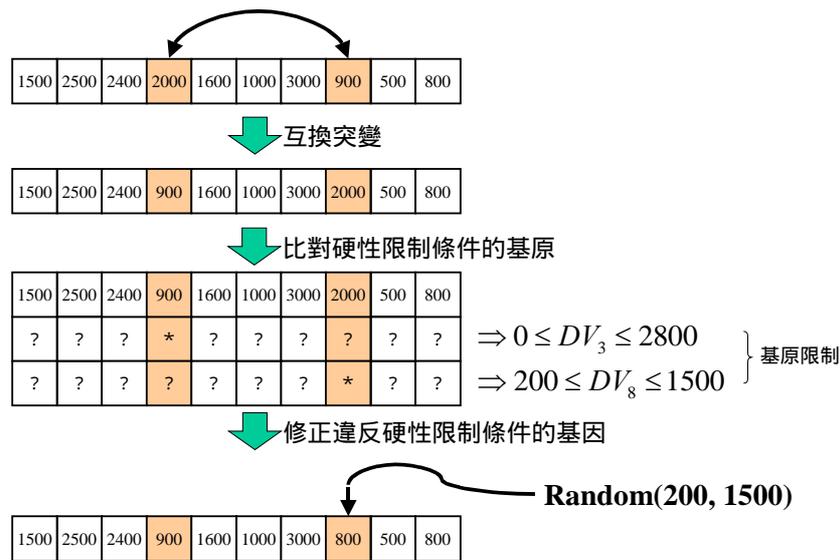


圖 4.2 互換突變運算子的運算程序

(2) 均勻 (uniform) 交配運算子

均勻交配運算子是針對染色體中的每一個基因，以類似擲銅板的方式來決定該基因是否互換。簡言之，對每一個基因而言，其互換的機率皆為 50%。通常採用均勻交配運算子之前，會對每一個基因隨機產生 0 或 1 的整數，0 表示此基因不用互換，1 表示此基因要互換，而此 0、1 的值稱為遮罩值。其運算步驟說明如下：

步驟一：從族群中選擇一個染色體 $gene[a]$ ，初始值 $a = 0$ 。

步驟二：隨機產生該染色體的交配值 n ， n 值介於包含 0 而不包含 1 的區間。

步驟三：判斷 n 值是否小於交配率（是否進行交配的運算程序），如果是的話，則進行步驟四；反之，則進行步驟六。

步驟四：累計可交配染色體的個數（ $d++$ ，初始值 $d = 0$ ）。

步驟五：將該染色體 $gene[a]$ 放入交配池 $cp[d]$ 。

步驟六：判斷族群中每個染色體是否都被選出來（ $a == pop_size?$ ），如果是的話，則進行步驟八；反之，則進行步驟七。

- 步驟七：選擇下一個染色體 ($++a$)。
- 步驟八：判斷可交配染色體的個數是否為偶數 (是否配對)，如果是的話，則進行步驟十；反之，則進行步驟九。
- 步驟九：刪除最後一個可交配的染色體 ($cp[d], d--$)。
- 步驟十：選取最後二個 ($cp[d]$ 和 $cp[(d-1)]$) 染色體 x 和 y 進行均勻交配 ($d = d - 2$)。
- 步驟十一：依均勻機率分配函式，隨機產生染色體 x 和 y 相對位置 i 的遮罩值 b ，其值為 0 或 1 ($b = \{0|1\}$)，初始值 $i = 0$ 。
- 步驟十二：判斷遮罩值是否為 1 ($b == 1?$)，即染色體 x 和 y 的相對位置 i 是否可以進行交配程序，如果是的話，則進行步驟十三；反之，則進行步驟十五。
- 步驟十三：進行交配程序，將 x 和 y 的相對位置 i 上之基因值互換。
- 步驟十四：進行滿足硬性限制條件的程序。此步驟的運算程序如同「互換突變運算子」的運算步驟說明中，第十至第十六步驟的運算邏輯。
- 步驟十五：判斷每個相對位置 i 是否都產生過遮罩值 b ($cp[0].length() == (i + 1)?$)，如果是的話，則進行步驟十七；反之，則進行步驟十六。
- 步驟十六：前進一個相對位置 ($++i$)。
- 步驟十七：判斷交配池是否是空的 ($d == -2?$)，即已無染色體進行交配，如果是的話，則規劃結束；反之，則進行步驟十。

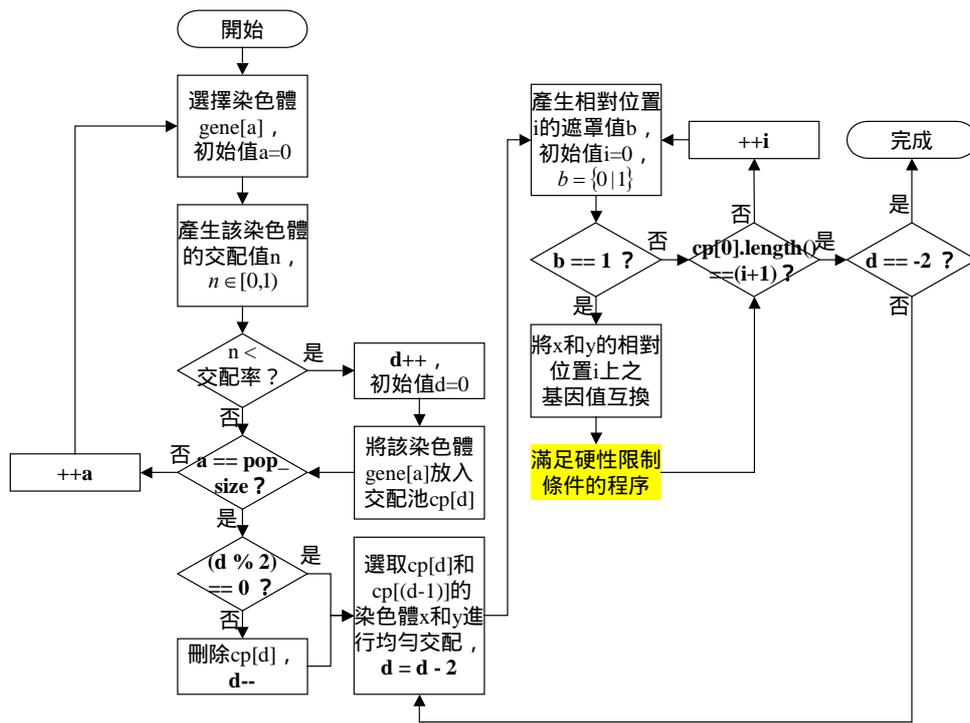


圖 4.3 均勻交配運算子的運算邏輯

其運算邏輯及運作範例如圖 4.3 與圖 4.4 所示。圖 4.4 中針對進行交配的二個母代產生對應的遮罩值，符合交配條件者（遮罩值為 1）則進行「均勻交配」，所以二個母代中第 1、3、5、8 個位置的基因互換其基因值。接著，從限制條件池中找出與第 1、3、5、8 個位置所代表的決策變數相關之硬性限制條件的基原，進而與交配後的染色體「比對硬性限制條件的基原」（即進入疊置運算子的運作流程），如果違反硬性限制條件的話，則「修正違反硬性限制條件的基因」。否則，則不進行「修正違反硬性限制條件的基因」的程序（本例假設不違反硬性限制條件，所以不進行「修正違反硬性限制條件的基因」的程序）。



圖 4.4 均勻交配運算子的運算程序

(3) 排序 (rank) 複製運算子

排序複製運算子是將族群中所有染色體依適應值從小至大加以排序。適應值最低的染色體其排序值為 1，適應值最高的染色體其排序值等於當時族群的總染色體數。然後，再以排序值做為輪盤法中機率的大小，取代了直接以適應值做為機率大小的依據。換句話說，排序複製運算子是以染色體在整個族群中的相對位置為依據，而不再以適應值的大小做為選擇母代的直接依據。因此，不管族群中染色體適應值的分佈情形，其最好染色體被選中的機率永遠是最壞染色體的 100 倍。其運算步驟說明如下：

步驟一：計算目前母代的染色體數 e 。

步驟二：判斷 $((pop_size * 2) > e ?)$ ，如果是的話，則進行步驟三；反之，則規劃結束。

步驟三：計算複製數 $(f = 2 * pop_size - e)$ 。由於本論文所提出基因演算法為簡單型，所以運作機制為產生子代的母代之染色體個數必須是族群個數的兩倍，也就是經突變與交配運算程序後，母代將進行複製運算程序，進而滿足母代之染色體數等

於「兩倍族群個數」的條件。

步驟四：依適應值由小至大的方式將母代的染色體 ($gene[a]$) 重新排序，此適應值是由簡單型基因演算法的運作邏輯中，第四與第八步驟所計算得來的。

步驟五：設定排序值 ($f[a] = e$)，初始值 $a = 0$ 。

步驟六：選擇下一個染色體 ($++a$)。

步驟七：判斷每個染色體是否都設定過排序值 ($a == e?$)，如果是的話，則進行步驟八；反之，則進行步驟五。

步驟八：計算排序值之總和 ($s = (1 + e) * e / 2$)。

步驟九：計算適應率 ($fr[a] = f[a] / s$)，初始值 $a = 0$ 。

步驟十：選擇下一個染色體 ($++a$)。

步驟十一：判斷每個染色體是否都計算過適應值 ($a == e?$)，如果是的話，則進行步驟十二；反之，則進行步驟九。

步驟十二：計算累積適應率 ($fr[a] = fr[a] + fr[(a + 1)]$)，初始值 $a = 1$ 。

步驟十三：選擇下一個染色體 ($++a$)。

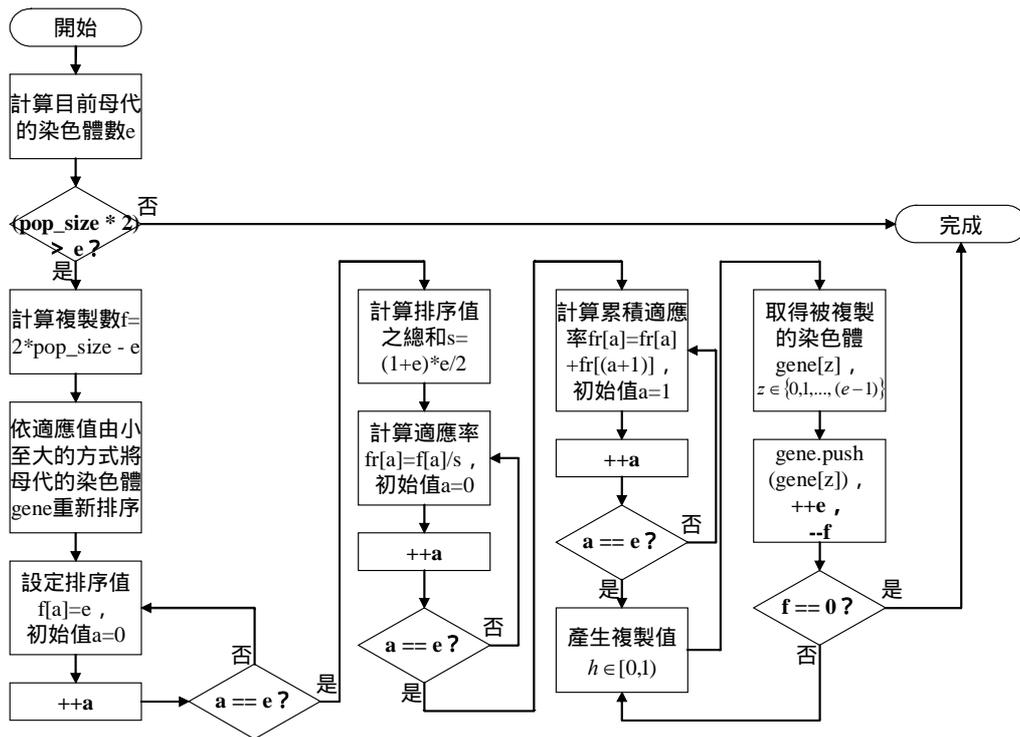
步驟十四：判斷每個染色體是否都計算過累積適應率 ($a == e?$)，如果是的話，則進行步驟十五；反之，則進行步驟十三。

步驟十五：隨機產生複製值 h ， h 值介於包含 0 而不包含 1 的區間。

步驟十六：複製累積適應率 ($fr[a]$) 小於 h 值且 a 值最小的染色體 $gene[z]$ ， $z \in \{0, 1, \dots, (e-1)\}$

步驟十七：將複製後的染色體插入母代中，且 ($gene.push(gene[z])$)， $++e$ ， $--f$)。

步驟十八：判斷母代之染色體數是否等於「兩倍族群個數」的條件 ($f == 0?$)，如果是的話，則規劃結束；反之，則進行步驟十五。



$$z = \min\{a; h < fr[a] | a = 0, 1, \dots, (e-1)\}$$

圖 4.5 排序複製運算子的運算邏輯

其運算邏輯及運作範例如圖 4.5 與圖 4.6 所示。假設 C_1 、 C_2 、 C_3 、 C_4 、 C_5 、 C_6 與 C_7 等七個染色體分別計算出適應值為 1180、1210、1230、1380、1180、1050 與 1320，經過「排序」後得到圖 4.6 中的表格。表格中染色體依小至大排序（假設求解問題為目標函式最小化），得到 C_6 、 C_1 、 C_5 、 C_2 、 C_3 、 C_7 與 C_4 等順序並且分別給予排序值 7、6、5、4、3、2 與 1，其排序值總和為 28，所以可以求得每個染色體的適應率（排序值÷排序值總和）。計算出各個染色體適應率後，以均勻分配的概念劃分各個染色體的適應範圍，再隨機產生亂數值，由亂數值所落入的範圍決定哪個染色體是被複製或選擇至下一代中，並且重複此程序直至滿足複製或選擇個數為止。



以適應率來選擇存活至下一代的染色體

圖 4.6 排序複製運算子的運算程序

(4) 疊置 (superposition) 運算子

Handa 等與陳氏[53]皆應用基原概念求解於限制條件滿足問題，且在問題求解的品質上都有不錯的表現，所以本論文採用相同概念應用於處理硬性限制條件的運算子，稱為疊置運算子，其運作範例如圖 4.7 所示。疊置運算子的運作流程如同互換突變運算子與均勻交配運算子中的「滿足硬性限制條件的程序」，其運算邏輯如圖 4.1 中「滿足硬性限制條件的程序」，而運算步驟說明如下：

- 步驟一：從限制條件池 (pool) 找出所有硬性限制條件的基原。
- 步驟二：尋找與基因相關之基原 $hc[k]$ ，初始值 $k = 0$ 。
- 步驟三：將突變後的染色體 $gene[a]$ 比對基原 $hc[k]$ 的硬性限制。
- 步驟四：判斷染色體 $gene[a]$ 是否滿足硬性限制，如果是的話，則進行下一個步驟(互換突變運算子或均勻交配運算子的運算步驟); 反之，則進行步驟七。
- 步驟五：判斷染色體 $gene[a]$ 是否都比對過基原 $hc[k]$ 的硬性限制($k == h_cst_size ?$), 如果是的話，則進行下一個步驟 (互換突變

運算子或均勻交配運算子的運算步驟); 反之, 則進行步驟四。

步驟六: 選擇下一個基原 $hc[k]$ 的硬性限制 ($++k$)。

步驟七: 以硬性限制重新產生合理的隨機值代換非法的基因值。

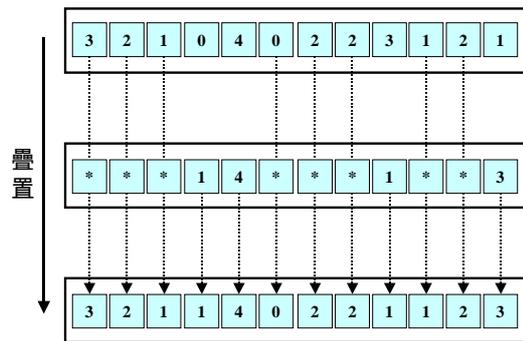


圖 4.7 疊置運算子的運算程序

4.4. 以限制條件為基礎之基因演算法的運作流程

本論文分別針對必須完全完全遵守而不能違反的「硬性限制條件」與可不遵守但必須付出相對成本的「軟性限制條件」, 提出不同的處理方法。處理硬性限制條件方面, 應用疊置運算子結合基原概念, 將硬性限制納入突變、交配與複製運算子的運算程序中, 將搜尋點空間縮小至完全遵守硬性限制條件的解空間, 避免浪費不必要的搜尋過程, 進而加快搜尋速度; 處理軟性限制條件方面, 採用建構懲罰策略之加法形式的評估函式, 避免基因演算法於搜尋過程中過早收斂而落入區域解空間。以限制條件為基礎之基因演算法的運算步驟說明如下:

步驟一: 剖析限制條件。

- 步驟二：形成硬性限制條件的基原。
- 步驟三：以硬性限制條件的基原產生初始的族群。
- 步驟四：計算適應值。
- 步驟五：選擇子代。
- 步驟六：進行「互換突變」的運算程序。
- 步驟七：進行「均勻交配」的運算程序。
- 步驟八：計算適應值。
- 步驟九：進行「排序複製」的運算程序。
- 步驟十：將突變、交配與複製運算程序所產生的新染色體插入母代。
- 步驟十一：判斷是否滿足停止規則（本論文是以滿足某一固定代數，則停止進行基因演算法的運作程序），如果是的話，則規劃結束；反之，則進行步驟五。

其運算邏輯如圖 4.8 所示。

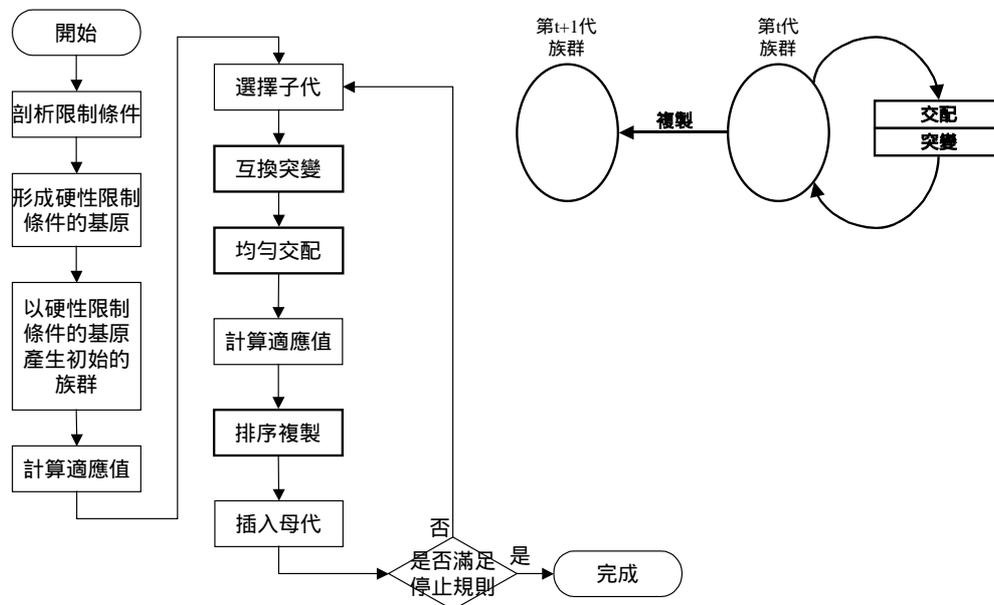


圖 4.8 簡單型基因演算法的運作邏輯

對於演算法而言，能否有能力搜尋到可行解是重要的。若演算法能符合單調性 (monotone) 的條件，那麼他一定會具有收斂的能力。單調性指的是基因演算法所產生每一代的族群 P^0, P^1, \dots, P^t ，能符合以下條件：

$$\forall t \in N : \text{Min.}\{f(a) | a \in P^{t+1}\} \leq \text{Min.}\{f(a) | a \in P^t\} \quad \text{or}$$

$$\forall t \in N : \text{Max.}\{f(a) | a \in P^{t+1}\} \geq \text{Max.}\{f(a) | a \in P^t\}$$

如欲求最大值，族群 P^{t+1} 的最大適應值大於等於族群 P^t 的最大適應值，此稱為單調遞增 (monotonic increasing)。如欲求最小值，族群 P^{t+1} 的最大適應值小於等於族群 P^t 的最大適應值，此稱為單調遞減 (monotonic decreasing)。演算法的單調性，可確保演算法在演化的過程能朝可行解的方向搜尋。

本論文所提出的基因演算法中，複製運算子選擇排序複製運算子的方法，是由母代中各染色體的適應率大小來選取子代。由於隨機產生亂數的方式來決定子代，能確保在複製值的值域 $[0,1)$ 內，每個複製值出現的機率都相同。換句話說，當族群內染色體的個數愈多時，適應率較大的染色體被保留至子代的機會愈大 (也就是個數愈多)。因此，只要在族群數夠大及隨機值夠亂 (值域內各值出現的機率均等) 的二個假設條件下，就可確保子代族群最佳適應值不比母代族群最佳適應值來得差，以達到單調性。

由於本論文所提出的基因演算法是以系統時間為亂數種子產生隨機值 (夠亂)，而族群數的設定數是由實驗中取得能使得基因演算法收斂的參數 (夠大)，所以此基因演算法具有單調性，更能確保在演化的過程能朝可行解的方向搜尋。

第五章 液晶顯示器產業的範例驗證與評估

本章節將以一個簡單的範例來分別驗證本論文所提出的「以產銷平衡為基礎之規劃邏輯」、「訂單分配模式」及「以限制條件為基礎之基因演算法」等之可行性。因此，第 5.1 節將概述範例情節，並說明規劃結果；第 5.2 節分別針對本論文所提出的方法及模式加以分析及探討。

5.1. 範例情節概述

本範例情節假設 A 公司內有六個不同的生產工廠，其生產流程如圖 5.1 所示。六個生產工廠分別為二個薄膜電晶體製造廠、二個單元製造廠及二個模組組裝廠，且生產相同品項的工廠具有相同的製程能力，例如單元製造廠所需要的物料可以分別由二個薄膜電晶體製造廠來提供。整個生產流程是由薄膜電晶體製造廠所製造的薄膜電晶體，運送至單元製造廠進行單元切割及與彩色濾光片組裝的作業，再送至模組組裝廠做最後的組裝作業。由於本範例假設彩色濾光片為原物料，所以不考慮彩色濾光片製造廠的生產狀況。

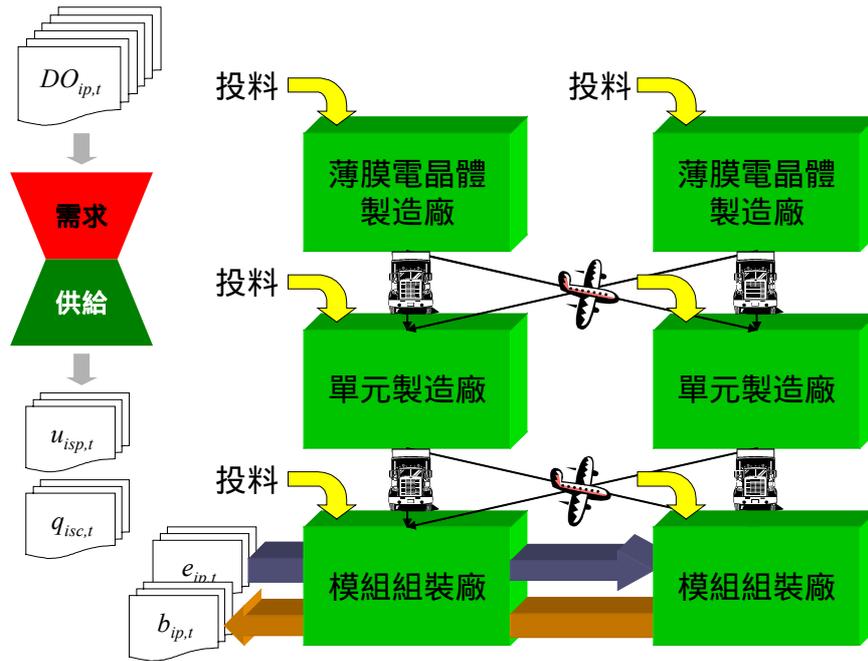


圖 5.1 薄膜電晶體液晶顯示器的多廠區生產系統

根據第 3.3 節的訂單分配模式之數學符號，可以將該多廠區生產系統轉換成圖 5.2 的範例。薄膜電晶體製造廠的平均製造週期為 10 天及輸出品項為薄膜電晶體 (TFT)，單元製造廠的平均製造週期為 3 天及輸出品項為液晶顯示器 (LCD)，模組組裝廠的平均製造週期為 6 天及輸出品項為液晶模組 (LCM)，也就是最終產品—薄膜電晶體液晶顯示器 (TFT-LCD)。

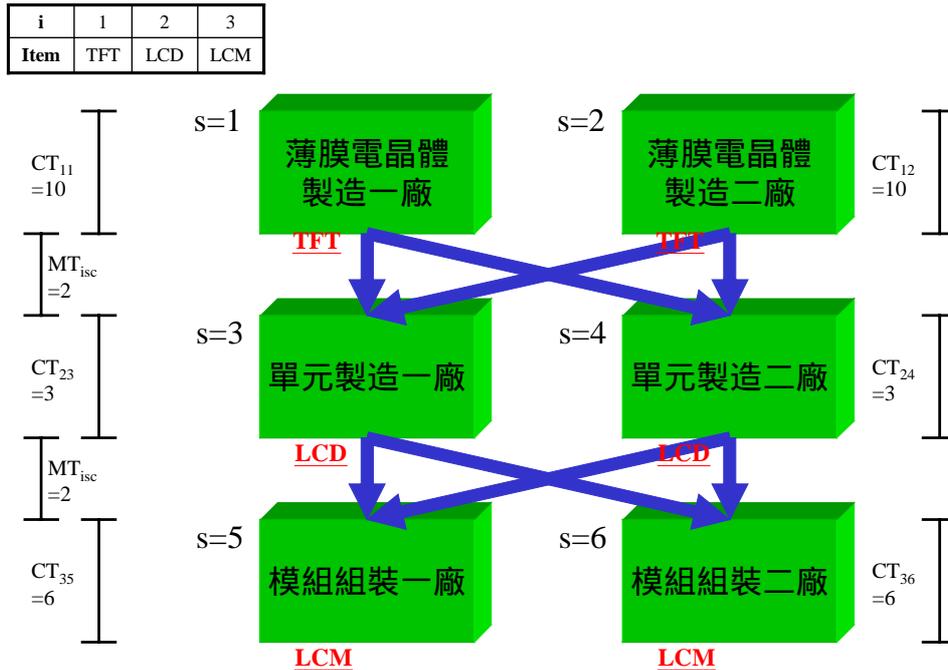


圖 5.2 多廠區生產系統的範例示意圖

假設 A 公司設定規劃週期為 6 個月，各個月的最終產品所彙總之需求量（包含預測及顧客需求），如表 5.1 所示。而本範例的需求主要分為二種尺寸：(1) 尺寸 1，是指玻璃基板為 15 吋的產品群組；(2) 尺寸 2，是指玻璃基板為 17 吋的產品群組。

表 5.1 各月彙總的需求表

月份	產品群組名稱	尺寸	需求量
3 月	LCM	1	440000
3 月	LCM	2	440000
4 月	LCM	1	260000
4 月	LCM	2	260000
5 月	LCM	1	340000
5 月	LCM	2	340000
6 月	LCM	1	240000

6月	LCM	2	240000
7月	LCM	1	160000
7月	LCM	2	160000
8月	LCM	1	280000
8月	LCM	2	280000

由於台灣地小運輸方便，所以本範例皆假設運輸的供給能力之上、下限分別為 270000 個及 150000 個，也就是由來源廠運送物料至消耗廠的上、下限值。而目標函式中的各成本常數如表 5.2 及表 5.3 所示，則其它的基本輸入資料如表附錄 C.1 所示。

表 5.2 各產品群組的製造及儲存成本

產品群組名稱	每單位製造成本	每單位儲存成本
TFT	3000	8
LCD	2500	12
LCM	3500	50

表 5.3 各廠區間的運輸成本

來源廠	消耗廠	每單位運輸成本	來源廠	消耗廠	每單位運輸成本
1	3	100	3	5	100
2	3	200	4	5	200
2	4	100	4	6	100
1	4	200	3	6	200

本論文以 C++ 程式語言撰寫基因演算法及訂單分配模式之程式 (其類別圖請參閱附錄 B)，並於 Microsoft Windows 2000 Professional 之作業系統的電腦 (CPU：P 800；RAM：384MB) 測試及實證該

程式。表 5.4 為基因演算法之參數設定表。而實驗結果之趨勢如圖 5.3 所示。

表 5.4 基因演算法之參數設定表

執行代數	7000
交配率	85%
突變率	5%
每單位懲罰值	1000000

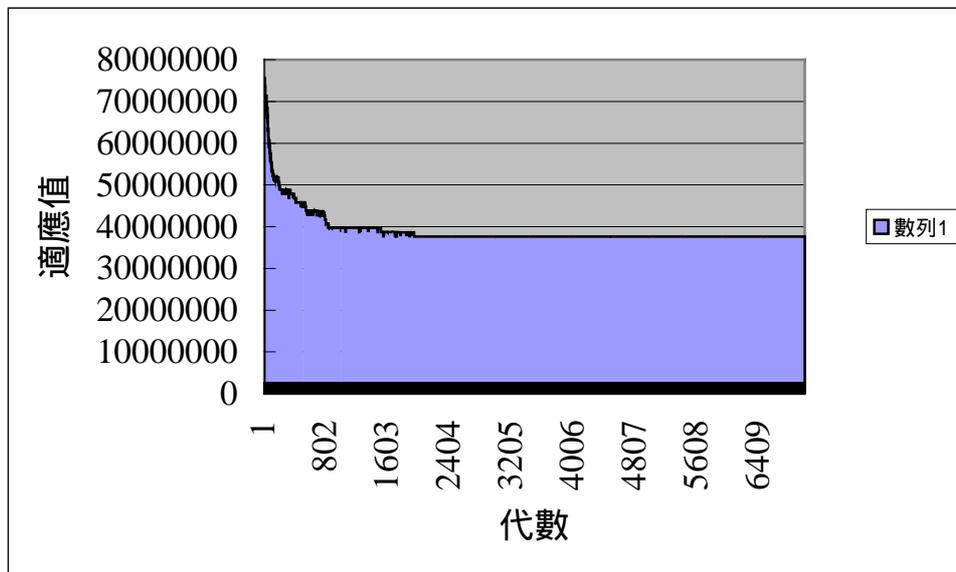


圖 5.3 實驗結果之區域圖

由圖 5.3 可知，其求解過程呈現收斂狀態，驗證本論文所提出以限制條件為基礎之基因演算法可以求解於訂單分配模式，並得到最小成本為 59503886，而 300 個限制條件中共違反 8 個限制條件，則其他輸出資料如表 5.5 至表 5.9 所示。

表 5.5 製造訂單的分配表

月份	品名	廠區	尺寸	投入量	月份	品名	廠區	尺寸	投入量
3月	TFT	1	1	140000	5月	TFT	2	1	136000
3月	TFT	1	2	110000	5月	TFT	2	2	114000
3月	TFT	2	1	138000	5月	LCD	3	1	146000
3月	TFT	2	2	112000	5月	LCD	3	2	114000
3月	LCD	3	1	150000	5月	LCD	4	1	150000
3月	LCD	3	2	110000	5月	LCD	4	2	110000
3月	LCD	4	1	146000	5月	LCM	5	1	158000
3月	LCD	4	2	114000	5月	LCM	5	2	112000
3月	LCM	5	1	160000	5月	LCM	6	1	160000
3月	LCM	5	2	110000	5月	LCM	6	2	110000
3月	LCM	6	1	160000	6月	TFT	1	1	136000
3月	LCM	6	2	110000	6月	TFT	1	2	114000
4月	TFT	1	1	136000	6月	TFT	2	1	138000
4月	TFT	1	2	114000	6月	TFT	2	2	112000
4月	TFT	2	1	136000	6月	LCD	3	1	94000
4月	TFT	2	2	114000	6月	LCD	3	2	166000
4月	LCD	3	1	100000	6月	LCD	4	1	114000
4月	LCD	3	2	160000	6月	LCD	4	2	146000
4月	LCD	4	1	100000	6月	LCM	5	1	134000
4月	LCD	4	2	160000	6月	LCM	5	2	136000
4月	LCM	5	1	146000	6月	LCM	6	1	124000
4月	LCM	5	2	124000	6月	LCM	6	2	146000
4月	LCM	6	1	156000	7月	TFT	1	1	140000
4月	LCM	6	2	114000	7月	TFT	1	2	110000
5月	TFT	1	1	138000	7月	TFT	2	1	140000
5月	TFT	1	2	112000	7月	TFT	2	2	110000

7月	LCD	3	1	146000	8月	TFT	2	1	136000
7月	LCD	3	2	114000	8月	TFT	2	2	114000
7月	LCD	4	1	146000	8月	LCD	3	1	150000
7月	LCD	4	2	114000	8月	LCD	3	2	110000
7月	LCM	5	1	160000	8月	LCD	4	1	146000
7月	LCM	5	2	110000	8月	LCD	4	2	114000
7月	LCM	6	1	160000	8月	LCM	5	1	160000
7月	LCM	6	2	110000	8月	LCM	5	2	110000
8月	TFT	1	1	126000	8月	LCM	6	1	124000
8月	TFT	1	2	124000	8月	LCM	6	2	146000

表 5.6 運輸訂單的分配表

月份	品名	來源	消耗	數量	月份	品名	來源	消耗	數量
3月	TFT	1	3	0	4月	LCD	3	6	150000
3月	TFT	2	3	150000	5月	TFT	1	3	150000
3月	TFT	2	4	0	5月	TFT	2	3	150000
3月	TFT	1	4	158000	5月	TFT	2	4	150000
3月	LCD	3	5	150000	5月	TFT	1	4	0
3月	LCD	4	5	150000	5月	LCD	3	5	150000
3月	LCD	4	6	150000	5月	LCD	4	5	150000
3月	LCD	3	6	2000	5月	LCD	4	6	150000
4月	TFT	1	3	150000	5月	LCD	3	6	150000
4月	TFT	2	3	166000	6月	TFT	1	3	150000
4月	TFT	2	4	150000	6月	TFT	2	3	0
4月	TFT	1	4	150000	6月	TFT	2	4	150000
4月	LCD	3	5	150000	6月	TFT	1	4	150000
4月	LCD	4	5	0	6月	LCD	3	5	150000
4月	LCD	4	6	150000	6月	LCD	4	5	150000

6月	LCD	4	6	150000	7月	LCD	3	6	150000
6月	LCD	3	6	0	8月	TFT	1	3	150000
7月	TFT	1	3	150000	8月	TFT	2	3	150000
7月	TFT	2	3	150000	8月	TFT	2	4	150000
7月	TFT	2	4	150000	8月	TFT	1	4	150000
7月	TFT	1	4	150000	8月	LCD	3	5	150000
7月	LCD	3	5	150000	8月	LCD	4	5	150000
7月	LCD	4	5	150000	8月	LCD	4	6	150000
7月	LCD	4	6	150000	8月	LCD	3	6	150000

表 5.7 存貨單的分配表

月份	品名	廠區	存貨量	月份	品名	廠區	存貨量
3月	TFT	1	3000	5月	LCM	5	957000
3月	TFT	2	3000	5月	LCM	6	807000
3月	LCD	3	121000	6月	TFT	1	74000
3月	LCD	4	1000	6月	TFT	2	74000
3月	LCM	5	607000	6月	LCD	3	301000
3月	LCM	6	342000	6月	LCD	4	446000
4月	TFT	1	72000	6月	LCM	5	839000
4月	TFT	2	72000	6月	LCM	6	994000
4月	LCD	3	428000	7月	TFT	1	78000
4月	LCD	4	278000	7月	TFT	2	78000
4月	LCM	5	976000	7月	LCD	3	436000
4月	LCM	6	966000	7月	LCD	4	436000
5月	TFT	1	78000	7月	LCM	5	1107000
5月	TFT	2	78000	7月	LCM	6	1107000
5月	LCD	3	286000	8月	TFT	1	76000
5月	LCD	4	286000	8月	TFT	2	76000

8月	LCD	3	434000	8月	LCM	5	1099000
8月	LCD	4	434000	8月	LCM	6	0

表 5.8 訂單開立表

月份	品名	尺寸	數量	類型	月份	品名	尺寸	數量	類型
3月	LCM	1	120000	積壓	6月	LCM	1	18000	填補
3月	LCM	2	220000	積壓	6月	LCM	2	42000	填補
4月	LCM	1	42000	填補	7月	LCM	1	160000	填補
4月	LCM	2	22000	積壓	7月	LCM	2	60000	填補
5月	LCM	1	22000	積壓	8月	LCM	1	4000	填補
5月	LCM	2	118000	積壓	8月	LCM	2	24000	積壓

表 5.9 供需平衡的挪移表

積壓訂單				填補產能訂單			
月份	品名	尺寸	數量	月份	品名	尺寸	數量
3月	LCM	1	120000	7月	LCM	1	160000
5月	LCM	1	22000				
5月	LCM	2	118000	6月	LCM	2	42000
				7月	LCM	2	60000

表 5.5 中以 3 月份的薄膜電晶體製造一廠之分配結果為例，為了避免重複製造某特定尺寸的品項，其投入量不能大於製造上限，所以尺寸 1 及尺寸 2 的製造上限量分別為 140000 及 240000 個(公式 3.4)。由於各廠的物料投入量同時必須等於該廠的投入量上限，而薄膜電晶體製造一廠的投入量上限為 250000 個(如表附錄 C.1 所示)，也就是尺寸 1 及尺寸 2 的物料投入量總和必須等於 250000 個(公式 3.3)。因此，尺寸 1 及尺寸 2 的物料投入量分別求得 140000 及 110000 個。

同理，運輸訂單中薄膜電晶體製造一廠的運輸量，受到運輸供給能力的上、下限所限制（公式 3.1 及 3.2），表 5.6 中薄膜電晶廠製造一廠運送 158000 個薄膜電晶體至單元製造二廠。其結果也顯示該廠並沒有運送薄膜電晶體至單元製造一廠，也就是違反運輸供給能力的下限限制（公式 3.2），而此限制式於本範例中視為軟性限制條件，因此，其結果是被容許的，但是必須付出違反限制條件的懲罰成本。

表 5.9 是根據表 5.8 所開立的積壓訂單及填補產能訂單來做供需平衡表。以「3 月尺寸 1 的 LCM」之積壓訂單及「5 月尺寸 1 的 LCM」之積壓訂單為例，二者的數量總合剛好可以以「7 月尺寸 1 的 LCM」之填補產能訂單來平衡，也就是將原來無法生產的積壓訂單挪移至有多餘產能的月份（即有開立填補產能訂單的月份）生產。換句話說，因為提高資源使用率所開立的填補產能訂單，可以提前或延遲消耗其他月份無法生產的積壓訂單，如圖 3.12 所示。

5.2. 實驗評估與分析

本論文將分別對「訂單分配模式之可行性」及「基因演算法之求解能力」等二方面進行評估與分析，進而驗證模式與方法的可行性及正確性。並且將實驗數據帶入統計軟體 STATISTICA，進而取得各類統計圖表。

5.2.1. 訂單分配模式之可行性

為了驗證本論文所提出的訂單分配模式之可行性，本實驗將以 Microsoft Excel 2000 中「規劃求解」模組內不同的線性規劃解法，來求解於訂單分配模式。

Microsoft Excel 2000 中「規劃求解」模組內非線性模式的解法，是使用 Leon Lasdon（University of Texas at Austin）和 Allan Waren（Cleveland State University）所發展的 Generalized Reduced Gradient（GRG2）非線性最佳化程式碼。而線性模式的解法，是使用單純

(simplex) 方法，並對變數加入限制，以及分支界限 (branch-and-bound) 方法，這些方法是由 Frontline Systems 公司的 John Watson 和 Dan Fylstra 設計的。

本實驗主要分為線性及非線性模式的解法，及以不同的誤差容忍度、估計式、導函數及搜尋法的組合來分別求出最小成本，如表 5.11 及表 5.12 所示，並且各做二次重複實驗，則求解結果如表附錄 C.2 至表附錄 C.6 所示。其參數設定表如表 5.10 所示。

由 64 組實驗數據得知，無論那一種解法得到相同的最小成本，因此，可以驗證本論文所提出的訂單分配模式至少存在一個可行解。而不同的線性規劃解法得相同的最小成本，此一致性的結果反應出數學模式具有一定的可行性。

表 5.10 線性規劃之參數設定表

最長運算時間	1000 秒
反覆運算次數	10000 次
精確度	0.9
收斂度	0.1

表 5.11 線性模式的實驗數據

		誤差容忍度 5%		誤差容忍度 10%	
		正切函數 估計式	二次方程 估計式	正切函數 估計式	二次方程 估計式
前推離 導函數	牛頓 搜尋法	62050590	62050590	62050590	62050590
	共軛 搜尋法	62050590	62050590	62050590	62050590
	牛頓 搜尋法	62050590	62050590	62050590	62050590
	共軛 搜尋法	62050590	62050590	62050590	62050590
中央離	牛頓 搜尋法	62050590	62050590	62050590	62050590
	共軛 搜尋法	62050590	62050590	62050590	62050590

導函數	共軛 搜尋法	62050590	62050590	62050590	62050590
		62050590	62050590	62050590	62050590

表 5.12 非線性模式的實驗數據

		誤差容忍度 5%		誤差容忍度 10%	
		正切函數 估計式	二次方程 估計式	正切函數 估計式	二次方程 估計式
前推離 導函數	牛頓 搜尋法	62050590	62050590	62050590	62050590
		62050590	62050590	62050590	62050590
	共軛 搜尋法	62050590	62050590	62050590	62050590
		62050590	62050590	62050590	62050590
中央離 導函數	牛頓 搜尋法	62050590	62050590	62050590	62050590
		62050590	62050590	62050590	62050590
	共軛 搜尋法	62050590	62050590	62050590	62050590
		62050590	62050590	62050590	62050590

5.2.2. 基因演算法之求解能力

為了驗證以限制條件為基礎之基因演算法應用於訂單分配模式具有求解能力，本實驗依表 5.4 的參數設定值，做 50 次實驗，獲取基因演算法進入收斂狀態的初始代數之實驗數據，如表 5.13 所示。由 50 個實驗數據可知，基因演算法均能於 7000 代內收斂至穩定狀態，乃說明基因演算法具有收斂的特性。

表 5.13 50 次實驗進入收斂狀態的初始代數

4162	6468	6303	6738	6554	6721	6825	6543	6679	6421
6891	6158	6777	6911	5008	6835	5133	6634	5864	6734
6145	5756	6569	5686	6672	5399	6502	6931	5949	6883

5715	6067	5639	6834	5967	6311	6210	3849	6767	6884
6449	6910	6602	4652	6313	5821	6986	5359	6441	6303

此外，本實驗更進一步地收集 24 次實驗的最小成本，如表 5.14 所示。而圖 5.4 是表 5.14 所繪製而成的直方圖，由此可以看出 24 組實驗數據所呈現的直方圖類似常態分配。依中央極限定理，若實驗次數愈多時，其直方圖愈近似常態分配。換句話說，此基因演算法收斂特性趨近於某個成本值，也就是顯示其具有一定程度的穩定性。

表 5.14 24 次實驗的最小成本（軟、硬性限制條件）

59503886	59120248	58891598	58159702	58077600	59770406
59467266	58070376	59885362	58782774	58925100	58302222
58988300	59387370	59237090	58413630	59027820	57931666
59228314	59652422	59792044	58904414	58888762	57521130

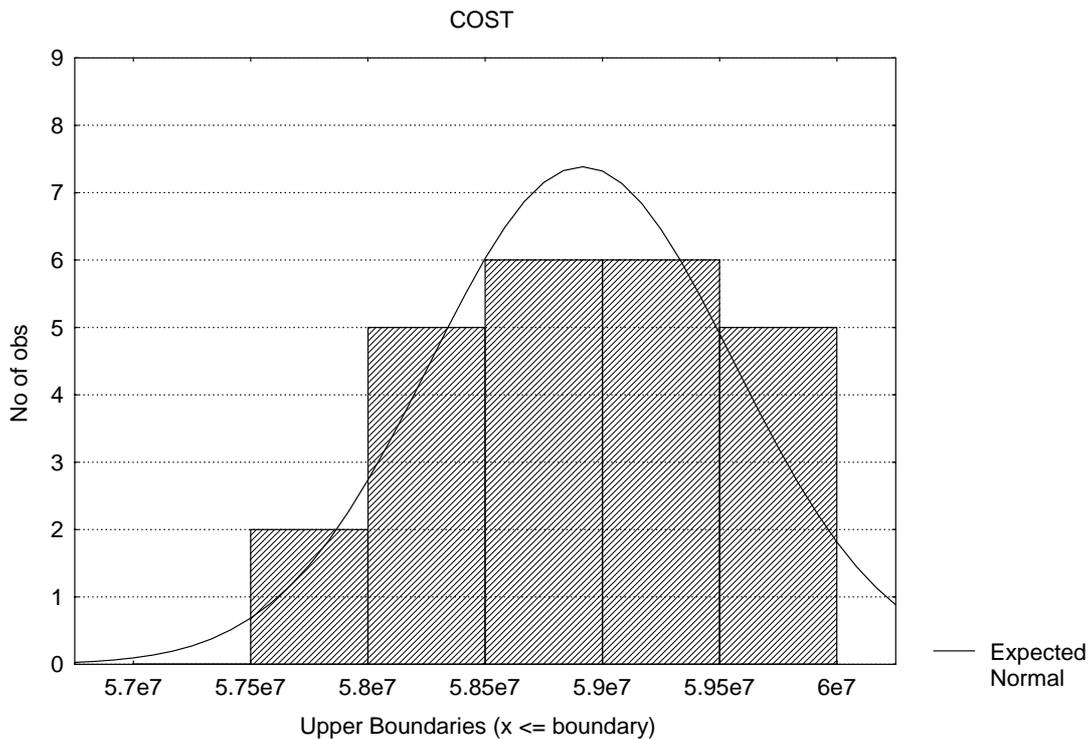


圖 5.4 最小成本的直方圖

為了比較基因演算法的求解品質，分別以第 3.2 節所提出的規劃邏輯(啟發式法則)，及基因演算法與線性規劃等求解方法做一比較。而啟發式法則的輸出結果，如表附錄 C.7 至表附錄 C.10 所示。將各種方法所得到的各期存貨量繪製成折線圖，如圖 5.5 所示。由圖 5.5 可以發現存貨量相當高，這是因為重視資源使用率的液晶顯示器產業為了提高設備使用率，將各廠區的產能滿載(如公式 3.3)而造成存貨上升，特別是本論文所提出的訂單分配模式並非以存貨水準最小化為主。但對於機器設備的購買成本遠大於物料存貨成本的情況下，犧牲掉物料存貨成本來提高設備使用率所造成的成本效益是對公司有利的。

圖 5.5 中三種規劃方法所求解出來的各期存貨量，能夠看出啟發式法對於控制存貨的能力方面表現較差，並且具有不斷遞增的特性。相較與啟發式法則，線性規劃與基因演算法對於控制存貨的能力較好，但是線性規劃所求得的存貨量仍具有順時間遞增的趨勢，而基因演算法則較有平準化存貨的特性。針對三種不同規劃方法之特性彙整成比較表如表 5.15 所示。

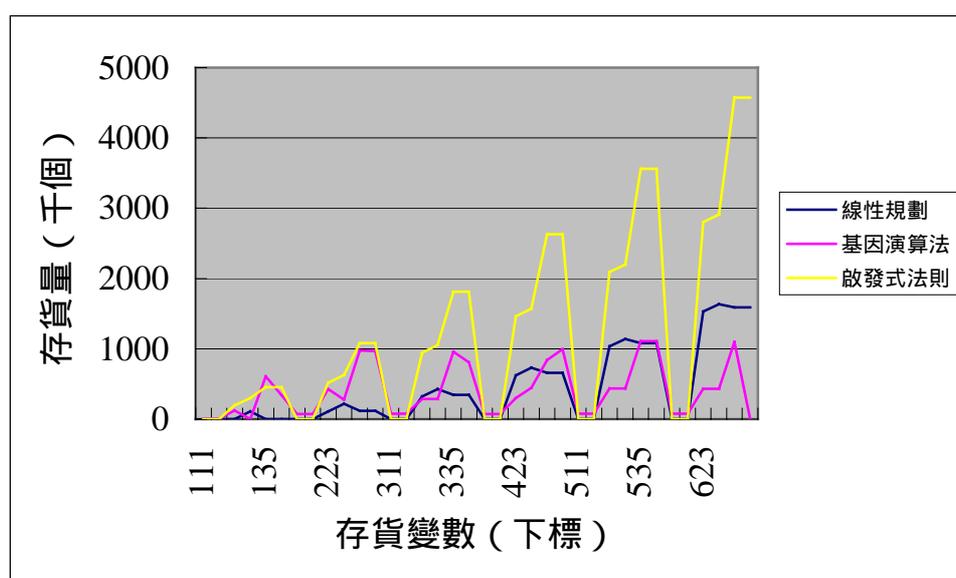


圖 5.5 各期存貨量之折線圖

表 5.15 求解方法的比較表

	啟發式法則	以限制條件為基礎之基因演算法	線性規劃
最小成本	62183801	62166800	62050590
求解速度	快 (1 秒鐘)	慢 (平均 4 分鐘)	次之 (3 秒鐘)
處理限制 條件能力	具備 (經驗法則)	同時具備處理硬性及軟性限制 條件的能力	僅硬性限 制條件
違反限制 條件個數	10 個	平均 10 個	無
控制存貨 能力	最差， 陡峭遞增	較平準化	次好，但有 遞增趨勢

表 5.16 24 次實驗的最小成本 (硬性限制條件)

62169846	62166624	62169785	62168108	62162840	62163345
62163909	62170195	62170737	62165358	62162944	62169905
62162556	62169940	62170129	62161827	62168362	62163159
62167317	62165497	62170581	62164673	62171037	62164538

為了有效地比較「啟發式法則」、「基因演算法」及「線性規劃」之間的差異性，表 5.15 中「最小成本」及「求解速度」的數據，是將基因演算法處理限制條件的方法全部以硬性限制條件來處理，取得表 5.16 的數據，則其他方面以表 5.14 的實驗數據做比較。由表 5.15 的比較可知，基因演算法所求算的成本介於其他二者之間，但是求解速度是三者中最慢的，這也顯示基因演算法為了求取較高求解品質卻犧牲掉求解速度之特點，而其所求算的最小成本與線性規劃所求算的最小成本相差不大，也就是說，其求解能力與線性規劃的求解能力相

差不大（意謂著基因演算法具有相當程度的求解品質）。但線性規劃僅能處理硬性限制條件，而基因演算法具有處理硬性及軟性限制條件的能力，因此，基因演算法能夠滿足硬性限制條件的情況下，以違反較少個數的軟性限制條件來換取比線性規劃更少的成本。為了驗證基因演算法確實比線性規劃所求解出來的成本還要好，利用統計檢定的方式來比較兩種求解方法的成本效益是否有顯著的差異。假設

H_0 ：兩者求解方法沒有顯著的差異。

H_1 ：兩者求解方法有顯著的差異。

以兩組獨立樣本 t 檢定求得報表如表 5.17 所示。

表 5.17 t 檢定的報表

基因演算法的平均值	線性規劃的平均值	t 值	自由度	P 值
58913729.25	62050590	-23.7075	46	0

顯著水準 =0.01

由表 5.17 可知，P 值遠小於顯著水準 =0.01，故拒絕 H_0 的假設。因此，本實驗有 99% 的信心顯示基因演算法與線性規劃有顯著的差異，而基因演算法的平均值大於線性規劃的平均值，由此可知，基因演算法的求解品質比線性規劃來得好。

整體來說，本小節所做的實驗驗證了本論文所提出的基因演算法能求解於訂單分配模式，而其求解品質也相較與啟發式法則及線性規劃來得好。換句話說，本論文所提出的基因演算法可以做為訂單分配模式之求解方法。

第六章 結論與建議

6.1. 結論

本論文主要提出以產銷平衡為考量基礎之訂單分配的決策模式，供台灣液晶顯示器產業做為參考，進而減少銷售與生產之間的失調問題。總結，研究結果可歸納如下：

- (1) 本論文所提出應用於台灣產業之多廠區生產系統下產銷平衡的規劃邏輯，是以訂單為主的分配方式，主要提高資源使用率為決策依據，並且考量物料及產能的需求。因此，符合液晶顯示器產業以「提高資源使用率」為生產目標之特性，且由範例驗證的結果可知，該規劃邏輯具有不錯的求解品質及速度。
- (2) 針對多廠區生產環境的特性所發展產銷平衡的訂單分配模式，不僅考量運輸、資源、數量、產銷平衡及物料需求的限制條件，而且在三種不同規劃方法所求算的規劃結果中，皆能求解出不錯的可行解，更進一步地規劃出積壓訂單及填補產能訂單的開立建議表，提供給生產人員做為參考之用。
- (3) 實作以限制條件為基礎的基因演算法來求解於訂單分配模式，經由範例驗證的結果證實其方法具有一定的可行性，特別是在控制存貨能力方面比線性規劃來得好，更加符合本論文應用於計畫式生產的基本假設。

6.2. 未來發展方向

除了改善本論文之缺點及限制，可以做為未來研究方向外，針對本論文所提出以產銷平衡（supply-demand balancing）為基礎之訂單分配的決策模式採用中央式（centralized）規劃方式，僅應用於同一企業下多廠區生產系統，且生產資訊完全透明化的生產環境。對於台灣本土內的多廠區生產系統而言，由於運輸便利使得各廠區之間的支援能力相當充足，且生產資訊彙整容易，適合於採用中央式規劃方

式。面臨全球運籌化與液晶顯示器的供應鏈體系日趨完整的趨勢下，處於生產資訊分散於全球各地且資料量龐大而彙整不易的情形，中央式規劃方式未來可能不適用於此生產環境，而分散式（distributed）規劃方式具有「在不同企業間生產資訊不完全透明化的條件下，各廠區能各自達成生產目標，且藉由彼此的合作關係完成共同目標」的特性。對不同企業下生產資訊取得不易的多廠區生產系統，可以發展出同步考量各廠區不同生產目標的多廠區生產排程系統。

因此，未來的研究方向可歸納如下：

- (1) 針對薄膜電晶體液晶顯示器產業之多廠區生產環境的特性，發展產銷平衡的訂單分配模式。此訂單分配模式主要考量一般性液晶顯示器產業的限制條件，並未能真正考量實際業界真實的限制條件，僅能部份解決實際遭遇的問題。特別是當多廠區生產系統之生產流程相當複雜時，是否能藉由此訂單分配模式延伸出更複雜的問題，進而求解該問題，也將是未來研究方向的一大挑戰。
- (2) 本論文並不著重於演算法之研究，一旦資料量龐大且限制條件更加複雜時，很難確保該方法能保持一定的求解速度及品質，因此，未來可針對此一方向加以延伸與探討。
- (3) 提出多廠區分散式排程（distributed multi-site scheduling）系統之系統架構，且實作出以代理人為基礎（agent-based）的多廠區分散式排程系統。
- (4) 為了有效求解於具有多目標決策之多廠區生產系統的排程問題，可運用多重代理人系統（Multi-Agent System；MAS）具有分散式自主決策的能力，同時考量全方位製造系統（Holonc Manufacturing System；HMS）的控制邏輯能夠以階層式的分散控制及執行同時與非同步的排程等優點，輔助代理人較不具有合作的特性，進而發展一滿足整體目標且兼顧各自生產目標的運作機制。發展多目標決策的溝通（communication）與協調（negotiation）機制。

參考文獻

- [1] Abad, P. I., “Supplier pricing when the buyer’s annual requirements are fixed,” *Computers Operations Research*, 21, 155-167, 1994.
- [2] Adexa 軟體的「產品使用說明書」。
- [3] Banerjee, A., “A joint economic-lot-size model for purchaser and vendor,” *Decision Science*, 17, 292-311, 1986.
- [4] Bullinger, H. J., K. P. Faehrich and H. P. Laubscher, “Planning of multi-site production — an objected-oriented model,” *International Journal of Production Economics*, 51, 1-2, 19-35, August 15, 1997.
- [5] Cheng, R., M. Gen and Y. Tsujimura, “A tutorial survey of job-shop scheduling problems using genetic algorithms –i. representation,” *Computers ind. Engng*, 30, 4, 983-997, 1996.
- [6] Cheng, R., M. Gen and Y. Tsujimura, “A tutorial survey of job-shop scheduling problems using genetic algorithms, part ii: hybrid genetic search strategies,” *Computers ind. Engng*, 36, 343-364, 1999.
- [7] Chiang, W. C., J. Fitzsimons, Z. M. Huang and S. X. Li, “A game-theoretic approach to quantity discount problems,” *Decision Science*, 25, 153-168, 1993.
- [8] Coit, D. W., “Penalty guided genetic search for reliability design optimization,” *Computers Industrial Engineering*, 30, 4, 895-904, 1996.
- [9] Dasci, A. and V. Verter, “A continuous model for production-distribution system design,” *European Journal of Operational Research*, 129, 287-298, 2001.
- [10] Erengus, S. S., N. C. Simpson and A. J. Vakharia, “Integrated production/distribution planning in supply chains: an invited review,” *European Journal of Operational Research*, 115, 219-236, 1999.

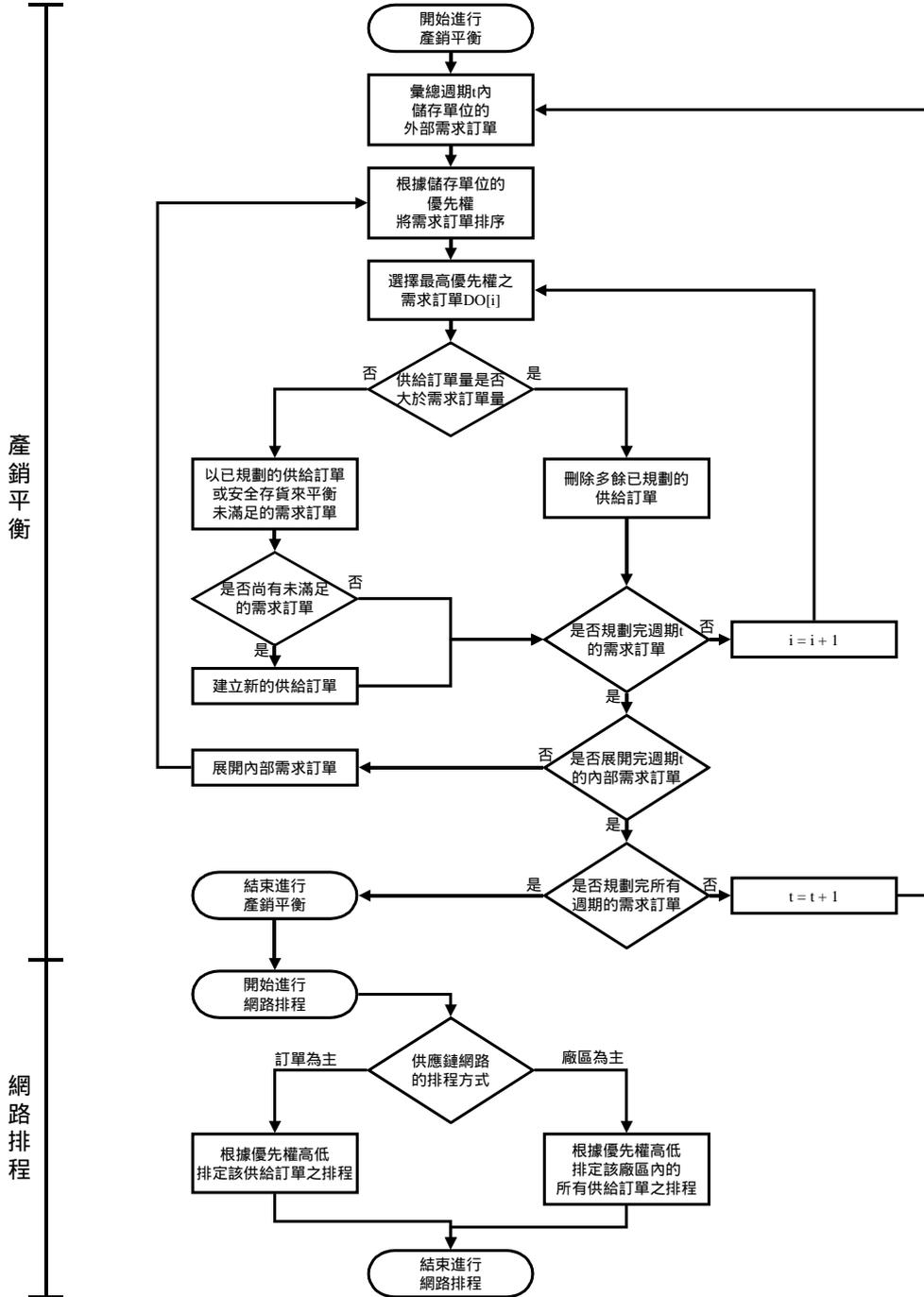
- [11] Falkenauer, E. and S. Bouffoix, "A genetic algorithm for job shop," *in Proceedings of the IEEE International Conference on Robotics and Automation*, 824-829, 1991.
- [12] Frederix, F., "Planning and scheduling multi-site semiconductor production chains: a survey of needs, current practices and integration issues," <http://www.nimblesite.com/xcittic/articles.htm>.
- [13] Gen, M. and R. Cheng, *Genetic algorithms and engineering optimization*, Addison-Wiley, New York, 2000.
- [14] Goldberg, D. E., *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Mass, 1989.
- [15] Guinet, A., "Multi-site planning: a transshipment problem," *Int. J. Production Economics*, 74, 21-32, 2001.
- [16] Gurnani, H., "Optimal ordering policies in inventory systems with random demand and random deal offerings," *European Journal of Operational Research*, 1-15, 94, 1996.
- [17] Handa, H., O. Katai, N. Baba and T. Saeaeagi, "Solving constraint satisfaction problems by using coevolutionary genetic algorithm," *IEEE*, 1998. Handa, H., K. Watanabe, O. Katai, T. Konishi and M. Baba, "Coevolutionary genetic algorithm for constraint satisfaction with a genetic repair operator for effective schemata formation," *IEEE*, 1999.
- [19] Hung, Y.-F. and Leachman, R. C., "A Production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 9, No. 2, 257-269, 1996.
- [20] i2 Technology, <http://www.i2.com>, August 1, 2001.
- [21] Jeong, B., S.-W. K. and Y. Lee, "An assembly scheduler for TFT LCD manufacturing," *Computers & Industrial Engineering*, 41, 37-58, 2001.

- [22] Lambert, D. M., J. R. Stock and L. M. Ellram, *Fundamentals of logistics management*, Irwin/McGraw-Hill, Boston, MA, 1998.
- [23] Leachman, R. C., R. F. Benson, C. Liu and D. J. Raar, “IMPreSS: an automated production-planning and delivery-quotation system at harris corporation—semiconductor sector,” *INTERFACES*, 26, 6-37, January-February, 1996.
- [24] Liepins, G., M. Hilliard, J. Richardson, and M. Pallmer, “Genetic algorithm application to set covering and traveling salesman problem”, in Brown, editor, *OR/AI: The Integration of Problem Solving Strategies*, 1990.
- [25] Manugistics 軟體的「產品使用說明書」。
- [26] Metters, R., “Quantifying the bullwhip effect in supply chains,” *Journal of Operations Management*, 15, 89-100, 1997.
- [27] Michalewicz, Z., *Genetic algorithms + data structures = evolution programs*, 2nd, extended ed., Springer-Verlag, New York, 1994.
- [28] Michalewicz, Z. and G. Nazhiyath, “A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints,” *Proceeding of the 2nd IEEE ICEC*, D. Fogel(ed.), 1995.
- [29] Paredis, J., *Exploiting constraints as background knowledge for evolutionary algorithm*, The Handbook of Evolutionary Computation, Back, T., Fogel, D, 1997.
- [30] Paredis, J., “Exploiting constraints as background knowledge for genetic algorithm: a case-study for scheduling,” *Proceedings of the Second Conference on Parallel Problem Solving from Nature*, 1992.
- [31] Paredis, J., “Genetic state-space search for constrained optimization problems,” *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1993.
- [32] Pirkul, H. and V. Jayaraman, “A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient

- heuristic solution,” *Computers Ops. Res.*, 25, 10, 869-878, 1998.
- [33] Ross, A. D., “Performance-based strategic resource allocation in supply networks,” *Int. J. Production Economics*, 63, 255-266, 2000.
- [34] Sabri, E. H. and B. M. Beamon, “A multi-objective approach to simultaneous strategic and operational planning in supply chain design,” *Omega*, 28, 581-598, 2000.
- [35] Sauer, J. and H.-J. Appelrath, “Integrating transportation in a multi-site scheduling environment,” *Proceeding of the 33rd Hawaii International Conference on System Sciences*, 2000.
- [36] Sauer, J., G. Suelmann and H.-J. Appelrath, “Multi-site scheduling with fuzzy concepts,” *International Journal of Approximate Reasoning*, 19, 145-160, 1998.
- [37] Sue, A.-H., “Network design in supply chain management,” *International Journal of Agile Management Systems*, 1/2, 99-106, 1999.
- [38] Thierry, C., P. Besnard, D. Ghattas and G. Bel, “Multi-site planning: non flexible production units and set-up time treatment,” *IEEE*, 1995.
- [39] Timpe, C. H. and J. Kallrath, “Optimal planning in large multi-site production networks,” *European Journal of Operational Research*, 126, 422-435, 2000.
- [40] Tsang, E., *Foundations of constraint satisfaction*, Academic Press, 1993.
- [41] Victoria, L. C., “Close the marketing/manufacturing gap,” *Sloan Management review*, 1992.
- [42] Zhou, Q. and C. B. Besant, “An information management architecture for multi-site production planning and control,” <http://www.nimblesite.com/xcittic/articles.htm>.
- [43] 蔡與哲, 「運用先進規劃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。

- [44]郭乃綦，「晶圓製造廠多廠間訂單抵換機制之構建」，交通大學工業工程與管理學系，碩士論文，民國 90 年。
- [45]陳碧暉，「半導體多廠區生產計劃」，清華大學工業工程與工程管理學系，碩士論文，民國 90 年。
- [46]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。
- [47]陳智勇，「供應商於供給不足下分配方法之研究」，台灣大學商學研究所，碩士論文，民國 89 年。
- [48]葉建緯，“景氣變化對供應鏈管理的衝擊與回應（上）”，
http://www.advtek.com.tw/Index/index_flash.asp, 2001.
- [49]葉建緯，“景氣變化對供應鏈管理的衝擊與回應（下）”，
http://www.advtek.com.tw/Index/index_flash.asp, 2001.
- [50]黃敏樟，「電子業供應鏈上下游同步規劃模式之探討」，東海大學工業工程研究所，碩士論文，民國 90 年。
- [51]黃黎毅，「製造業訂單管理系統之資訊流程與決策模式的建立」，東海大學工業工程研究所，碩士論文，民國 86 年。
- [52]廖祐笙，「整合基因演算法與限制條件滿足之研究」，東海大學工業工程研究所，碩士論文，民國 88 年。
- [53]陳文瑞，「嵌入限制條件於基因演算法之研究—以自然產及剖腹產資源分配為例」，東海大學工業工程研究所，碩士論文，民國 90 年。
- [54]楊依倫，「以基因演算法為基礎之供應鏈物料配置決策模式」，東海大學工業工程研究所，碩士論文，民國 90 年。
- [55]曾煥雯，「跨廠訂單分配模式之構建—應用模擬退火演算法」，元智大學工業工程研究所，碩士論文，民國 89 年。

附錄 A



圖附錄 A.1 一般性多廠區生產規劃流程圖
—啟發式 (heuristics) 流程

產銷平衡規劃流程主要是決定供給訂單 (Supply Order ; SO) 的數量。網路排程規劃流程則是決定供給訂單的交期。所謂供給訂單指的是用來滿足需求訂單 (Demand Order ; DO) 的訂單。需求訂單的來源有二：一是顧客需求，即顧客訂單；一是預測需求，即預測訂單。因為此需求皆來自於外部，故又稱外部訂單 (external order)，而內部訂單 (internal order) 指的是來自相依性需求的需求訂單。

產銷平衡規劃流程的運作步驟如下：

- 步驟一、彙總週期 t 內儲存單位 (Stock Keeping Unit ; SKU) 的外部需求訂單。儲存單位是某特定存貨區/廠區 (location/site) 的品項 (item)，其功用在於控制存貨，例如，在同一存貨中，兩件不同顏色的吊扇可能在不同的儲存單位。
- 步驟二、根據儲存單位的優先權 (priority)，將需求訂單排序。
- 步驟三、選擇最高優先權之需求訂單 $DO[i]$ 。
- 步驟四、判斷供給訂單量是否大於需求訂單，如果是的話，則進行步驟六；反之，則進行步驟五。
- 步驟五、刪除多餘已規劃的供給訂單，接著進行步驟九。
- 步驟六、以已規劃的供給訂單或安全存貨來平衡未滿足的需求訂單。
- 步驟七、判斷是否尚有未滿足的需求訂單，如果是的話，則進行步驟八；反之，則進行步驟九。
- 步驟八、建立新的供給訂單。
- 步驟九、判斷是否規劃完週期 t 的需求訂單，如果是的話，則進行步驟十；反之，則計算 $i = i + 1$ ，接著進行步驟三。
- 步驟十、判斷是否展開完週期 t 的內部需求訂單，如果是的話，則進行步驟十二；反之，則進行步驟十一。展開內部需求訂單是根據該儲存單位的生產或運輸流程，以後推 (backward) 方式追溯上游廠區之內部需求訂單。

步驟十一、依顧客指定生產或運輸流程來展開內部需求訂單，接著進行步驟二。

步驟十二、判斷是否規劃完所有週期的需求訂單，如果是的話，則進行步驟十三；反之，則計算 $t = t + 1$ ，接著進行步驟一。

步驟十三、結束進行產銷平衡規劃流程，開始進行網路排程規劃流程。
網路排程規劃流程的運作步驟如下：

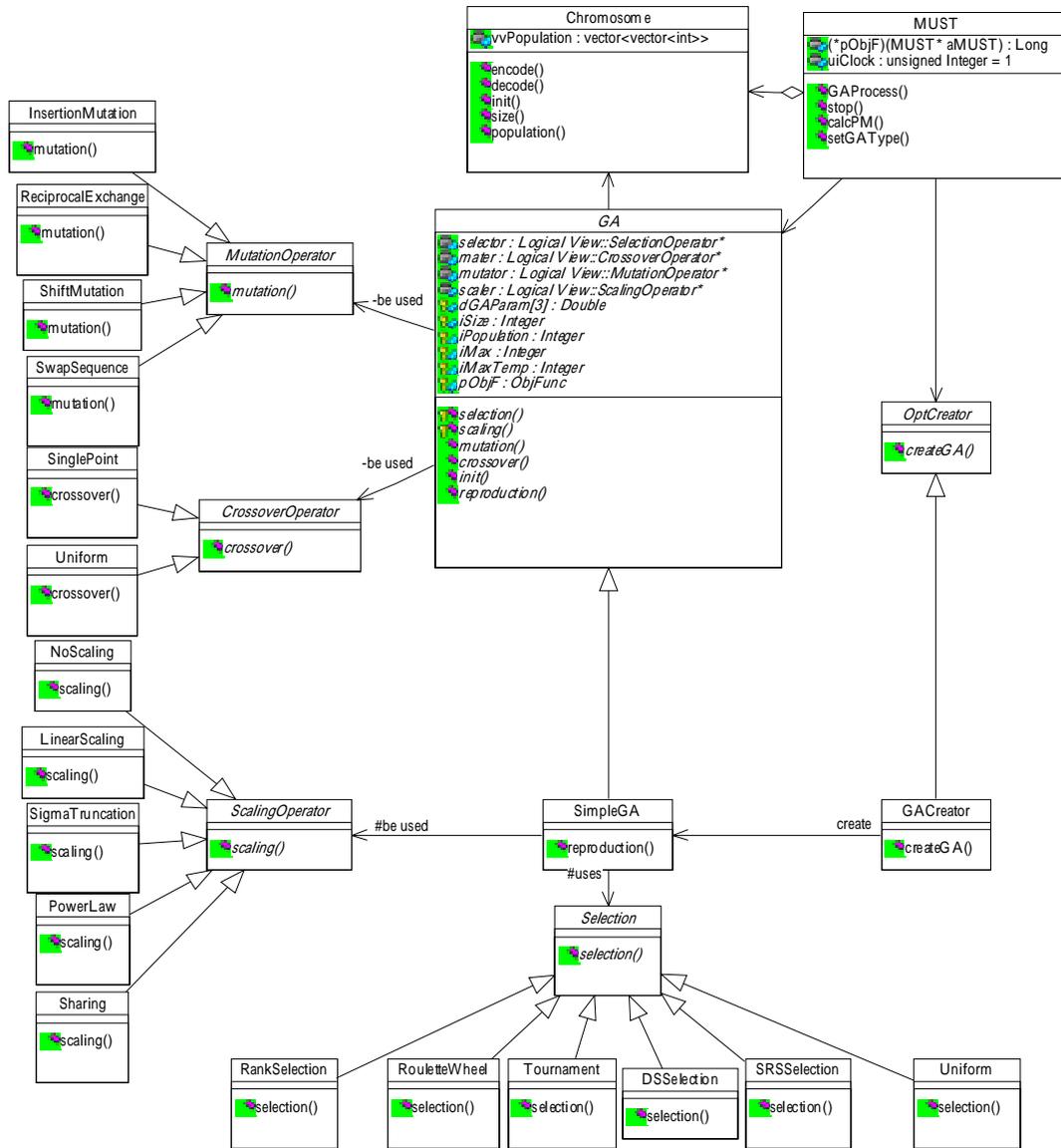
步驟一、決定供應鏈網路的排程方式，如果以訂單為主的話，則進入步驟二；如果以廠區為主的話，則進入步驟三。

步驟二、根據訂單優先權高低，排定該供給訂單的排程，直到排完所有的供給訂單，至步驟四。排程方式是視該訂單所在存貨位置/廠區（location/site）而決定。

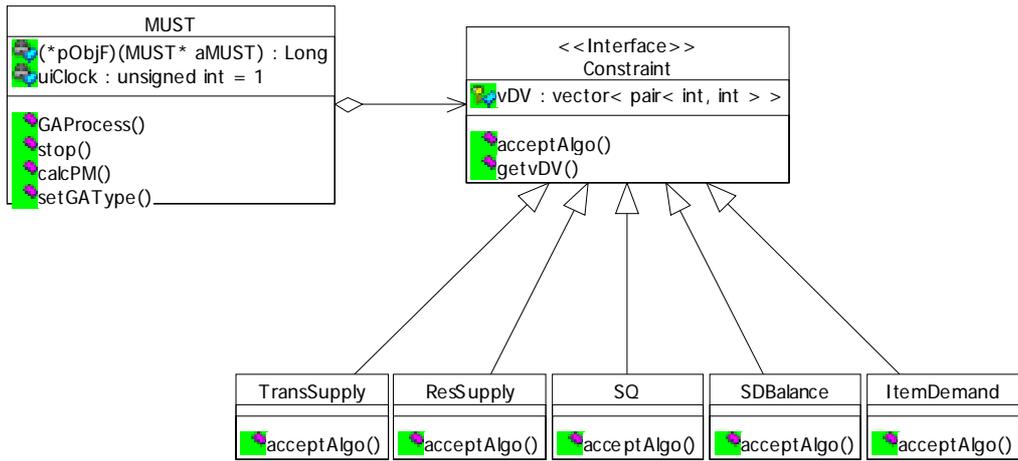
步驟三、根據廠區優先權高低，排定該廠區內的所有供給訂單之排程，直到排完所有廠區的供給訂單。

步驟四、完成多廠區生產規劃流程。

附錄 B



圖附錄 B.1 基因演算法之類別圖



圖附錄 B.2 訂單分配模式的類別圖

附錄 C

表附錄 C.1 基本輸入資料

月份	品名	廠區	尺寸	分級率	良率	製造上限	投入量上限
3月	TFT	1	1	0.95	0.95	140000	250000
3月	TFT	1	2	0.95	0.97	240000	250000
3月	TFT	2	1	0.95	0.95	140000	250000
3月	TFT	2	2	0.95	0.97	240000	250000
3月	LCD	3	1	0.93	0.95	150000	270000
3月	LCD	3	2	0.93	0.97	250000	270000
3月	LCD	4	1	0.93	0.95	150000	270000
3月	LCD	4	2	0.93	0.97	250000	270000
3月	LCM	5	1	0.97	0.95	170000	270000
3月	LCM	5	2	0.97	0.97	270000	270000
3月	LCM	6	1	0.97	0.95	170000	270000
3月	LCM	6	2	0.97	0.97	270000	270000
4月	TFT	1	1	0.95	0.95	140000	250000
4月	TFT	1	2	0.95	0.97	240000	250000
4月	TFT	2	1	0.95	0.95	140000	250000
4月	TFT	2	2	0.95	0.97	240000	250000
4月	LCD	3	1	0.93	0.95	150000	270000
4月	LCD	3	2	0.93	0.97	250000	270000
4月	LCD	4	1	0.93	0.95	150000	270000
4月	LCD	4	2	0.93	0.97	250000	270000
4月	LCM	5	1	0.97	0.95	170000	270000
4月	LCM	5	2	0.97	0.97	270000	270000

4月	LCM	6	1	0.97	0.95	170000	270000
4月	LCM	6	2	0.97	0.97	270000	270000
5月	TFT	1	1	0.95	0.95	140000	250000
5月	TFT	1	2	0.95	0.97	240000	250000
5月	TFT	2	1	0.95	0.95	140000	250000
5月	TFT	2	2	0.95	0.97	240000	250000
5月	LCD	3	1	0.93	0.95	150000	270000
5月	LCD	3	2	0.93	0.97	250000	270000
5月	LCD	4	1	0.93	0.95	150000	260000
5月	LCD	4	2	0.93	0.97	250000	260000
5月	LCM	5	1	0.97	0.95	160000	270000
5月	LCM	5	2	0.97	0.97	260000	270000
5月	LCM	6	1	0.97	0.95	160000	270000
5月	LCM	6	2	0.97	0.97	260000	270000
6月	TFT	1	1	0.95	0.95	140000	250000
6月	TFT	1	2	0.95	0.97	240000	250000
6月	TFT	2	1	0.95	0.95	140000	250000
6月	TFT	2	2	0.95	0.97	240000	250000
6月	LCD	3	1	0.93	0.95	150000	260000
6月	LCD	3	2	0.93	0.97	250000	260000
6月	LCD	4	1	0.93	0.95	150000	260000
6月	LCD	4	2	0.93	0.97	250000	260000
6月	LCM	5	1	0.97	0.95	160000	270000
6月	LCM	5	2	0.97	0.97	260000	270000
6月	LCM	6	1	0.97	0.95	160000	270000
6月	LCM	6	2	0.97	0.97	260000	270000
7月	TFT	1	1	0.95	0.95	140000	250000
7月	TFT	1	2	0.95	0.97	240000	250000

7月	TFT	2	1	0.95	0.95	140000	250000
7月	TFT	2	2	0.95	0.97	240000	250000
7月	LCD	3	1	0.93	0.95	150000	260000
7月	LCD	3	2	0.93	0.97	250000	260000
7月	LCD	4	1	0.93	0.95	150000	260000
7月	LCD	4	2	0.93	0.97	250000	260000
7月	LCM	5	1	0.97	0.95	160000	270000
7月	LCM	5	2	0.97	0.97	260000	270000
7月	LCM	6	1	0.97	0.95	160000	270000
7月	LCM	6	2	0.97	0.97	260000	270000
8月	TFT	1	1	0.95	0.95	140000	250000
8月	TFT	1	2	0.95	0.97	240000	250000
8月	TFT	2	1	0.95	0.95	140000	250000
8月	TFT	2	2	0.95	0.97	240000	250000
8月	LCD	3	1	0.93	0.95	150000	260000
8月	LCD	3	2	0.93	0.97	250000	260000
8月	LCD	4	1	0.93	0.95	150000	260000
8月	LCD	4	2	0.93	0.97	250000	260000
8月	LCM	5	1	0.97	0.95	160000	270000
8月	LCM	5	2	0.97	0.97	260000	270000
8月	LCM	6	1	0.97	0.95	160000	270000
8月	LCM	6	2	0.97	0.97	260000	270000

表附錄 C.2 製造訂單的分配表 (線性規劃)

月份	品名	廠區	尺寸	投入量	月份	品名	廠區	尺寸	投入量
3月	TFT	1	1	125000	5月	TFT	2	1	125000
3月	TFT	1	2	125000	5月	TFT	2	2	125000
3月	TFT	2	1	125000	5月	LCD	3	1	130000
3月	TFT	2	2	125000	5月	LCD	3	2	130000
3月	LCD	3	1	130000	5月	LCD	4	1	130000
3月	LCD	3	2	130000	5月	LCD	4	2	130000
3月	LCD	4	1	130000	5月	LCM	5	1	135000
3月	LCD	4	2	130000	5月	LCM	5	2	135000
3月	LCM	5	1	135000	5月	LCM	6	1	135000
3月	LCM	5	2	135000	5月	LCM	6	2	135000
3月	LCM	6	1	135000	6月	TFT	1	1	125000
3月	LCM	6	2	135000	6月	TFT	1	2	125000
4月	TFT	1	1	125000	6月	TFT	2	1	125000
4月	TFT	1	2	125000	6月	TFT	2	2	125000
4月	TFT	2	1	125000	6月	LCD	3	1	130000
4月	TFT	2	2	125000	6月	LCD	3	2	130000
4月	LCD	3	1	130000	6月	LCD	4	1	130000
4月	LCD	3	2	130000	6月	LCD	4	2	130000
4月	LCD	4	1	130000	6月	LCM	5	1	135000
4月	LCD	4	2	130000	6月	LCM	5	2	135000
4月	LCM	5	1	135000	6月	LCM	6	1	135000
4月	LCM	5	2	135000	6月	LCM	6	2	135000
4月	LCM	6	1	135000	7月	TFT	1	1	125000
4月	LCM	6	2	135000	7月	TFT	1	2	125000
5月	TFT	1	1	125000	7月	TFT	2	1	125000
5月	TFT	1	2	125000	7月	TFT	2	2	125000

7月	LCD	3	1	130000	8月	TFT	2	1	125000
7月	LCD	3	2	130000	8月	TFT	2	2	125000
7月	LCD	4	1	130000	8月	LCD	3	1	130000
7月	LCD	4	2	130000	8月	LCD	3	2	130000
7月	LCM	5	1	135000	8月	LCD	4	1	130000
7月	LCM	5	2	135000	8月	LCD	4	2	130000
7月	LCM	6	1	135000	8月	LCM	5	1	135000
7月	LCM	6	2	135000	8月	LCM	5	2	135000
8月	TFT	1	1	125000	8月	LCM	6	1	135000
8月	TFT	1	2	125000	8月	LCM	6	2	135000

表附錄 C.3 運輸訂單的分配表 (線性規劃)

月份	品名	來源	消耗	數量	月份	品名	來源	消耗	數量
3月	TFT	1	3	150000	4月	LCD	3	6	150000
3月	TFT	2	3	150000	5月	TFT	1	3	150000
3月	TFT	2	4	150000	5月	TFT	2	3	150000
3月	TFT	1	4	150000	5月	TFT	2	4	150000
3月	LCD	3	5	150000	5月	TFT	1	4	150000
3月	LCD	4	5	150000	5月	LCD	3	5	150000
3月	LCD	4	6	150000	5月	LCD	4	5	150000
3月	LCD	3	6	150000	5月	LCD	4	6	150000
4月	TFT	1	3	150000	5月	LCD	3	6	150000
4月	TFT	2	3	150000	6月	TFT	1	3	150000
4月	TFT	2	4	150000	6月	TFT	2	3	150000
4月	TFT	1	4	150000	6月	TFT	2	4	150000
4月	LCD	3	5	150000	6月	TFT	1	4	150000
4月	LCD	4	5	150000	6月	LCD	3	5	150000
4月	LCD	4	6	150000	6月	LCD	4	5	150000

6月	LCD	4	6	150000	7月	LCD	3	6	150000
6月	LCD	3	6	150000	8月	TFT	1	3	150000
7月	TFT	1	3	150000	8月	TFT	2	3	150000
7月	TFT	2	3	150000	8月	TFT	2	4	150000
7月	TFT	2	4	150000	8月	TFT	1	4	150000
7月	TFT	1	4	150000	8月	LCD	3	5	150000
7月	LCD	3	5	150000	8月	LCD	4	5	150000
7月	LCD	4	5	150000	8月	LCD	4	6	150000
7月	LCD	4	6	150000	8月	LCD	3	6	150000

表附錄 C.4 存貨單的分配表 (線性規劃)

月份	品名	廠區	存貨量	月份	品名	廠區	存貨量
3月	TFT	1	0	5月	LCM	5	344000
3月	TFT	2	0	5月	LCM	6	344000
3月	LCD	3	0	6月	TFT	1	0
3月	LCD	4	108000	6月	TFT	2	0
3月	LCM	5	0	6月	LCD	3	622000
3月	LCM	6	0	6月	LCD	4	730000
4月	TFT	1	0	6月	LCM	5	658000
4月	TFT	2	0	6月	LCM	6	657000
4月	LCD	3	108000	7月	TFT	1	0
4月	LCD	4	216000	7月	TFT	2	0
4月	LCM	5	120000	7月	LCD	3	1034000
4月	LCM	6	120000	7月	LCD	4	1141000
5月	TFT	1	0	7月	LCM	5	1082000
5月	TFT	2	0	7月	LCM	6	1081000
5月	LCD	3	320000	8月	TFT	1	0
5月	LCD	4	428000	8月	TFT	2	0

8月	LCD	3	1530000	8月	LCM	5	1589000
8月	LCD	4	1637000	8月	LCM	6	1588000

表附錄 C.5 訂單開立表 (線性規劃)

月份	品名	尺寸	數量	類型	月份	品名	尺寸	數量	類型
3月	LCM	1	169000	積壓	6月	LCM	1	31000	填補
3月	LCM	2	169000	積壓	6月	LCM	2	31000	填補
4月	LCM	1	11000	填補	7月	LCM	1	111000	填補
4月	LCM	2	11000	填補	7月	LCM	2	111000	填補
5月	LCM	1	69000	積壓	8月	LCM	1	9000	積壓
5月	LCM	2	69000	積壓	8月	LCM	2	9000	積壓

表附錄 C.6 供需平衡的挪移表 (線性規劃)

積壓訂單				填補產能訂單			
月份	品名	尺寸	數量	月份	品名	尺寸	數量
3月	LCM	1	169000	6月	LCM	1	31000
				7月	LCM	1	111000
3月	LCM	2	169000	6月	LCM	2	31000
				7月	LCM	2	111000
8月	LCM	1	9000	4月	LCM	1	11000
8月	LCM	2	9000	4月	LCM	2	11000

表附錄 C.7 製造訂單的分配表 (啟發式法則)

月份	品名	廠區	尺寸	投入量	月份	品名	廠區	尺寸	投入量
3月	TFT	1	1	10000	5月	TFT	2	1	10000
3月	TFT	1	2	240000	5月	TFT	2	2	240000
3月	TFT	2	1	10000	5月	LCD	3	1	10000
3月	TFT	2	2	240000	5月	LCD	3	2	250000
3月	LCD	3	1	10000	5月	LCD	4	1	10000
3月	LCD	3	2	250000	5月	LCD	4	2	250000
3月	LCD	4	1	10000	5月	LCM	5	1	10000
3月	LCD	4	2	250000	5月	LCM	5	2	260000
3月	LCM	5	1	10000	5月	LCM	6	1	10000
3月	LCM	5	2	260000	5月	LCM	6	2	260000
3月	LCM	6	1	10000	6月	TFT	1	1	10000
3月	LCM	6	2	260000	6月	TFT	1	2	240000
4月	TFT	1	1	10000	6月	TFT	2	1	10000
4月	TFT	1	2	240000	6月	TFT	2	2	240000
4月	TFT	2	1	10000	6月	LCD	3	1	10000
4月	TFT	2	2	240000	6月	LCD	3	2	250000
4月	LCD	3	1	10000	6月	LCD	4	1	10000
4月	LCD	3	2	250000	6月	LCD	4	2	250000
4月	LCD	4	1	10000	6月	LCM	5	1	10000
4月	LCD	4	2	250000	6月	LCM	5	2	260000
4月	LCM	5	1	10000	6月	LCM	6	1	10000
4月	LCM	5	2	260000	6月	LCM	6	2	260000
4月	LCM	6	1	10000	7月	TFT	1	1	10000
4月	LCM	6	2	260000	7月	TFT	1	2	240000
5月	TFT	1	1	10000	7月	TFT	2	1	10000
5月	TFT	1	2	240000	7月	TFT	2	2	240000

7月	LCD	3	1	10000	8月	TFT	2	1	10000
7月	LCD	3	2	250000	8月	TFT	2	2	240000
7月	LCD	4	1	10000	8月	LCD	3	1	10000
7月	LCD	4	2	250000	8月	LCD	3	2	250000
7月	LCM	5	1	10000	8月	LCD	4	1	10000
7月	LCM	5	2	260000	8月	LCD	4	2	250000
7月	LCM	6	1	10000	8月	LCM	5	1	10000
7月	LCM	6	2	260000	8月	LCM	5	2	260000
8月	TFT	1	1	10000	8月	LCM	6	1	10000
8月	TFT	1	2	240000	8月	LCM	6	2	260000

表附錄 C.8 運輸訂單的分配表 (啟發式法則)

月份	品名	來源	消耗		月份	品名	來源	消耗	數量
3月	TFT	1	3	156000	4月	LCD	3	6	0
3月	TFT	2	3	0	5月	TFT	1	3	116000
3月	TFT	2	4	156000	5月	TFT	2	3	0
3月	TFT	1	4	0	5月	TFT	2	4	116000
3月	LCD	3	5	260000	5月	TFT	1	4	0
3月	LCD	4	5	0	5月	LCD	3	5	260000
3月	LCD	4	6	260000	5月	LCD	4	5	0
3月	LCD	3	6	0	5月	LCD	4	6	260000
4月	TFT	1	3	134000	5月	LCD	3	6	0
4月	TFT	2	3	0	6月	TFT	1	3	94000
4月	TFT	2	4	134000	6月	TFT	2	3	0
4月	TFT	1	4	0	6月	TFT	2	4	94000
4月	LCD	3	5	260000	6月	TFT	1	4	0
4月	LCD	4	5	0	6月	LCD	3	5	260000
4月	LCD	4	6	260000	6月	LCD	4	5	0

6月	LCD	4	6	260000	7月	LCD	3	6	0
6月	LCD	3	6	0	8月	TFT	1	3	57000
7月	TFT	1	3	77000	8月	TFT	2	3	0
7月	TFT	2	3	0	8月	TFT	2	4	57000
7月	TFT	2	4	77000	8月	TFT	1	4	0
7月	TFT	1	4	0	8月	LCD	3	5	260000
7月	LCD	3	5	260000	8月	LCD	4	5	0
7月	LCD	4	5	0	8月	LCD	4	6	260000
7月	LCD	4	6	260000	8月	LCD	3	6	0

表附錄 C.9 存貨單的分配表 (啟發式法則)

月份	品名		存貨量	月份	品名	廠區	存貨量
3月	TFT	1	0	5月	LCM	5	1811000
3月	TFT	2	0	5月	LCM	6	1811000
3月	LCD	3	194000	6月	TFT	1	0
3月	LCD	4	301000	6月	TFT	2	0
3月	LCM	5	456000	6月	LCD	3	1463000
3月	LCM	6	456000	6月	LCD	4	1571000
4月	TFT	1	0	6月	LCM	5	2629000
4月	TFT	2	0	6月	LCM	6	2628000
4月	LCD	3	517000	7月	TFT	1	0
4月	LCD	4	625000	7月	TFT	2	0
4月	LCM	5	1080000	7月	LCD	3	2092000
4月	LCM	6	1080000	7月	LCD	4	2199000
5月	TFT	1	0	7月	LCM	5	3560000
5月	TFT	2	0	7月	LCM	6	3559000
5月	LCD	3	946000	8月	TFT	1	0
5月	LCD	4	1054000	8月	TFT	2	0

8月	LCD	3	2804000	8月	LCM	5	4573000
8月	LCD	4	2911000	8月	LCM	6	4571000

表附錄 C.10 訂單開立表 (啟發式法則)

月份	品名	尺寸	數量	類型		品名	尺寸	數量	類型
3月	LCM	1	420000	積壓	6月	LCM	1	220000	積壓
3月	LCM	2	80000	填補	6月	LCM	2	280000	填補
4月	LCM	1	240000	積壓	7月	LCM	1	140000	積壓
4月	LCM	2	260000	填補	7月	LCM	2	360000	填補
5月	LCM	1	320000	積壓	8月	LCM	1	260000	積壓
5月	LCM	2	180000	填補	8月	LCM	2	240000	填補

簡歷

姓 名：王世欽 (Shih-Chin Wang)

籍 貫：台灣省台南縣

生 日：民國 65 年 11 月 7 日

學 歷：民國 89 年東海大學工業工程學系畢

民國 91 年東海大學工業工程與經營資訊研究所畢

聯絡處：台南市東區崇德二十六街 3 號二樓

電 話：(06) 3361495

