

東 海 大 學

工業工程與經營資訊研究所

碩士論文

考量超時作業成本下物流中心

最佳補貨排程之研究

研 究 生：黃 綉 菱

指 導 教 授：姚 銘 忠 博 士

黃 嘉 彥 博 士

中 華 民 國 九 十 六 年 七 月

**On the optimal replenishment scheduling for a  
distribution center with overtime operational costs**

By

Hsiu-Ling Huang

Advisor: Prof. Ming-Jong Yao

Prof. Jia-Yen Huang

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 and Enterprise Information

July 2007

Taichung , Taiwan , Republic of China

# 考量超時作業成本下物流中心最佳補貨排程之研究

學生：黃綉菱

指導教授：姚銘忠

黃嘉彥

東海大學工業工程與經營資訊研究所

## 摘要

物流中心在供應鏈體系中扮演物料集散與配銷的角色，主要功能係將來自供應商的產品集中分類檢貨後，並分散轉運至各個零售商。過去在一般物流中心配送排程問題的模式中，大多假設物流中心內部的產能為無限，物流中心的內部排程皆能即時滿足顧客的需求。然而在現實的狀況，由於人員一般有工作時間的限制，人員的加班會產生額外的人工成本，若不考量加班時間的因素探討，則無法與實務的狀況相互配合，且會誤導零售業者經理人的決策。本研究運用二冪策略 (Power-of-Two policy)；即假設供應鏈體系中，由供應商到達物流中心（進站路線）的週期時間與由物流中心到達各個零售商（離站路線）的週期時間，為二冪整數乘上一個基本週期(basic period)。本研究在考量物流中心人員在超過產能限制時會發生超時作業成本的前提下，推導本供應鏈系統的數學模式，並推出遺傳演算法與啟發式演算法(heuristic) 求得進站路線與離站路線之最佳發車週期時間，並規劃補貨排程以降低物流中心作業的尖峰負載量以減少超時作業成本，使供應鏈系統的平均總成本達到最小。數據實驗的顯示，本研究提出的兩個解法，可以有效求得品質優良的解答。

**關鍵字詞：**物流中心、二冪策略、遺傳演算法、啟發式演算法

# **On the optimal replenishment scheduling for a distribution center with overtime operational costs**

Student : HSIU-Ling Huang

Advisor : Ming-Jong Yao  
Jia-Yen Huang

Department of Industrial Engineering and Enterprise Information  
Tunghai University

## **ABSTRACT**

In the supply chain system, the distribution center (DC) plays a role of collecting products from the suppliers and distributing them to the retailers after the operations of sorting and picking. Most of the studies on the lot scheduling of the DC did not take into account the limit of capacity and assumed that the DC was able to fulfill the scheduled operations satisfactorily. However, there is only limited capacity for the operations in the DC and it usually incurs an over-time cost when the capacity load exceeds the limit. Therefore, if the over-time cost is not considered in the scenario, the managers' decision-making could be seriously biased. In this study, we employ the "Power-of-Two" (PoT) policy in which the arrival cycle time (arrival from a supplier to the DC) and the departure cycle time (departing the DC to a retailer) must be a PoT-integer multiple times of a basic period. By taking into accounts the over-time cost, we derive a mathematical model for the supply chain system and propose two solution approaches, namely, a genetic algorithm and a heuristic. Our solution approaches not only obtains the optimal arrival and departure cycle times of the transportation fleet, but also reduces the peak load in the DC to decrease the possible over-time cost to minimize total average total costs incurred in the supply chain system. Our numerical results show that our solution approaches are able to obtain a close-to-optimal solution within a reasonable run time.

**Keywords : distribution center, power-of-two policy, genetic algorithm, heuristic**

## 致 謝

在東海兩年的研究生活中，種種回憶令人感觸良多。首先誠摯的感謝指導教授姚銘忠教授，姚老師教給我的除了是課業上的指導之外，更教導了我許多生活上的態度與做人做事的方法，給予我許多的啟發，而且老師對學問的嚴謹更是我學習的典範。另外，也相當感謝黃嘉彥老師每星期特地到東海大學給予論文上的建議，在完成此論文的過程中給予我很多的指導與正面的鼓勵。也感謝口試委員邱天嵩老師於口試期間所給予的建議與指正，謝謝老師的不吝惜指教，使這篇論文能夠更加完備。本論文的完成另外亦得感謝的係上的助教宏華以及瑋珊，因為你們的鼓勵，讓我可以繼續堅持下去，也謝謝你們偶爾充當我的心理輔導老師。

感謝已經畢業的紹維、折鋒、昕煒和智鈞學長總能在我迷惘時為我解惑，最感謝的是好姐妹美翎，研究所這兩年多虧你的陪伴，大家一起熬夜寫程式、解決課業上的困難，以及生活上面的互相扶持；也感謝一群好朋友們，Gaho、古、茜雯、豪哥、永順，研究所的生活因為有你們增色不少。還有研究室的何陽、政益、豪哲以及正偉和亦茵，你們的幫忙及搞笑我銘感在心。兩年裡的日子，實驗室裡共同的生活點滴，學術上的討論、言不及義的閒扯、讓人又愛又怕的宵夜、趕作業的革命情感，感謝眾位學長姐、同學、學弟妹的共同砥礪，你們的陪伴讓兩年的研究生活變得絢麗多彩。

最感謝的莫過於家人，沒有父母親和哥哥的全力的支持與你們的鼓勵與包容，我無法堅持到現在，誠摯的感謝我的家人，僅以此論文獻給我最親愛的家人及所有關心我的朋友。

黃綉菱 謹誌於

東海大學工業工程與經營資訊研究所

中華民國九十六年

# 目錄

摘要.....	I
ABSTRACT.....	II
致 謝.....	III
目 錄.....	IV
圖目錄.....	VII
表目錄.....	VIII
符號定義.....	1
第一章 緒論.....	2
1.1 研究背景與動機.....	2
1.2 研究目的.....	3
1.3 研究流程與架構.....	4
第二章 文獻探討.....	6
2.1 物流中心作業.....	6
2.2 物流與運輸系統中的人力排程.....	7
2.3 車輛排程問題.....	9
2.4 文獻小結.....	12
第三章 數學模式.....	13
3.1 問題描述.....	13
3.2 限制條件與決策參數.....	15
3.2.1 基本假設.....	15
3.2.2 條件限制.....	15
3.2.3 決策變數.....	15
3.3 零售配銷體系中的各個成本項.....	16
3.3.1 零售配銷體系之進貨端成本.....	16
3.3.2 物流中心成本.....	20
3.2.3 零售配銷體系之出貨端成本.....	23
3.4 零售配銷體系下之數學模式.....	25

第四章 運用遺傳演算法求解 .....	27
4.1 遺傳演算法架構.....	27
4.1.1 產生初始母代與染色體之編碼(Encoding) .....	27
4.1.2 解碼(Decoding) .....	30
4.1.3 尋找基本週期(Basic period) .....	31
4.1.4 評估適應力函數值(Fitness Evaluation).....	34
4.1.5 選擇.....	35
4.1.6 交配.....	36
4.1.7 突變.....	38
4.1.8 參數設定 .....	39
4.2 遺傳演算法求解過程 .....	39
4.2.1 演算範例之產生初始母代與編碼.....	40
4.2.2 演算範例之間距解碼.....	41
4.2.3 演算範例之尋找基本週期.....	42
4.2.4 演算範例之評估適應值與選擇染色體.....	42
4.2.5 演算範例之交配 .....	43
4.2.6 演算範例之突變 .....	44
4.3 小結 .....	46
第五章 分裝作業排程的啟發式解法 .....	47
5.1 ProcOTM 介紹 .....	47
5.2 初步排程程序(ProcIS).....	49
5.3 產品排程程序(ProcPS).....	52
5.4 排程撫平程序(ProcSS).....	54
5.5 加班成本最小化演算程序 .....	55
第六章 數據範例.....	62
6.1 研究工具.....	62
6.2 數據實驗.....	62
6.3 隨機實驗.....	71

第七章 結論與未來研究方向 .....	73
7.1 結論與建議 .....	73
7.2 未來研究方向 .....	73
參考文獻 .....	74
附錄 A 車輛選擇步驟 .....	77
附錄 B 物流中心之平均存貨成本 .....	79
附錄 C 成本函數簡化 .....	80
附錄 D ProcOTM 完整範例說明 .....	81

## 圖目錄

圖 1.1 研究步驟流程圖.....	5
圖 2.1 物流中心作業系統.....	6
圖 2.2 每日作業分佈圖.....	9
圖 3.1 零售配銷體系成員.....	13
圖 3.2 車輛選擇流程圖.....	19
圖 3.3 供應商 1 對零售賣場 3 之存貨水準.....	22
圖 3.4 零售賣場存貨水準.....	24
圖 4.1 染色體的組成.....	27
圖 4.2 遺傳演算法流程圖.....	28
圖 4.3 遺傳演算法之虛擬碼.....	29
圖 4.4 染色體解碼.....	31
圖 4.5 利用共同間距法尋找基本週期之上下界.....	33
圖 4.6 輪盤選擇機制.....	36
圖 4.7 選擇交配之染色體示意圖.....	37
圖 4.8 單點交配示意圖.....	37
圖 4.9 雙點交配示意圖.....	38
圖 4.10 均勻交配示意圖.....	38
圖 4.11 遺傳演算法之突變運作.....	39
圖 4.12 範例中選擇交配之染色體示意圖.....	44
圖 4.13 均勻交配.....	44
圖 4.14 突變機制.....	45
圖 5.1 初步排程程序流程圖.....	51
圖 5.2 產品排程程序流程圖.....	53
圖 5.3 移出程序流程圖.....	58
圖 5.4 成對互換程序流程圖.....	59
圖 5.5 二對一互換程序流程圖.....	60
圖 5.6 加班成本最小化演算程序流程圖.....	61
圖 6.1 結合遺傳演算法與分裝作業排程流程圖.....	63

## 表目錄

表 3.1 貨車類型與費用.....	18
表 3.2 物流中心之淨產品量與存貨水準.....	21
表 3.3 供應商 1 對零售商 3 之淨進貨量與累積存貨水準.....	22
表 4.1 第一條染色體與各基本週期之總成本.....	34
表 4.2 適應值函數.....	35
表 4.3 零售體系下各供應商與零售賣場之供給需求.....	40
表 4.4 貨車載重量與貨車費用.....	40
表 4.5 零售配銷系統之參數設定值.....	40
表 4.6 遺傳演算法之相關參數設定.....	40
表 4.7 初始母體矩陣.....	41
表 4.8 進站間距解碼.....	42
表 4.9 範例間距之染色體解碼.....	42
表 4.10 母體中各染色體適應值函數.....	43
表 4.11 經 100 代演化後之目標函數值與基本週期.....	45
表 5.1 分裝作業排程矩陣示意圖.....	48
表 5.2 分裝作業排程.....	49
表 6.1 第一代的第九條染色體解碼後之間距乘數.....	64
表 6.2 單位產品平均花費時間(單位：分/kg).....	64
表 6.3 第一代染色體之目標函數值.....	65
表 6.4 參照表格.....	66
表 6.5 第一代染色體之適應力函數值.....	67
表 6.6 本研究執行 100 代之結果.....	69
表 6.7 黃紹維(2006)執行 100 代之結果(未考慮人員產能限制).....	70
表 6.8 將黃紹維(2006)成本項加入超時作業成本.....	71
表 A.1 載重量 1000kg 之車輛選擇.....	78
表 D.1 設定起始參數.....	81
表 D.2 部分初始生產排程.....	81
表 D.3 利用 ProcIS 的初始排程.....	82
表 D.4 移出程序之移出排程.....	86
表 D.5 移出程序排列後的排程.....	86
表 D.6 成對互換移入排程.....	87
表 D.7 成對互換排列後的排程.....	88
表 D.8 欲進行二對一的初始作業排程.....	88
表 D.9 二對一移入的排程.....	89
表 D.10 二對一排列後的排程.....	90
表 D.11 最終排程.....	91

## 符號定義

$Q_i^a$	上游供應商 $i$ 供給率
$k_i^a B$	進站路線 $i$ 的進站間距為
$k_i^a B Q_i^a$	進站路線 $i$ 的產品載運至物流中心的總量
$X_i$	物流中心作業人員將進站路線 $i$ 之產品由貨卡車卸下並進行揀貨分裝，每單位產品平均花費時間
$Q_j^d$	下游零售賣場 $j$ 需求率
$k_j^d B$	離站路線 $j$ 的離站間距
$X_j$	離站路線 $j$ 依零售賣場 $j$ 的補貨需求進行揀貨並裝載上車，每單位產品平均花費時間
$k_i^a$	進站路線間距乘數
$k_j^d$	離站路線間距乘數
$B^*$	零售配銷系統之最佳基本週期
$P_x$	向第三方物流業者租借貨車的費用
$V_i^a$	貨車行駛時間
$Q_i^a$	載運的產品量
$h^s$	產品時間之單位在途成本
$A$	每進站碼頭設備之使用成本(固定成本)
$X$	每單位產品的處理成本 (變動成本)
$q_{ij}$	零售賣場 $j$ 對供應商 $i$ 的單位產品需求量
$h^r$	單位產品時間之庫存成本
$P_i^a$	進站間距的乘冪次方
$P_j^d$	離站間距 $k_j^d$ 的乘冪次方

# 第一章 緒論

## 1.1 研究背景與動機

在電子商務時代由於消費者對速度的要求與重視，企業必須借助一個專業且有效率的物流配送服務，達成少量多樣、宅配送達、貨物追蹤及快速退換貨等服務。台灣企業多屬於中小型企業，難以設立屬於自己的物流中心，故常將物流作業外包給專業的「第三方物流中心」，如此企業則可以專注於生產、行銷及研發等活動，發展自己的核心競爭力。

董福慶和陳明德(1995)研究中指出：「根據統計，一件產品最後售價的30%是來自於物流的成本費用」。物流中心基本作業流程主要包括進貨、搬運、儲存、盤點、訂單處理、揀貨、補貨、出貨及配送等九項作業。根據 Hwang (1988) 的研究得知，物流作業之成本結構中，揀貨成本約佔了40%。當訂購之產品從碼頭卸貨之後，物流中心的人員將產品品項進行點收完到將其進行上架儲存前，因為這些產品分別要送到各個不同的零售端及客戶手上，因此需將產品以同類或同一個需求端的方式先做分類的動作。待訂單來時依訂單需求揀貨，最後將產品出貨配送完成作業。物流作業當中最花時間的除了分類作業之外，最主要的就是揀貨作業。根據 Hwang (1988) 研究得知：揀貨作業是直接相關人力投入，約佔整體物流中心投入人力的50%左右；在整體物流中心的作業時間當中，揀貨的時間亦佔30%~40%。綜合以上得知，不論從成本、人力，或時間的角度分析，都顯示揀貨作業之重要性。從研究中不難發現，「人力」是倉儲中心內最主要的作業瓶頸，故必須先妥善處理產能排程，方能適當的解決物流中心配送排程問題。

台灣零售賣場為了民眾購買方便，一般將賣場設在市中心，零售業者只關心所訂購之產品是否可以準時送達，並未針對車輛的排班問題詳加考量。由於台灣地區地小人稠道路狹隘，車輛進入零售賣場的時間接續在同一點，容易造成鄰近地區塞車的現象。貨卡車業者為了能準時進入到零售業者原先排定的隊伍當中相對花費許多的等待時間，造成機會成本而無法做以其他的用途，如此的問題在美國較不會發生，但依台灣之國情確實有實務考量之必要。因此，如何有效進行貨物的配送規劃，為物流業者一重要之課題。

產品由工廠端製造完之後，經由車隊將下游顧客(或供應商)所需的先行載運至物流中心進行集散與配銷，物流中心根據顧客所下的訂單，經由一連串

的接收、儲存、揀取、運送等作業，將產品作重新分類及包裝後，把各個零售賣場所需之數量及品項裝載上車，送達顧客的手中。在這一連串的物流作業當中，零售體系主管的目標為在交期時間內將物品送達顧客的手中且使平均總成本達到最小，並在人力及車輛等產能有限的狀況下，其排程為可行。

物流中心在供應鏈體系中扮演著物料集散與配銷的角色，主要功能係將產品集中並分散至零售業。根據 Hwang (1988) 的研究得知，揀貨作業其相關直接人力，占整體物流人力的 50% 以上，在物流中心內部作業的運作過程中，揀貨作業約佔總物流中心作業成本的 40%，因此規劃合理的人力資源排程，以改善揀貨作業績效，對提升物流中心整體的營運績效具有決定性的影響。此外，如未能有良好的車輛排程造成貨卡車業者額外等候的機會成本，對企業而言也是一另外的負擔。當規劃一有效率之車輛排程，搭配人員揀貨產能的調配，兩者互相搭配，才能適當的將揀貨作業完成並將物品送至顧客端。本研究有鑑於此，希望針對企業內部的的人員作業瓶頸問題及外部的車輛排班問題做有效率的規劃排程，讓整體作業流程得以順暢運行。

## 1.2 研究目的

目前在台灣地區零售賣場相當普及，賣場所販售商品眾多，舉凡如生活用品、食品、電器... 等等，這些商品來自各個不同供應商，而同種類的商品又有多家供應商提供商品，增加了管理的複雜性，同時也增加了公司的管理成本。且對企業而言，要維持一個完整的運輸車隊，每個月除了要支付駕駛人員的薪資外，公司還需要對貨車進行固定維修保養，這些問題都會讓營運成本居高不下，再加上少量的產品無法符合運輸經濟效益。因此決策者就需要衡量每次的訂購時機以及如何協調運輸車輛的排程，讓每次的訂購成本為最經濟，負責運輸的車輛都能夠發揮其最適的運輸效益，使得企業能夠決策下獲得最大利益。

為了解決物流中心配送排程問題，本研究運用遺傳演算法(Genetic Algorithms ; GA)可多點平行搜尋的優點求得進站路線與離站路線之最佳發車間距乘數週期，同時考量物流中心人員產能資源限制，運用另一個啟發式演算法(heuristic)降低物流中心作業的尖峰負載量，使得物流中心總加班工時的人工成本最小。

身為一位零售體系的決策經理人而言，在探討零售配銷體下的供應商端發

生的成本、物流中心的營運成本及零售賣場端發生的成本下，考量發車間距和規劃貨車到達物流中心作業的排程，排定物流與人員作業的排程，且在物流中心既有人力資源下，應如何排定人員與車輛的排程，在滿足零售端的客戶需求的條件下，能達成平均總成本最小的目標。

### 1.3 研究流程與架構

本研究論文的內容共分七章，如圖 1.1 所示。第一章簡述本研究之研究動機、研究目的與問題的定義。第二章整理及探討與本研究相關的文獻，包括物流中心的作業，並自其中探討物流及運輸系統中的人力排程及車輛運輸問題。第三章對於本研究探討的整個零售配銷系統，推導所有可能發生的成本並完整建立數學模式。第四章介紹運用遺傳演算法求解本研究的最佳配銷與排程。因考量物流中心人工產能限制，因此在第五章提出啟發式方法，以降低物流源自於超時作業的人工成本最小。第六章進行數據實驗驗證本研究所提出之遺傳演算法，可以在合理的運算時間之內，獲致品質優良的解。第七章提出本研究之結論及未來研究之方向與建議。

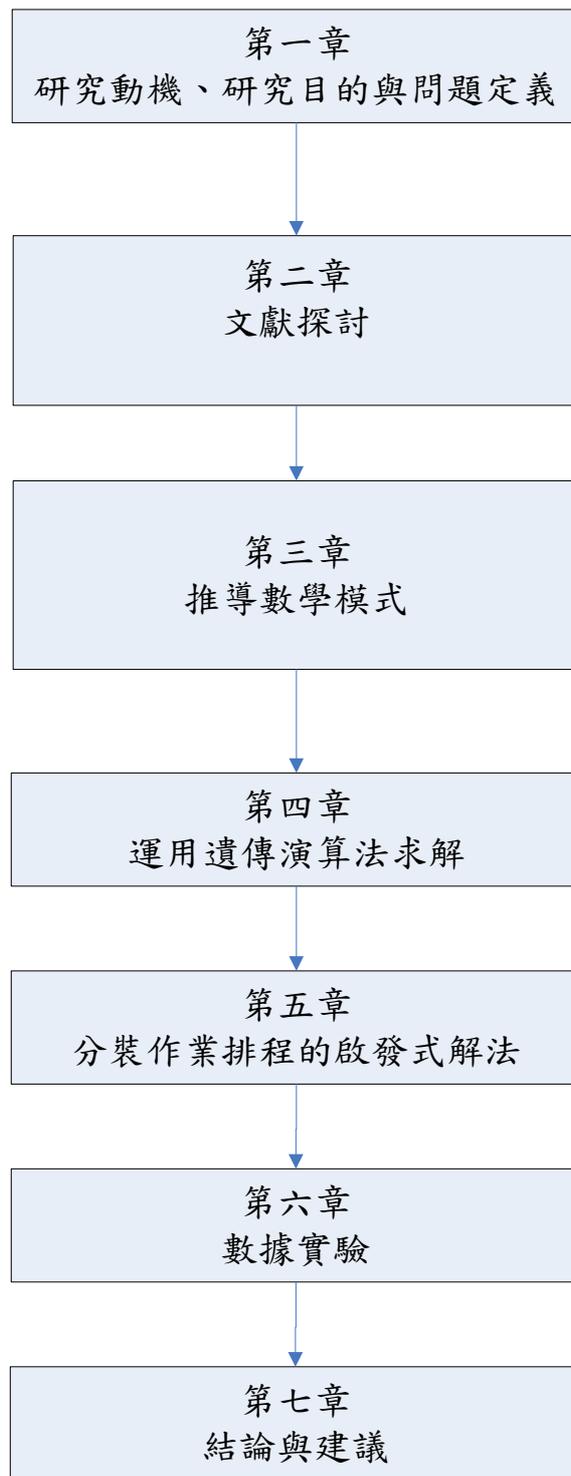


圖 1.1 研究步驟流程圖

## 第二章 文獻探討

在本章節中，第 2.1 節探討物流中心相關文獻，並指出物流中心的核活動為倉儲及運輸。第 2.2 節為物流與運輸系統中的人力排程相關文獻，第 2.3 節探討車輛排程問題相關的文獻，最後在第 2.4 節提出小結。

### 2.1 物流中心作業

經濟部商業司定義物流中心如下：物流中心大致針對銷售物流而言，為了使銷售過程作更有效處理而設置的單位。故凡從事將商品由製造商或進口商運送至零售商的中間流通業者，有連結上游製造業至下游消費者，滿足消費者多樣少量的市場需求、縮短流通通路，降低流通成本等關鍵性機能即可稱之為「物流中心」(經濟部商業司，1996)。

物流中心每日需處理大量的配銷作業，揀取並分類眾多的品項及大量的產品，必須以最有效率的方式滿足客戶訂單需求。經濟部商業司(1996)將物流中心主要作業分為進貨、搬運、儲存、盤點、訂單處理、揀貨、補貨、出貨及配送作業等九項，如圖2.1所示。

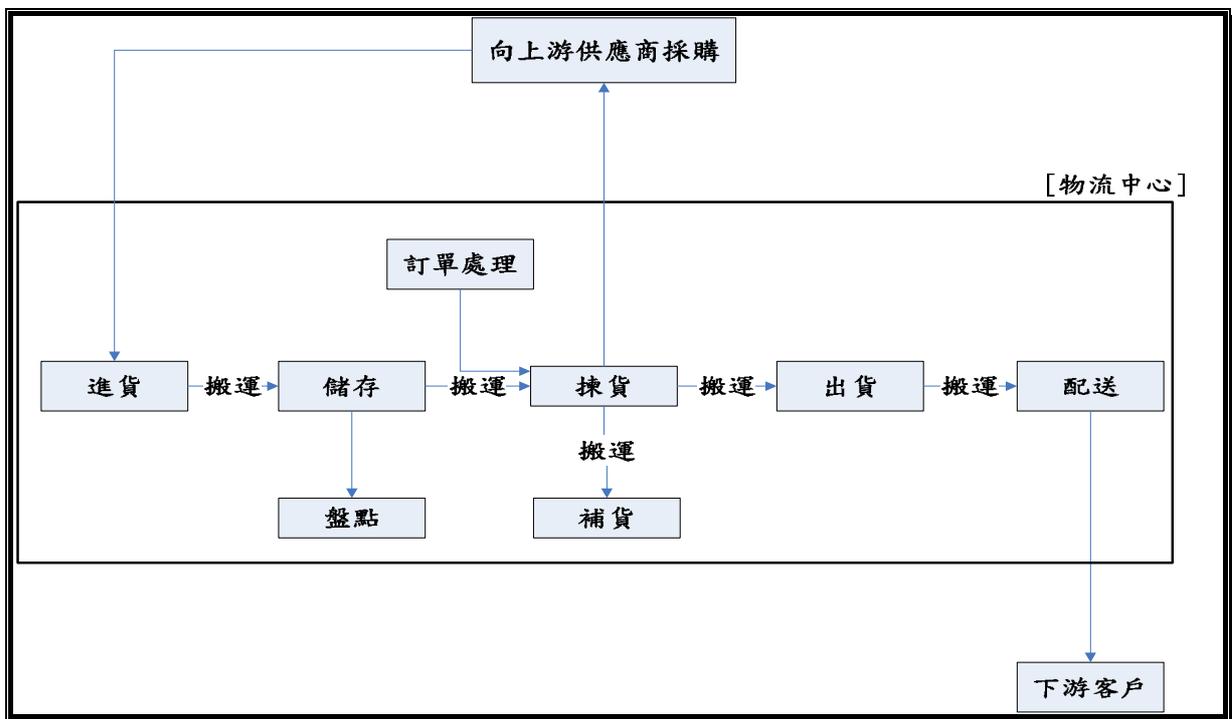


圖 2.1 物流中心作業系統

基本上，物流中心具備四項核心的作業機能：接收產品、儲存產品、揀取產品、運送產品(Tompkins，1998)。其中，儲存產品的存貨成本及揀取產品

的人力成本為物流中心兩個主要的成本項目(Kevin,2001)。根據 Hwang(1988)的研究得知，揀貨作業約佔了物流中心投入人力 50%左右，亦佔了大約 30%~40%物流中心作業時間。董福慶、陳明德(1995)對揀貨功能描述如下：「揀貨係基於在接受訂單的商業活動中，將顧客的訂購品從庫存儲位中選出，並進行出庫之業務」。揀貨作業的目的為如何在無揀貨錯誤率的情況下將正確的產品及正確的數量，在正確的時間內，及時配送給顧客，是揀貨作業最終的目的及功能。

訂單揀貨作業是物流中心內部一個最重要的決定性任務(Lin,1999)，物流中心雖有許多自動化的機械設備應可節省人力，但仍有諸多工作如運送、車輛調度、拆箱改包裝、揀貨、貼標籤等等作業均屬勞力密集的工作，仍需仰賴第一線理貨及搬運人員。因此，規劃合理的人力資源策略，對提升物流中心整體營運績效具有決定性的影響。

## 2.2 物流與運輸系統中的人力排程

本研究關心的問題是如何在滿足顧客需求的條件下規劃物流中心人力排程，使其活動、運作的作業平均總成本達到最小。良好的人員排班方式，會影響企業的相關成本支出與工作效率。近年來相關的研究相當多，研究的對象層面也相當廣。

Bard(2005)在探討人力排程的問題時，提出對信件處理及配送中心這類的服務性質企業來說，應根據需求預測調整排程。人員排班依產業或複雜度的不同，所需的排班規則有所不同，求解方法亦有差異。根據 Beasley and Cao (1996) 將人員排班依其複雜度，大致分為大型問題、中型問題及小型的一般人員排班問題。

### (1) 大型人員排班問題

大型人員排班問題以航空公司的人員排班最具代表性。由於航空公司的機組人員眾多(包括飛行員、機務員及空服員)、航線複雜、機隊龐大等因素，其限制條件繁雜，所以此等問題的規模較龐大難解。

### (2) 中型人員排班問題

中型人員排班問題，研究大多以大眾運輸人員排班最多，尤以火車、捷運或公車此等大眾運輸系統的排班較為常見。由於大眾運輸具有旅客容

量、運輸線路、列車交會、轉乘、站台等限制，因此對人員排班的需求格外迫切。

### (3) 小型人員排班問題

小型人員排班問題，大多是指從事服務業所需的工作人員排班，例如：警察、護士或是工廠作業人員之排班，由於一般人員排班有人力、機具、休假等限制，但又不如航空公司及大眾運輸人員排班來的複雜。因此，一般人員排班多屬中、小型排班問題。

本研究之物流中心人員排班的問題應屬於小型人員排班問題，在即時滿足顧客需求的條件下，希望藉由適當的人力排程降低尖峰負載量，以求得物流中心總加班工時的人工成本最小。

陳軍凱(2004)考量以往航空貨運公司對收貨人員排班大多以人工經驗進行，然而實際營運上若每日人力需求隨機性過大，則可能使原規劃人員供給班表失去其最佳性。因此在考量現實環境中貨運量變動的不確定因素，針對長期規劃與短期規劃上不同之限制條件，使用模擬技巧與 CPLEX 數學規劃軟體規劃最佳的人員供給班次表。

另外，在排定人力排程的過程當中，應考量每天不同時段所需的人力及貨物進入中心的狀況。Frazelle(2002)在「World-Class Warehousing and Material Handling」一書當中指出在物流倉儲中心當中每天的作業分佈，如圖 2.2 所示。每日作業分佈(daily activity distribution)可以顯示出進貨、儲存、揀貨及出貨各項作業中在一天中不同時段的尖離峰狀態，因此我們可以依照尖峰作業期的需求來設計，為平衡尖峰工作量，可以進行工作排班的調度、或跨部門人力的調度。不同的時段應該依據真實的狀況，對物流中心內產品及人力需求在相對的排程。

人員排班問題中，不論是組員排班問題或是指派問題，一般多定義為二元整數或混合整數規劃問題。例如，Bartholdi(1981)已證明人員排班問題為 NP-complete 問題，故針對大型問題，難以在合理的運算時間內求得最佳解。為能在短時間內有效率地找到近似最佳解，一般多採用啟發式解法求解。在人員排班求解方法，一般可分為啟發式解法與整數規劃法兩大類，過去一般求解整數線性規劃法(Integer Linear Programming; ILP)的商用軟體，如：LINGO 所使用的分支界線法(branch and bound)常造成過度資料簿記負載問題，導致求解較無效率。而使用遺傳演算法(GA)需進行選擇、交配及突變等

運算過程，且經過多世代的演進，亦需耗費較多時間求得一組作業排程解。

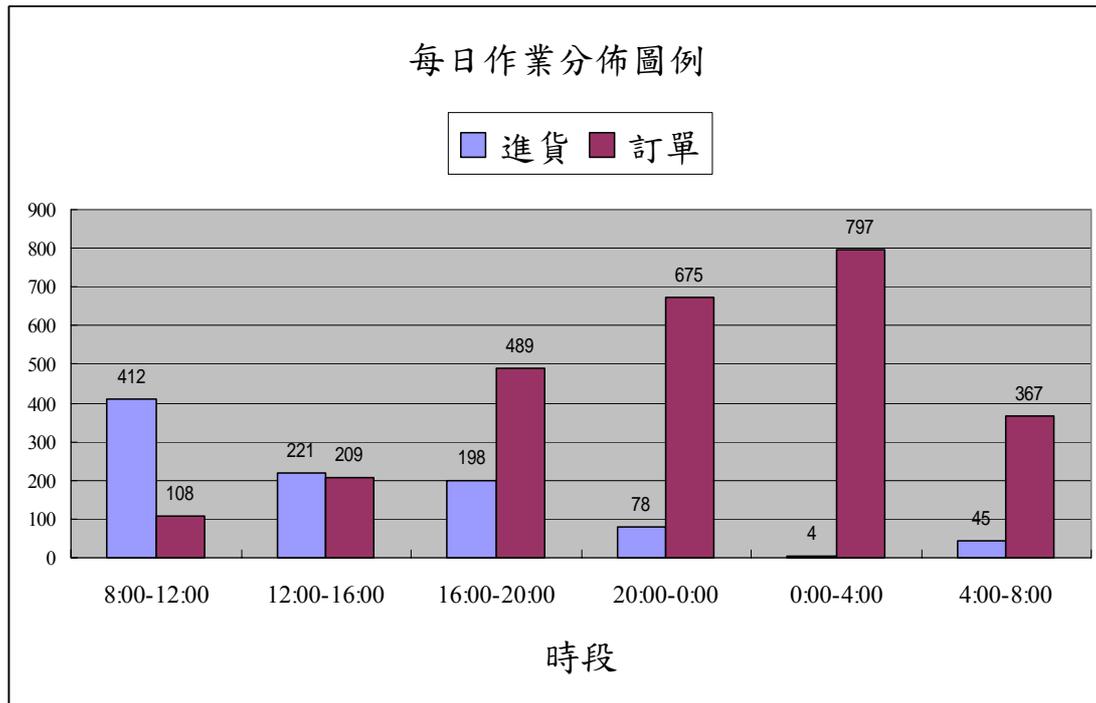


圖 2.2 每日作業分佈圖

一般物流中心多以機器設備及訂單為考量的因素，但是因為機器設備仍須由人員操作，因此本研究視人員為物流中心主要的產能瓶頸。在目前物流中心各項內部作業中，揀貨作業是一項重要且繁雜的工作，約佔物流中心作業時間的 40 ~50% 的時間，是否能在合理的時間內完成揀貨作業，直接關係著物流中心的經營成本與服務水準。排程的目的在於希望在交期時間內，將產品完成揀貨、分類與裝運，使其得以順利送達顧客的手中。實務上的人力分派問題有許多問題需要加以考量，若物流中心無良好的人員排程制度，容易造成額外不需要的加班引發企業增加額外的加班成本。本研究之目標為考量既有的人力資源下，應如何排定人員的作業排程，在滿足客戶需求的前提下，使平均總成本達到最小。

### 2.3 車輛排程問題

針對物流中心車輛排程問題，Michiko et al.(2001)提出因為實體配送系統多屬於組合最佳化的問題，因此難以應用一般的數學規劃軟體迅速有效地求解，作者提出可以利用遺傳演算法的排程去取替一般的排程方法，利用 GA

編碼規則的技術及適應力函數當作一評估的準則，將其應用在實體配送中心當中，使其管理更有效率。黃紹維(2006)運用遺傳演算法求解接駁式轉運中心車輛排程之研究，貨物在上游產出後，需利用各種的運輸設備，將貨物交付至下游顧客。如何整合進站和離站路線車輛排程，能達到降低物流成本使系統平均總成本達到最小。利用遺傳演算法具有的強大的平行運算能力，以及跳脫區域最佳解的良好特性，求解接駁式轉運的車輛排程問題。但是，其論文中未考量有限產能下的超時作業，此與現實狀況不符。Hyun et al.(2005)探討運用 GA 決定動態配送網路的結構最佳解，並利用模擬根據平均服務時間去評估倉庫的容量，進而根據這些資訊決定車輛的派遣、人力資源的配置與物料需求處理等。Baumol and Vinod (1970) 利用貨物運輸量來建立一理論性的庫存模式。藉由直接運送成本、運輸持有成本、訂購成本與庫存成本之加總找出最佳的發車間距，並依據存貨模式無異曲線找出簡化之最佳安全庫存量，以提供不同成本參數影響下安全庫存量的改變。Blumenfeld et al. (1985) 藉由分析貨物於路網配送時所發生的成本來決定各種不同運送情況下最佳的配送策略。Lee(2003)研究在動態需求的考慮下，用多項式運算時間演算法 polynomial time algorithm 來計算產品離開物流中心應該多久進貨一次，而且數量應該為多少車輛派送問題排程。前述文獻多探討關於車輛配送排程的成本，而本研究除了車輛的相關成本項之外，並考量物流中心本身的運作相關成本。

另外，黃惠民(2001)指出專業化物流公司(Lead Logistic Provider, LLP)是利用最佳化的物流網路對外部物料運輸以小批量多頻度之交貨方式，在指定時間運達，符合顧客需求。其具體的要求事項如下：

1. LLP 是管理協力廠商交貨運輸，以確保物料在指定時間內送達。
2. LLP對整體供應鏈提供最佳化的運輸與物流管理，它必須具有熟悉的作業系統，工具與方法及管理物流網路之專業，使達到可靠的、可信賴的作業流程。
3. LLP 必須使用即時性的資訊系統，且能與僱用服務公司的系統溝通，利用網際網路技術連結與傳送固定的，平準化的排程及預測資訊，使協力廠商能依生產排程交貨。
4. LLP 必須依排定的交貨時間表運送適質、適量之物料到指定交貨點，此將要求小批量,多頻度之交貨，以降低廠內之庫存量。

5. LLP 必須採用精實生產的技術，減少批量大小及符合顧客需求之同步化生產。

專業物流公司的興起使得在物流作業能夠更快速的進行貨物運輸，由於目前的資訊交換越來越快速，使得物流公司需要大量資金來投資各種資訊建設，而若是以一家非以物流為核心的企業而言，這些的投資都是一項極為沈重的負擔，加上目前的產業為顧客市場為導向，許多的顧客都希望能夠在最快的時間獲得商品，因此若是產品要快速的交付顧客又要符合運輸成本的經濟效益，幾乎是很難同時滿足，則可以為企業帶來此項服務的效益。

蔣明晃(2001)研究指出美國企業物流委外能夠盛行的因素約可分為：

1. 從 1990 年代開始解除運輸的多項限制，運輸體系逐漸成熟，使得企業得以透過穩定的運輸作業減少作業的不穩定性。
2. 專業物流公司成長快速：根據 Lieb(1999)的報告指出，美國專業物流公司 1999 年的平均成長率為 17%，而預計未來三年的每年成長率亦高達 16%。
3. 全球化趨勢：隨著企業全球化市場的興起，企業很難將所有物流作業由自己來運作，必須藉助於專業物流公司的協助，方能準確地滿足顧客的需求。
4. 資訊與決策支援工具的運用，使運輸倉儲體系發生極大改善，亦使成本降低了約 10~15%。

在過去充分依賴人工作業的物流中心，由於近年來的決策工具的運用，使得運輸作業能夠充分掌握。而就如蔣明晃(2001)所提出的，資訊與決策工作的運用，使得成本降低了約 10%~15%，就光以物流產業的成本結構而言，加上物流產業需要許多軟硬體的成本支出之外，這樣的降低營運成本的程度是個很可觀的比例，若是在未來企業能夠將物流作業交付給專業物流公司進行處理，那麼企業將不需在負擔額外的物流成本支出，而只需與第三方物流公司進行協商，使得雙方都能夠獲得各自最滿意的需求與目的。

目前大部分的企業都無法負擔完善的運輸車隊，由於運輸車輛的成本過高，可能造成企業獲利能力下降使得企業尋求配銷委外方式，利用專業化的物流公司進行貨物配送作業。本研究欲規劃一套完善的車輛排程，如何有效

進行貨物的配送規劃，運用整體配送成本最小化的方式進行貨物運輸，讓企業針對本身之營運成本進行有效控管，為物流業者一重要之課題。

## 2.4 文獻小結

本研究所關心的決策情境是在產品由工廠完成製造後，經由物流中心將產品作重新分類包裝後，將各個零售賣場所需之數量裝載上車，在這一連串的物流作業當中，零售體系經理人的目標是在交期時間內把產品送達顧客的手中且使平均總成本最小，並在人力及車輛等產能有限的狀況下使其排程為可行。

過去在一般物流中心配送排程問題的模式中，大多假設物流中心內部的產能為無限，物流中心的內部排程皆能即時滿足顧客的需求。然而在現實的狀況，由於人員一般有工作時間的限制。人員的加班會產生企業額外的人工成本，若不考量加班時間的因素探討，則無法與實務的狀況相互配合，且會誤導零售業者經理人的決策。但是在上述的文獻中卻未見有所著墨。因此，本研究將探討同時考量物流中心內部的人員排程與物流中心外部的車輛配送排程問題。

### 第三章 數學模式

本章節主要說明研究主題的範圍和考量之限制條件：在第 3.1 節首先描述本研究的決策情境，第 3.2 節說明限制條件與決策變數，在第 3.3 節推導零售配銷體系中各個成本項，第 3.4 節彙整零售配銷體系下之數學模式。

#### 3.1 問題描述

假設在一連鎖零售體系下，共有  $m$  家零售賣場，分別是  $R_1, R_2, R_3, \dots, R_m$ ，零售賣場設置了一物流中心 (Distribution Center, DC)，與連鎖零售賣場有合作關係之供應商有  $n$  家，分別是  $S_1, S_2, S_3, \dots, S_n$ ，如圖 3.1 所示。物流中心功用在於整合供應端產品：當上游供應商供給產品抵達中心時，物流中心作業人員根據下游零售賣場需求，運用適當的排程，將各零售賣場需求產品進行分裝整理，並裝載至貨車運抵零售賣場，在滿足顧客需求的條件下，使上述供應鏈作業的平均總成本達到最低。

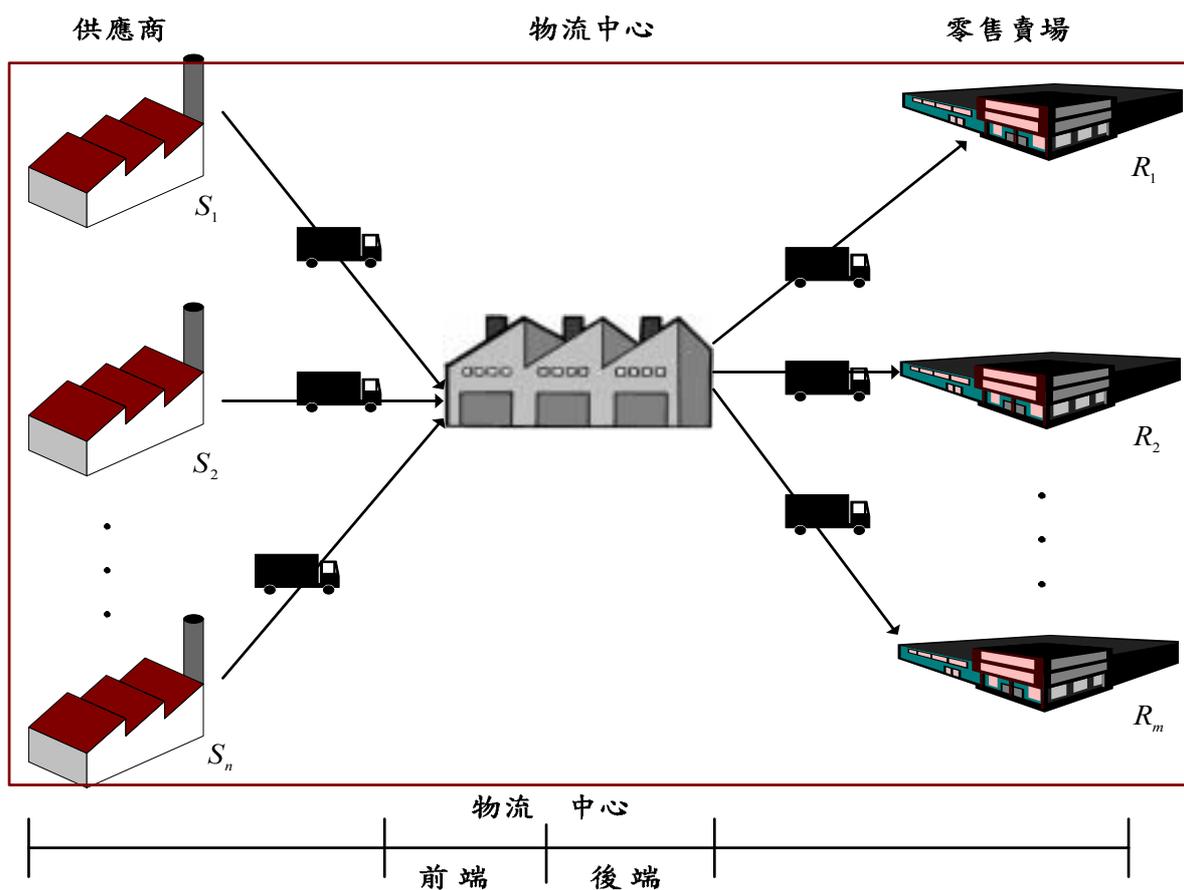


圖 3.1 零售配銷體系成員

物流中心作業分為前端及後端兩個部分。已知上游供應商  $i$  供給率為  $Q_i^a$  (公斤/天)，若進站路線  $i$  的進站間距為  $k_i^a B$ ，則進站路線  $i$  的產品載運

至物流中心的總量（即一次的載運批量）為 $k_i^a BQ_i^a$ 。「進貨」這項作業是所有倉儲作業的暖身，如果進貨作業的時間沒有進行適當的協調，會導致入庫、儲存、揀貨或出貨等作業調度上的困難。研究中將物流中心前端主要需花費時間之作業分為兩大類：卸貨作業及揀貨作業。因為從供應商端運送至物流中心的這些產品分別要送至各個不同的零售端手上，因此，需將產品以同類或同一個零售賣場的方式進行分類的動作。令物流中心作業人員將進站路線 $i$ 之產品由貨卡車卸下並進行揀貨分裝，每單位產品平均花費時間為 $X_i$ （公斤/分鐘），因此進站路線 $i$ 的產品在物流中心進貨端所耗用的時間共為 $X_i K_i^a BQ_i^a$ 。

同理，物流中心後端主要將作業分為揀貨及裝車等動作。下游零售賣場 $j$ 需求率為 $Q_j^d$ （公斤/天），若離站路線 $j$ 的離站間距為 $k_j^d B$ ，則離站路線 $j$ 的產品經整合後載運至零售賣場的總量（即一次的載運批量）為 $K_j^d BQ_j^d$ 。在物流中心的出貨端，離站路線 $j$ 依零售賣場 $j$ 的補貨需求進行揀貨並裝載上車，每單位產品平均花費時間為 $X_j$ 。因此當產品分類完運離物流中心時，離站路線 $j$ 的產品在物流中心出貨端所耗用的時間共為 $X_j K_j^d BQ_j^d$ 。根據 Hwang（1988）的研究得知，揀貨作業是直接相關人力的投入，約佔了整個物流中心投入人力的 50% 左右；而在整體物流中心的作業時間當中，花在揀貨的時間亦佔了大約 30%~40%。Frazelle (2002) 亦指出在物流倉儲中心當中，不同時段的尖離峰狀態是有所差異的，應該依據真實的狀況在不同的時段對物流中心內工作排班或人力進行調度。因此期望物流中心在考量超時作業人工成本，及能滿足零售端客戶需求的前提條件下，運用適當的排程，將各零售賣場的需求裝載至貨車運抵零售賣場，使上述整個供應鏈作業的平均總成本達到最低。

本研究係參考物流業者實際作業之模式，將進貨作業及出貨作業分開進行處理；以台灣聯合物流為例，該公司的物流中心早上只進行進貨的工作，而下午不進貨，只進行出貨的動作。將進貨作業及出貨作業時間分別處理主要考量的原因有二：對物流中心動線而言，如此的設計可以增加物流中心揀貨動線的流暢性，避免進出貨作業同時進行容易發生工安的意外。另外，在實務的考量上，因為物流中心的人員有限，讓同一批人在上

午只進行進貨的工作，下午可將同一批人移至出貨區進行產品裝載上車的動作，不僅可以提高效率，如此在物流中心的人力也能得到較為妥當調度的安排。

## 3.2 限制條件與決策參數

由於實務產業環境的複雜與多變，造成決策經理人不易掌控整體企業運作情況，無法對企業運作問題提出一個有效的解決方案。因此本研究希望能夠在合理的假設下，以契合實務界運作的模式，提供決策經理人一套有效的決策依據。

### 3.2.1 基本假設

本研究車輛排程問題的情境中，運用下列幾項假設：

1. 各供應點與需求點之供給率與需求率已知。
2. 進行產品運輸時，將運輸時間考量為貨車前置時間。
3. 產品為單向運輸，亦即由供應商運至物流中心時，回程時不載運產品。
4. 產品在運送過程中不會遭受損害或耗損。
5. 物流中心僅提供集散與配銷，自己並不需求產品。
6. 運送到物流中心的產品一定會在當天完成分裝作業排程。
7. 需求端不會產生缺貨。

### 3.2.2 條件限制

1. 離站間距的最小值應大於或等於( $\geq$ )進站間距的最大值，避免缺貨問題發生。
2. 作業時間超過物流中心人員的基本工時，則需計算超時人工成本。

### 3.2.3 決策變數

在此配銷系統當中，身為一位連鎖零售體系的決策經理人而言，除了各賣場的銷售情況之外，對公司營運成本掌控更是一個需要關注的重點。在整體供應鏈體系下，為能及時滿足顧客需求，並有效降低營運成本，完善的配銷規劃將成為企業一項重要的課題。

一套完善補貨排程規劃可以有效進行運輸成本管控，減少運輸時所產生之存貨成本負擔，零售配銷系統下的前後端貨車發車間距的協調將是本

研究所探討的重點之一。

本文假設決策的進站路線間距乘數 ( $k_i^a$ ) 與離站路線間距乘數 ( $k_j^d$ ) 符合二冪 (Power-of-Two, PoT) 策略，二冪策略是要求進站策略的週期為  $k_i^a B = 2^{P_i} B$  且離站週期為  $k_j^d B = 2^{P_j} B$ ，其中  $P_i$  及  $P_j$  皆為非負整數。本研究使用二冪策略的原因為，最大的補貨時間間隔可為連續，即為整個配銷運籌的循環時間  $T$ ， $T = LCM(\{k_i^a\}, \{k_j^d\})B = \text{Max}(\{k_i^a\}, \{k_j^d\})B$ 。假設在二冪策略下各路線的其發車間距分別為 2 天、4 天與 16 天，因此可以掌握此發車間距之循環週期為  $T = LCM(2,4,16) = \text{Max}(2,4,16) = 16$  天，循環週期代表著每 16 天，車輛排程會循環重複。反觀若是採用一般整數策略 (General-Integer Policy, GI)，假設各路線的發車間距為 5 天、11 天與 13 天皆為質數，故此發車的循環週期  $T = LCM(5,11,13) = 715$  天。因此，若是以一般整數策略方式，則需要 715 天才能夠形成一個循環週期，使決策者難以掌握發車排程計畫。若是採用二冪策略其循環週期只需 16 天，此較符合實務作業的需求。令間距乘數符合二冪策略，此為建構零售配銷體系數學模式之限制式。本研究中的決策變數如下：

1. 決定零售配銷系統之最佳基本週期，即為  $B^*$ 。
2. 決定各進站路線  $i$  及離站路線  $j$  之最佳間距乘數，即決定  $k_i^a$  與  $k_j^d$ ，並且  $k_i^a$  與  $k_j^d$  符合二冪 (PoT) 策略。

### 3.3 零售配銷體系中的各個成本項

在本小節探討零售配銷體系中之各個成本項的推導。因為產品由供應商端經由進站貨車運送至物流中心時，物流中心人員將產品依零售賣場需求進行分裝及整合，再由離站貨車將所需數量裝載上車運送至零售賣場。這一連串物流作業中，本研究將成本項劃分為三類，分別為 (1) 零售配銷體系之進貨端成本、(2) 在物流中心發生之營運成本與 (3) 在零售賣場端發生之成本，以此三類成本建構出整體配銷系統所有發生之成本。

#### 3.3.1 零售配銷體系之進貨端成本

零售配銷體系之進貨端平均成本，包含：貨車將製造完之產品由供應商端運送至物流中心時所產生之貨車操作成本，以及在運送過程所產生之產

品在途成本兩部分。

### 1. 進站車輛單位時間成本

操作成本與載運產品種類、數量與載運設備有關，依據 Jansson (1980) 所使用之單位車輛操作成本，可以線性方式表示，分成固定成本與變動成本兩部份：貨車固定成本為  $\alpha$  (元)，變動成本為  $\beta$  (元/公斤)，而貨車載運量為  $S$  (公斤)，故車輛操作成本  $G_T$  為

$$G_T = \alpha + \beta S \quad (3.1)$$

式(3.1)之成本表示方式是以產品量為考量因素的成本函數，而在丁慶榮與翁偉倫 (2003) 中所提出的貨車雖考慮往返車程，同時也考量了返程並無載運產品，僅考慮貨車之固定成本，故車輛操作成本為

$$C_T = C_{TG} + C_{TR} = \alpha + \beta S + \alpha = 2\alpha + \beta S \quad (3.2)$$

但是在丁慶榮與翁偉倫 (2003) 的研究中，車輛操作成本項目之變動成本與載運產品量 ( $Q$ ) 無關，而是與貨車載運量 ( $S$ ) 相關，此點並不合理。為了降低車輛操作成本的複雜性，研究中將進站與離站路線所使用車輛的固定成本與變動成本分別設定為  $\alpha$  與  $\beta$ ，車輛之成本函數會根據運送的產品量 ( $Q$ ) 成線性函數，為  $2\alpha + \beta Q$ 。舉例來說，一台貨車最多可以載重 2 噸，實際上載運量為 1.5 噸，若依丁慶榮與翁偉倫 (2003) 方式則以 2 噸計算成本，但因貨車實際載重量只有 1.5 噸，故應以 1.5 噸作為變動成本之計算。另外，由於單一路線下，並非只有單一車輛進行運輸工作，因此本研究為探討車輛排程，需先知道在單一路線下所使用的車輛數。使用車輛數的計算方式為：供應商運輸產品至物流中心之行駛時間 ( $V^a$ ) 除以發車間距

( $k_i^a B$ )，使用車輛數為  $\frac{V^a}{k_i^a B}$ 。在單一路線下的車輛操作成本為：車輛單位

操作成本 ( $C_T$ ) 與使用車輛數  $\frac{V^a}{k_i^a B}$  相乘，其車輛操作成本為

$$\frac{V^a}{k_i^a B} (2\alpha + \beta Q) \quad (3.3)$$

企業要維持一個完整的運輸車隊，每個月除了需要支付駕駛人員的薪資之外，還需要對貨車進行固定的維修與保養，這些問題都會讓企業的營運成本居高不下。改採第三方物流 (Third Party Logistic, 3PL) 公司的優點在

於，將配銷工作委託更專業的物流公司，讓運輸工作可以更有保障之外，企業本身不需對運輸車輛作額外的管理，如固定的維修或是保養。因此，本研究在車輛運輸成本的部分，改採與第三方物流公司簽訂契約方式租借貨車，使得企業所規劃的車輛排程能夠發揮更大的效益，並且藉由運輸工作委外讓企業的配銷成本有效降低，讓公司能夠更專注於本身的專業能力。在已知單位時間產品量的前提下，決策經理人能指派不同的貨車類型到指定地點運輸產品，第三方物流亦根據業者所需要的車輛類型進行發車。在研究中車輛搜尋方式係根據貨運量的多寡而選擇所需要的運輸車輛類型，目前的運輸車輛類型如表 3.1，運輸車輛共有 5 種類型，而每種類型車輛都有各自的載重限制。黃紹維(2006)貨車選擇方法步驟如下：

步驟 1：是否可以只用一輛超過貨物量但是最接近類型貨車就可承載

1.1 是，輸出此路線採用的運輸車輛類型與車輛數

1.2 否，選擇採用最大承載之貨車類型承載貨物，記錄目前所選擇之車輛與車輛數，記錄剩下貨物量，並且重複步驟 1。

表 3.1 貨車類型與費用

貨車類型 ( $\tau_x$ )	載重量 (公斤) ( $w_x$ )	貨車費用 ( $P_x$ )	單位重量費用 (公斤/元)
0.5TS	300	\$800	\$2.67
1.5TS	800	\$1,600	\$2.00
3.5TS	2000	\$3,000	\$1.50
8TS	5000	\$4,000	\$0.80
10.5TS	6000	\$4,500	\$0.75

(註：貨車類型與費用資料，參考華捷貨運公司網頁資料，2006)

假設目前供應商 1 的總貨運量為 7000 公斤，由於產品的總重量超過所有類型的車輛，因此選擇最大的載重貨車類型 10.5TS，此類型貨車的載重量為 6000 公斤，剩下的貨物量為  $7000-6000=1000$  公斤，記錄目前運輸車輛類型，回步驟 1 重新搜尋剩下的貨物量應選擇的車輛類型，此時會採用 3.5TS (載重量為 2000 公斤) 類型的運輸車輛，因此，供應商 1 所需要的運輸車輛分別為 1 台 10.5TS 與 1 台 3.5TS 類型的運輸車輛，運輸費用為  $\$4500 + \$3000 = \$7500$ ，但是這樣的搜尋方式並不合理，應根據實際載重量選取成本較低的車輛類型，本研究車輛選擇方式如圖 3.2 所示，詳細之選擇步驟參見附錄 A。以上述同樣的例子，當選取一台 10.5TS、一台 1.5TS 與一台 0.5TS 時，運輸費用為  $\$4500 + \$1600 + \$800 = \$6900$ ，較黃紹維(2006)的方法省了 600 元。使用這樣選擇方式的缺點為，因為每一次都要根據載

重量選擇不同的貨車類型，因此在運算上會較黃紹維(2006)的選擇方法花上較多的時間。

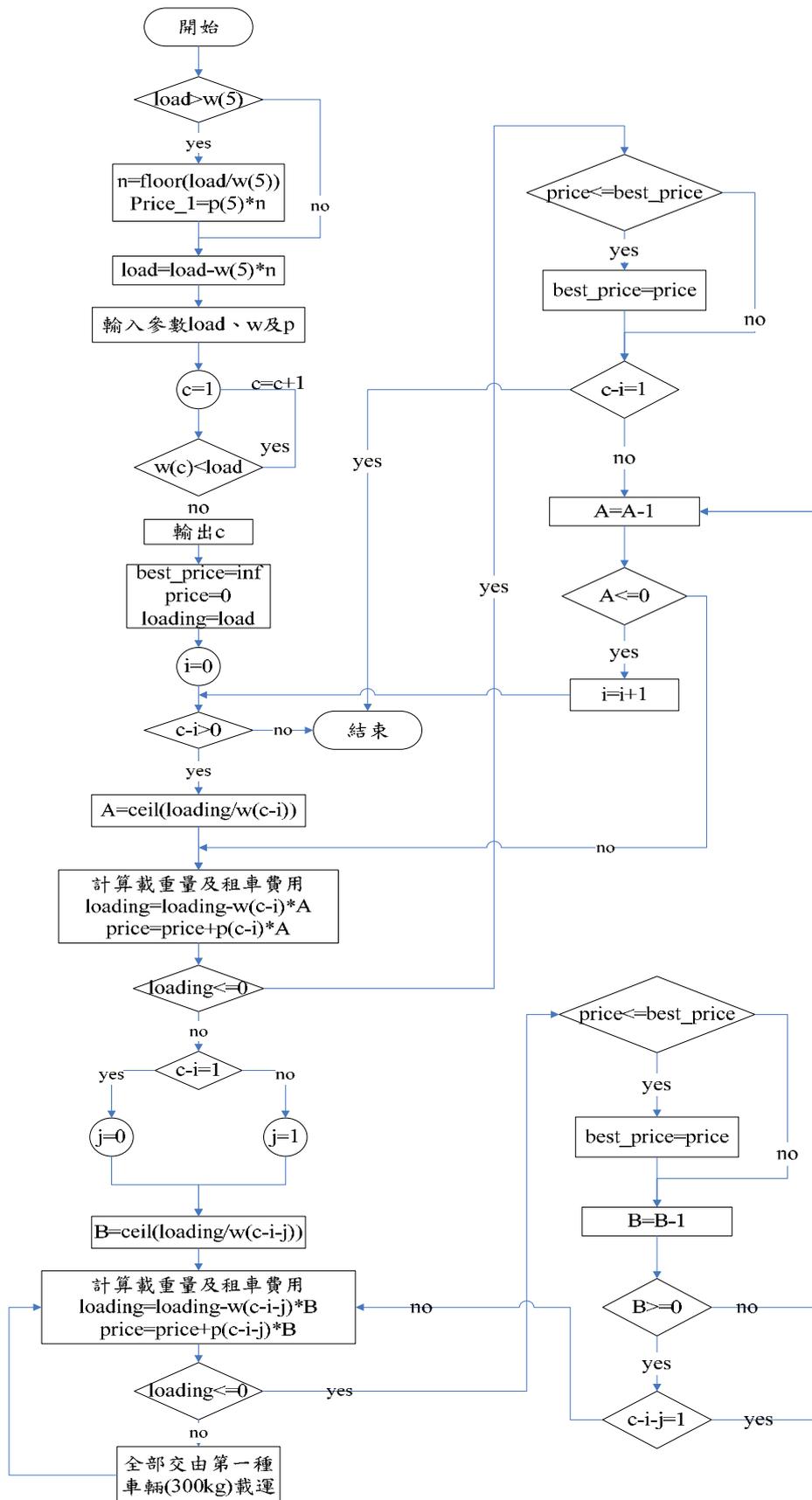


圖 3.2 車輛選擇流程圖

一旦決定貨車類型後，就能得知貨車租金費用，因此進站路線  $i$  車輛單位時間成本為  $\frac{P_x}{k_i^a B}$  (元/天)，其中  $P_x$  是向第三方物流業者租借貨車的費用，

$k_i^a B$  是進站間距天數。故進站路線車輛單位時間成本為

$$\sum_{i=1}^n \frac{P_x}{k_i^a B} \text{ (元/天)} \quad (3.4)$$

## 2. 進站路線單位時間產品在途成本

運輸貨車運送產品至物流中心進行整合時，由於產品在運輸途程上所耗費的時間，會產生產品於運輸途程上的在途成本。因此在單一路線下，根據貨車行駛時間 ( $V_i^a$ ) 與載運的產品量 ( $Q_i^a$ )，產品於進站路線時上所發生的在途成本為  $h^s V_i^a Q_i^a$ ，其中  $h^s$  為單位產品之在途成本 (元/公斤-天)。

供應商  $i$  之路線在途持有成本為  $h^s V_i^a Q_i^a$ ，因此對於供應路線的整體存貨持有成本為  $h^s V_1^a Q_1^a + h^s V_2^a Q_2^a + \dots + h^s V_n^a Q_n^a = h^s \sum_{i=1}^n V_i^a Q_i^a$ 。故單位時間進站路線之產品在途成本為

$$h^s \sum_{i=1}^n V_i^a Q_i^a \text{ (元/天)} \quad (3.5)$$

### 3.3.2 物流中心成本

物流中心的營運情況如同一個企業體，除了固定營運成本之外，尚有其他運作成本需要考量。研究中因為固定營運成本無關補貨排程之決策，因此關於物流中心的相關成本僅考量產品在物流中心內的處理成本，及因為等待離站車輛產品在物流中心內所引發的單位時間之存貨成本。

#### 1. 單位時間產品處理成本

在 Bartholdi and Gue (2000) 的研究當中，為了減少人員在搬運上所耗費的成本，並加快產品整合速度，運用搬運設備進行產品搬運。產品經由物流中心作業人員揀選分類完畢後，會利用搬運設備將產品從進站碼頭運送至離站碼頭等候運送，因此，本研究將物流中心的產品處理成本納入考量。首先，將每單位時間之處理產品成本分為固定成本 (A) 與變動成本 (X)。其中，固定成本 (A) 表示每進站碼頭設備之使用成本，而變動成本 (X) 為每單位產品的處理成本，如：作業人員處理產品分類成本，此

部分根據產品處理量的不同，而有所變動；變動成本會根據產品的數量而有所增減。因此單一進站路線產品處理成本為  $A + X(k_i^a B Q_i^a)$  (元)，而其進站路線  $i$  單位時間產品處理成本為  $\frac{A + X(k_i^a B Q_i^a)}{k_i^a B} = \frac{A}{k_i^a B} + X Q_i^a$  (元/天)。

因此產品在物流中心的產品處理成本為

$$\sum_{i=1}^n \left( \frac{A}{k_i^a B} + X Q_i^a \right) \quad (3.6)$$

## 2. 物流中心單位時間存貨成本

供應商供給產品運抵物流中心並進行產品整合，由於各供應商路線的間距皆小於零售賣場路線的發車間距，因此在離站路線車輛尚未抵達前，會產生短暫的停留時間，其等待的目的在於等待離站貨車到達。在等待的過程當中，物流中心會產生等待的存貨持有成本。

首先我們以一個供應商對一個零售賣場的運作模式作為一個範例，如表 3.2，假設進站路線間距與離站路線間距分別為 2 天與 4 天，單位存貨庫存成本皆為 0.6 元/公斤-天，供應商 1 需要滿足零售賣場的每天 10 公斤之需求。在範例的進站間距與離站間距為 2 天與 4 天，當進站貨車進入物流中心時，產品需要等到第 4 天時才會運送離開物流中心，產品平均存貨水準為  $40/4 = 10$  (公斤/天)，產品在物流中心庫存成本為  $1 \times 10 \times 0.6 = 6$  元/天。

表 3.2 物流中心之淨產品量與存貨水準

日期	供應商 1	零售賣場 1	當日淨進貨量	累積存貨
1	0	0	0	0
2	20	0	20	20
3	0	0	0	20
4	20	40	-20 (缺貨)	0

當進站路線貨車運抵產品至物流中心時，會在物流中心等待其他離站貨車的進站載運產品，加上各零售賣場的離站路線間距週期並不相同，因此會在物流中心產生產品庫存成本。如表 3.3 以供應商 1 與零售賣場 3 為例，供應商 1 的貨車進站間距為 2 天，零售賣場 3 的離站間距為 8 天，則其累積存貨水準為 8 天一次循環週期。當離站貨車到達之後馬上將這些的產品

運送至零售賣場，因此在週期在第 8 天時，物流中心的存貨就會變成零庫存。圖 3.3 之累計機率面積即為供應商 1 對零售賣場 3 的存貨水準，因此

得到平均存貨水準為  $\frac{(70 + 70 + 140 + 140 + 210 + 210)}{8} = 105 \text{ kg}$

表 3.3 供應商 1 對零售商 3 之淨進貨量與累積存貨水準

週期	供應商 1	淨進站 累積產品量	零售賣場 3	離站 產品量	當日淨產品量	物流中心 累積產品量
	間距		間距			
	2		8			
1	0	0	0	0	0	0
2	70	70	0	0	70	70
3	0	70	0	0	0	70
4	70	140	0	0	70	140
5	0	140	0	0	0	140
6	70	210	0	0	70	210
7	0	210	0	0	0	210
8	70	280	280	280	-210	0

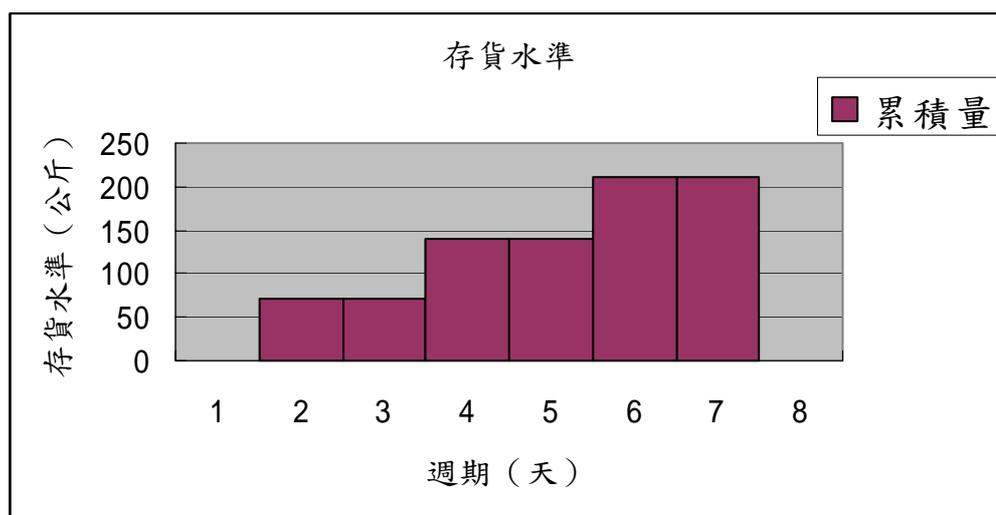


圖 3.3 供應商 1 對零售賣場 3 之存貨水準

例如：供應商 1 進站間距為 2 天，零售賣場 4 離站間距為 8 天，而零售賣場針對供應商 1 的單位需求產品量為 250 公斤/天， $B=1$  天，因此其平均存貨水準為 750 (公斤/天)，平均存貨成本為 450 元/天，詳細之平均存貨水準與平均存貨成本計算請見附錄 B。

**性質 1:** (a). 假設零售賣場  $j$  對供應商  $i$  的單位產品需求量为  $q_{ij}$ ，而供應商  $i$  的間距為  $k_i^a B$  及 (b). 在  $k_j^d \geq \max_i \{k_i^a\}$  的前提下，供應商  $i$  對於零售賣場  $j$  所

提供之產品的單位時間平均存貨水準為  $\frac{1}{2} \left( \frac{k_j^d}{k_i^a} - 1 \right) k_i^a Bq_{ij}$ 。

[證明]：供應商  $i$  對零售賣場  $j$  在物流中心的平均存貨水準為：

$$\begin{aligned}
 & \frac{k_i^a \left[ k_i^a Bq_{ij} + 2k_i^a Bq_{ij} + \dots + \left( \frac{k_j^d}{k_i^a} - 1 \right) k_i^a Bq_{ij} \right] + \left( \frac{k_j^d}{k_i^a} \right) k_i^a Bq_{ij} - k_j^d Bq_{ij}}{k_j^d} \\
 &= \frac{k_i^a Bq_{ij} \left[ k_i^a + 2k_i^a + \dots + \left( \frac{k_j^d}{k_i^a} - 1 \right) \right]}{k_j^d} \\
 &= \frac{k_i^{a^2} Bq_{ij}}{k_j^d} \left[ 1 + 2 + \dots + \left( \frac{k_j^d}{k_i^a} - 1 \right) \right] \\
 &= \frac{k_i^{a^2} Bq_{ij}}{k_j^d} \left( \frac{1 + \left( \frac{k_j^d}{k_i^a} - 1 \right)}{2} \cdot \left( \frac{k_j^d}{k_i^a} - 1 \right) \right) \\
 &= \frac{k_i^{a^2} Bq_{ij}}{k_i^a} \cdot \frac{k_j^d}{k_i^a} \cdot \frac{1}{2} \cdot \left( \frac{k_j^d}{k_i^a} - 1 \right) \\
 &= \frac{1}{2} \left( \frac{k_j^d}{k_i^a} - 1 \right) k_i^a Bq_{ij}
 \end{aligned} \tag{3.7}$$

因此，以第  $i$  家供應商與第  $j$  家零售賣場為例，根據性質 1，則此配銷體系下的存貨持有成本為  $h^r \frac{1}{2} \left( \frac{k_j^d}{k_i^a} - 1 \right) k_i^a Bq_{ij}$ 。物流中心因應  $n$  家供應商與  $m$  家零售賣場的存貨持有成本為

$$\sum_{i=1}^n \sum_{j=1}^m h^r \frac{1}{2} \left( \frac{k_j^d}{k_i^a} - 1 \right) k_i^a Bq_{ij} = \frac{1}{2} B h^r \sum_{i=1}^n k_i^a \sum_{j=1}^m \left( \frac{k_j^d}{k_i^a} - 1 \right) q_{ij} \tag{3.8}$$

### 3.2.3 零售配銷體系之出貨端成本

在零售配銷體系後端所發生的成本包含：(1)離站貨車之車輛操作成本、(2)離站路線在途存貨成本與(3)零售賣場庫存成本。

### 1. 離站貨車單位時間車輛操作成本

離站貨車之車輛成本觀念相同於進站貨車之車輛操作成本，離站路線  $j$  之貨車單位時間成本為

$$\sum_{j=1}^m \frac{P_x}{k_j^d B} \quad (\text{元/天}) \quad (3.9)$$

### 2. 離站路線單位時間在途存貨成本

從物流中心出發之運輸貨車運送產品至零售賣場時，而在途成本是根據產品載運量 ( $Q_j^d$ ) 與貨車從物流中心到零售賣場的行駛時間 ( $V_j^d$ ) 改變，產品時間之單位在途成本為  $h^s$ ，故在單一路線  $j$  的在途成本為  $h^s V_j^d Q_j^d$ 。所有離站路線上所發生的總在途存貨成本為

$$h^s \sum_{j=1}^m V_j^d Q_j^d \quad (3.10)$$

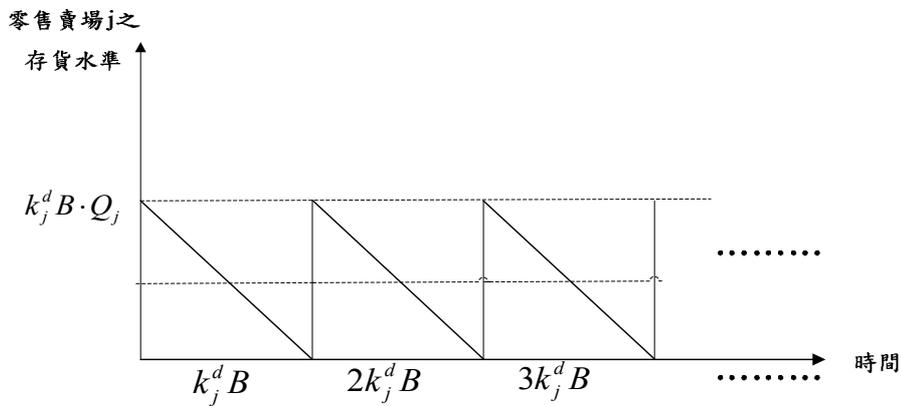


圖 3.4 零售賣場存貨水準

### 3. 零售賣場單位時間庫存成本

在單位時間  $k_j^d B$  (天) 下，零售賣場  $j$  產品量為  $Q_j^d$  (公斤/天)，單位產品時間總產品量為  $k_j^d B Q_j^d$ ，因此單位時間的平均產品水準為  $\frac{1}{2} k_j^d B Q_j^d$ ，如圖 3.4 所示。單位產品時間之庫存成本為  $h^r$  (元/公斤-天)，故零售賣場平均的庫存成本為  $\frac{h^r}{2} k_j^d B Q_j^d$ 。單位時間所有零售賣場之總平均庫存成本為

$$h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d \quad (3.11)$$

### 3.4 零售配銷體系下之數學模式

根據上述之成本模式建模步驟，此零售配銷體系下的成本模式可以劃分出幾個部份，分別是（1）供應商端平均成本（2）物流中心平均成本及（3）零售賣場端之平均成本；藉由此三項成本項目，可建構出本研究所欲探討之零售配銷系統之數學模式。

平均總成本=（供應商端平均成本）+（物流中心平均成本）+（零售賣場端平均成本）

(1). 供應商端成本=進站車輛平均成本+進站路線產品在途存貨平均成本；

參考(3.4)及(3.5)可得

$$= \sum_{i=1}^n \frac{P_x}{k_i^a B} + h^s \sum_{i=1}^n V_i^a Q_i^a \quad (3.12)$$

(2). 物流中心成本=產品處理成本+存貨庫存成本；參考(3.6)及(3.7)可得

$$= \sum_{i=1}^n \left( \frac{A}{k_i^a B} + X Q_i^a \right) + \frac{1}{2} h^r B \sum_{i=1}^n k_i^a \sum_{j=1}^m \left( \frac{k_j^d}{k_i^a} - 1 \right) q_{ij} \quad (3.13)$$

(3). 零售賣場端成本=離站車輛成本+離站路線產品在途存貨平均成本+零售賣場之庫存平均成本；參考(3.9),(3.10)及(3.11)可得

$$= \sum_{j=1}^m \frac{P_x}{k_j^d B} + h^s \sum_{j=1}^m V_j^d Q_j^d + h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d \quad (3.14)$$

因此本問題之數學模式列出如下：

$$\begin{aligned} & \text{Minimize } TC(k_1^a, \dots, k_n^a, k_1^d, \dots, k_m^d, B) \\ & = \sum_{i=1}^n \frac{P_x}{k_i^a B} + h^s \sum_{i=1}^n V_i^a Q_i^a + \\ & \sum_{i=1}^n \left( \frac{A}{k_i^a B} + X Q_i^a \right) + \frac{1}{2} h^r B \sum_{i=1}^n k_i^a \sum_{j=1}^m \left( \frac{k_j^d}{k_i^a} - 1 \right) q_{ij} + \\ & \sum_{j=1}^m \frac{P_x}{k_j^d B} + h^s \sum_{j=1}^m V_j^d Q_j^d + h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d \end{aligned} \quad (3.15)$$

Subject to

$$\sum_{i=1}^n q_{ij} = Q_j^d, j = 1, 2, \dots, m \quad (3.16)$$

$$\sum_{j=1}^m q_{ij} = Q_i^a, i = 1, 2, \dots, n \quad (3.17)$$

$$k_j^d \geq \max_i \{k_i^a\}, j = 1, 2, \dots, m \quad (3.18)$$

$$k_i^a = 2^{P_i}, P_i \in N, i = 1, 2, \dots, n \quad (3.19)$$

$$k_j^d = 2^{P_j}, P_j \in N, j = 1, 2, \dots, m \quad (3.20)$$

有關數學模式中的限制條件，茲說明如下：

1. 有關零售賣場  $j$  之產品需求，本研究架構在一連鎖賣場企業，下游零售賣場的需求需要由上游供應商來供給產品，而上游不同供應商所供給之產品種類不同，避免各零售賣場出現各種產品需求不足問題，所以要求零售賣場的產品需求為上游供應商之供給總和，如(3.16)所示。
2. 有關供應商  $i$  之產品供給，如(3.17)所示。由於各零售賣場所販售的產品皆相同，因此  $m$  家零售賣場所需的產品全部由  $n$  家供應商負責提供。
3. 離站間距  $j$  需大於等於最大進站間距，即(3.18)所示。由於各家供應商所提供的產品並不相同，為了避免在零售賣場的產品出現缺貨，或是造成物流中心的離站碼頭由於缺貨造成空車路線，所以本研究要求離站路線間距需要大於等於最大的進站間距(見 3.2.2 限制條件 2)。
4. 進站間距與離站間距皆須符合二冪策略，如(3.19)及(3.20)所示。為避免決策者掌握的時程過長，因此進站間距與離站間距限制為符合二冪策略，詳細說明請見 3.2.3 決策參數說明。

## 第四章 運用遺傳演算法求解

本研究運用遺傳演算法 (genetic algorithm, GA) 可多點平行搜尋避免侷限於區域最佳解且具有組合最佳化的特性，以求得物流中心車輛進站與離站間距乘數，以作為物流中心分裝作業排程輸入參數之依據。第 4.1 節說明遺傳演算法的架構，第 4.2 節應用遺傳演算法求解本問題之最佳解，第 4.3 節為小結。

### 4.1 遺傳演算法架構

遺傳演算法之基本理論由 John Holland 於 1975 年首先提出，其精神在於仿效生物界「物競天擇，適者生存」自然演化法則。演化過程中，運用疊代之方式選擇物種中具有較佳特性之上一母代，並隨機的相互交換彼此基因資訊，以期能產生較上一母代更優良之子代，如此不斷重覆，產生適應性強的最佳物種，基本步驟經由選擇 (selection)、交配 (crossover) 及突變 (mutation) 此三種機制完成演化過程；遺傳演算法流程如圖 4.2 所示。

本研究進站間距乘數  $k_i^a$  與離站間距乘數  $k_j^d$  是以二冪 (Power-of-Two, PoT) 策略為前提假設，因此在遺傳演算法當中，染色體是以各進站與離站路線的乘數冪次進行編碼。遺傳演算法計算時需經過數個運算過程，包含：產生初始母代、編碼、解碼、評估適應值、選擇、交配及突變等過程；遺傳演算法虛擬碼如圖 4.3 所示。

根據上述遺傳演算法流程，本研究將以此演算法找出最小成本下之發車間距，並同時求得各前後端路線應分別派遣何種類型車輛進行運輸作業，使得此配銷體系得以運作。

#### 4.1.1 產生初始母代與染色體之編碼(Encoding)

運用遺傳演算法時，依問題特性將可行解進行編碼。編碼為遺傳演算法中最重要也最困難之一部份，編碼後可行解稱為染色體 (Chromosome)，染色體由基因 (Genes) 所組成，舉例來說，圖 4.1 這一條染色體由 14 個基因所構成。

1	1	0	0	0	1	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

圖 4.1 染色體的組成

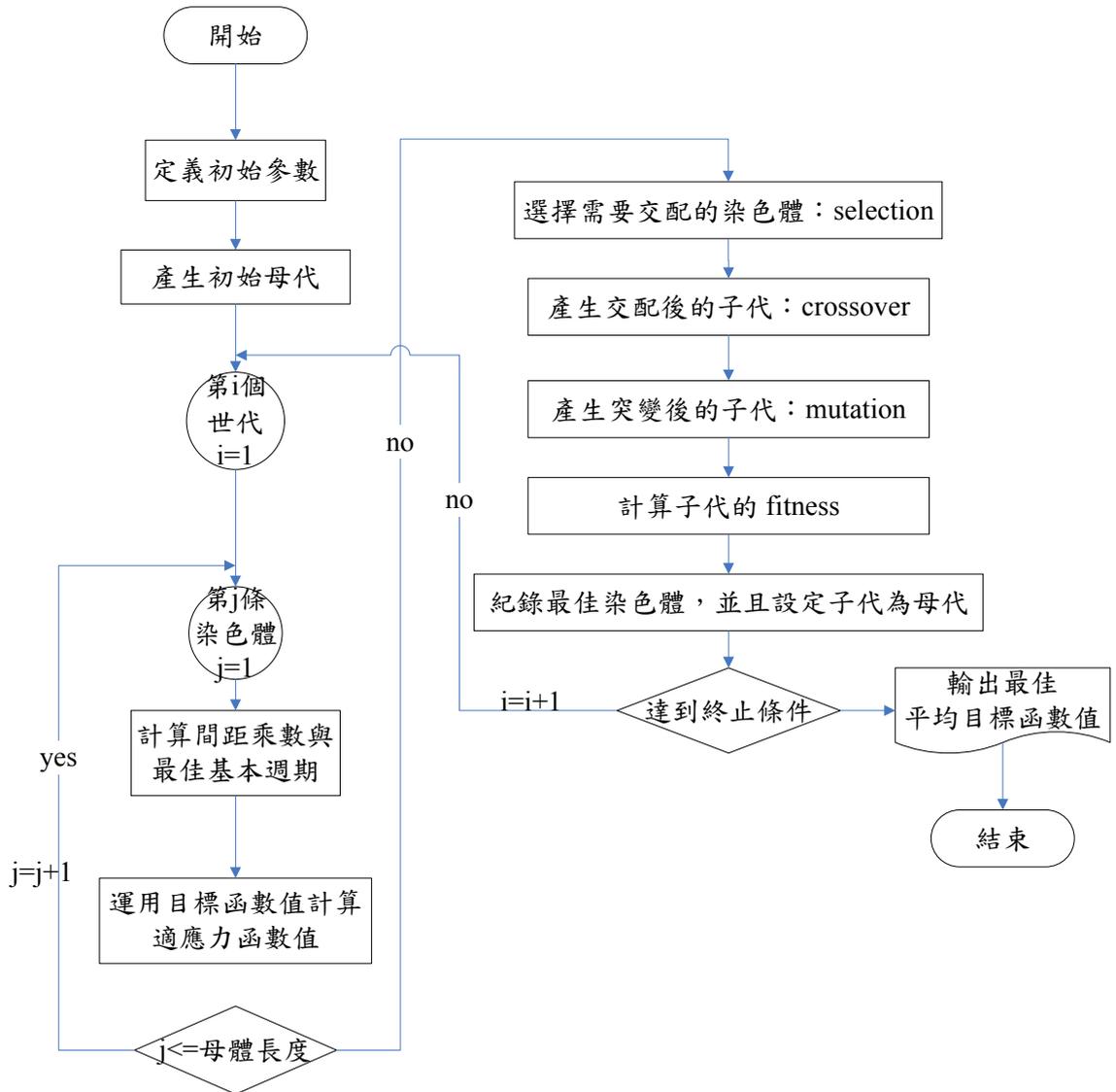


圖 4.2 遺傳演算法流程圖

遺傳演算法之編碼資料型態包含二進位編碼及整數編碼型態，而各種資料型態中又以二進位編碼最常用，主要的限制是問題解答要能夠以離散型態(discrete value)表達，故必須決定適當之染色體長度。

在 PoT 策略下，進站間距乘數  $k_i^a$  及與離站間距乘數  $k_j^d$  必須為 2 的乘冪次的正整數，即  $k_i^a = 2^{p_i^a}$  與  $k_j^d = 2^{p_j^d}$ 。因此在染色體基因組成的部分，是由每個間距乘數的冪次方所組成，即  $p_i^a$  或  $p_j^d$ 。其中每個產品所代表的基因長度由  $p_i^a(i)$  或  $p_j^d(j)$  之上界所決定。

開始： $t \leftarrow 0$

1. 產生初始母代 (initial population,  $P(t)$ )。
2. 進行適應值評估 (evaluate  $P(t)$ )。
  - 2.1 進行染色體解碼，獲得各路線間距 (decode)。
  - 2.2 運用成本曲線中各個斷點，尋找此組染色體(之路線間距乘數集合)的最佳基本週期  $B^*$
  - 2.3 記錄各組路線間距之最小成本與其斷點值，並將此最小成本設定為各染色體之適應值
3. 設定終止條件，並開始演化：

$t \leftarrow t+1$ 。

  - 3.1 從母代挑選適應值較佳之染色體進入子代 (selection)。
  - 3.2 針對子代之染色體進行交配 (crossover)。
  - 3.3 針對交配後之子代進行突變 (mutation)。
  - 3.4 複製子代為母代，評估適應值 (evaluate)
  - 3.5 子代最佳解與目前的最佳解進行比較，並進行最佳解之更新。
  - 3.6 是否到達終止條件
    - 是 → 進行步驟4。
    - 否 → 紀錄目前的最佳解，並繼續演化。
4. 演化結束，獲得最佳解組合與最小化成本。

結束

圖 4.3 遺傳演算法之虛擬碼

研究中之遺傳演算法運用隨機 (random) 方式產生初始的第一代，並根據第三章數學模式中，有  $m$  家零售賣場及  $n$  家供應商，因此共  $n+m$  個正整數型態的決策變數及向量，乘數  $K = (k_1^a, k_2^a, \dots, k_n^a, k_1^d, k_2^d, \dots, k_m^d)$ ，運用  $K$  向量中每個元素的冪次進行二元字串(binary string)編碼。舉例來說：若進站間距 1 間距乘數  $k_1^a$  的乘冪次方  $p_1^a$  上界為 5，因為  $2^2 < 5 < 2^3$ ，在染色體的表示法中使用 3 個位元長度來表示  $p_1^a$ 。當使用 3 個位元長度表示間距乘

數的乘冪次方  $p_i^a$  時，冪次範圍 0 至 7 之間的正整數皆包含其中，故進站間距的乘冪次方以 3 個位元長度表示。

離站間距  $k_j^d$  的乘冪次方  $p_j^d$  減少  $k_i^a$  一個位元數長度，即以 2 個位元長度表示。舉例來說：若離站間距 2 的離站間距乘數  $k_2^d$  之冪次方  $p_2^d$  為 3，因為  $2^1 < 3 < 2^2$ ，在染色體表示法中，以 2 個位元長度表示  $p_2^d$ 。當使用 2 個位元長度表示離站間距乘數之冪次方  $p_j^d$  時，冪次範圍 0 至 3 之間的正整數皆包含其中，且在第三章限制條件(3)中提出進站間距最大值應小於等於離站間距的最小值，故將離站間距加上此最大值 7，故  $p_2^d = 7 + 3 = 10$ 。其中，離站間距之乘冪次方以兩個位元長度表示的原因在於，以二冪策略解碼後， $2^{10} = 1024$ ，而當離站間距達到 1024 時，排程週期需要時間已經過長，故離站間距以 2 個位元長度表示之。

#### 4.1.2 解碼(Decoding)

產生初始母體之後，本研究為了避免零售賣場缺貨，因此提出了進貨端進站貨車進站間距最大值必須小於或等於後端離站貨車的最小值（請見限制條件(3)之說明）。假設有三個乘數  $k_1^a$ ， $k_2^a$  與  $k_3^a$  為進貨端路線之發車間距，進行隨機編碼之後，尋找此三個進貨端進站間距乘數的最大值，將此最大值設定為後端離站路線乘數可行之最小值；例如：當進貨端進站路線的最大進站間距冪次為 3，後端的離站路線間距乘數的冪次則以 3 的數開始編碼，如圖 4.4。當獲得進站間距乘數的冪次最大值為 3 之後，將此最大間距乘數冪次 3 作為後端離站間距冪次可行之最小值，後端路線解碼之後的乘數冪次都加上此可行之最小值，如： $P_1^d = 2 + 3 = 5$ 。由於本研究是以二冪 (PoT) 策略為前提假設，因此本研究的染色體解碼後，即為發車間距乘數之冪次，例如離站路線 1 間距乘數的乘冪次方  $p_1^d = 5$ ，則其解碼後之離站間距乘數為  $2^5 = 32$ ，其餘染色體則是以相同方式進行解碼。

$k_1^a = 2^{p_1^a}$			$k_2^a = 2^{p_2^a}$			$k_3^a = 2^{p_3^a}$			$k_1^d = 2^{p_1^d}$			$k_2^d = 2^{p_2^d}$		
$P_1^a$			$P_2^a$			$P_3^a$			$P_1^d$			$P_2^d$		
0	1	0	1	0	0	1	1	0	0	1	0	1	0	1
2			1			3			2			5		
↓														
2			1			3			5			8		
$k_1^a$			$k_2^a$			$k_3^a$			$k_1^d$			$k_2^d$		
4			2			8			32			256		

圖 4.4 染色體解碼

### 4.1.3 尋找基本週期(Basic period)

二元字串經編碼成正整數型態後，可以獲得各路線乘數之冪次  $(p_1^a, p_2^a, \dots, p_n^a, p_1^d, p_2^d, \dots, p_m^d)$ ，本研究將針對這些隨機產生的染色體進行適應值評估。在遺傳演算法當中，適應值評估是 GA 的性質指標，適應性函數值的大小可以反映出個體的優異程度。但是在獲得各染色體適應值之前，我們需要針對此組間距，找出在最小成本下的最佳基本週期( $B^*$ )。在已知一組車輛補貨間距的情況下，根據不同運輸車輛類型的載重限制，可以尋找此組間距的斷點(break point)，即為最佳基本週期(Basic period; BP)。計算斷點方法為

$$B = \frac{\text{貨車載重量}}{\text{間距乘數} \cdot \text{單位時間貨物量}} \cdot \text{貨車數} \quad (4.1)$$

由於貨車有多種類型提供選擇，斷點亦會隨著路線與貨車數量增加而增加，使得搜尋時間變長，故利用共同週期法先尋找出此配銷系統下的基本週期上下界，當斷點(break point)超過基本週期的上下界則不需再進行評估。

共同週期法 (Common Cycle, CC) 是假設所有的進站路線與離站路線都是運用相同週期，即  $k_i^a = k_j^d$ 。為了避免尋找過多的斷點(break point)耗費演算時間，因此在尋找其他已知路線間距的斷點(break point)之前，需要先計算共同週期的最小成本，使得共同週期能夠先找出基本週期的上下界，然後在此上下界範圍內找出已知間距的最佳基本週期，當已知基本週期之上下界時，可以將搜尋斷點(break point)的時間縮短並減少整體演算時間。

性質 1：基本週期值之下界  $B_{lb} = \max(B_1, B_f)$ ， $B_1$  為第一個斷點， $B_f$  是第一個臨界點

我們可以尋找各個基本週期的候選者，而當找出各個候選者之後，將這些的候選者與各路線的間距帶入評估函數式後，可以找到一個最小成本與相對應之最佳基本週期，運用目標函數值計算適應力函數值，目標函數值  $TC(K, B)$  表示如下式：

$$TC(K, B) = \sum_{i=1}^n \left( \frac{P_x}{k_i^a B} + \frac{A}{k_i^a B} - \frac{h^r B k_i^a}{2} \sum_{j=1}^m q_{ij} \right) + \sum_{j=1}^m \left( \frac{P_x}{k_j^d B} + \frac{k_j^d B h^r Q_j^d}{2} + \frac{1}{2} h^r B k_j^d \sum_{i=1}^n q_{ij} \right) + C$$

$$\frac{\partial TC}{\partial B} = \sum_{i=1}^n \left[ -\frac{P_x + A}{k_i^a B^2} - \frac{h^r k_i^a}{2} \sum_{j=1}^m q_{ij} \right] + \sum_{j=1}^m \left[ -\frac{P_x}{k_j^d B^2} + \frac{k_j^d h^r Q_j^d}{2} + \frac{1}{2} h^r k_j^d \sum_{i=1}^n q_{ij} \right]$$

$$\frac{\partial^2 TC}{\partial B^2} = \sum_{i=1}^n \frac{2(P_x + A)}{k_i^a B^3} + \sum_{j=1}^m \frac{P_x}{k_j^d B^3} \geq 0$$

$$\therefore \frac{\partial^2 TC}{\partial B^2} \geq 0, \frac{\partial TC}{\partial B} \leq 0, \text{ 因此 } TC(K, B) \text{ 為一凸性函數}$$

$$B_f = \sqrt{\frac{\sum_{i=1}^n \frac{P_x + A}{k_i^a} + \sum_{j=1}^m \frac{P_x}{k_j^d}}{\left( -\sum_{i=1}^n \frac{h^r k_i^a}{2} \sum_{j=1}^m q_{ij} + \sum_{j=1}^m \frac{k_j^d h^r Q_j^d}{2} + \frac{1}{2} h^r k_j^d \sum_{i=1}^n q_{ij} \right)}} \quad (4.2)$$

由於  $TC(K, B)$  為一凸性函數，因此  $TC(K, B_f) < TC_{cc}$  和  $TC(K, B_1) < TC_{cc}$ ，而  $B_f$  與  $B_1$  皆有可能產生  $TC(K, B)$  的最小成本，令  $B_{Lower}$  為經由共同週期法所獲得之基本週期之下界， $TC(\{k_i^a = k_j^d\}, B_{Lower}) = TC_{cc}$ ， $TC(K, B_f) < TC_{cc}$  和  $TC(K, B_1) < TC_{cc}$ ，即表示當基本週期小於  $B_f$  或  $B_1$  時，其解為不可行解，因此基本週期下界  $B_{lb} = \max(B_1, B_f)$ ，如圖 4.5 所示。

Heuristic Rule 1：已知路線間距的基本週期上界 ( $B_{up}$ ) 函數值為共同週期法

$$\text{最小成本 } TC(\{k_i^a = k_j^d\}, B_{Lup}) = TC_{cc}$$

令  $B_w$  為大於  $B_{up}$  之基本週期 ( $B_w > B_{up}$ )，由於  $TC(K, B)$  為一凸性函數，因此  $TC(K, B_{up}) \leq TC(K, B_w) \leq TC(K, B_{w+1}) \leq \dots \leq TC(K, B_{w+n})$ ，皆會大於共同週期下的最小成本  $TC_{cc} \leq TC(K, B_w) \leq \dots \leq TC(K, B_{w+n})$ ，且由式(4.2)，當間距

乘數  $k_i^a$  及  $k_j^d$  越大時，基本週期值  $B$  會越小，但因其中  $\left( -\sum_{i=1}^n \frac{h^r k_i^a}{2} \sum_{j=1}^m q_{ij} \right)$  項

並不確定會遵循如此規則，因此本研究訂定此一啟發式法則(heuristic rule)，以  $B_{up}$  為搜尋演算法基本週期的上界，如圖 4.5 所示。

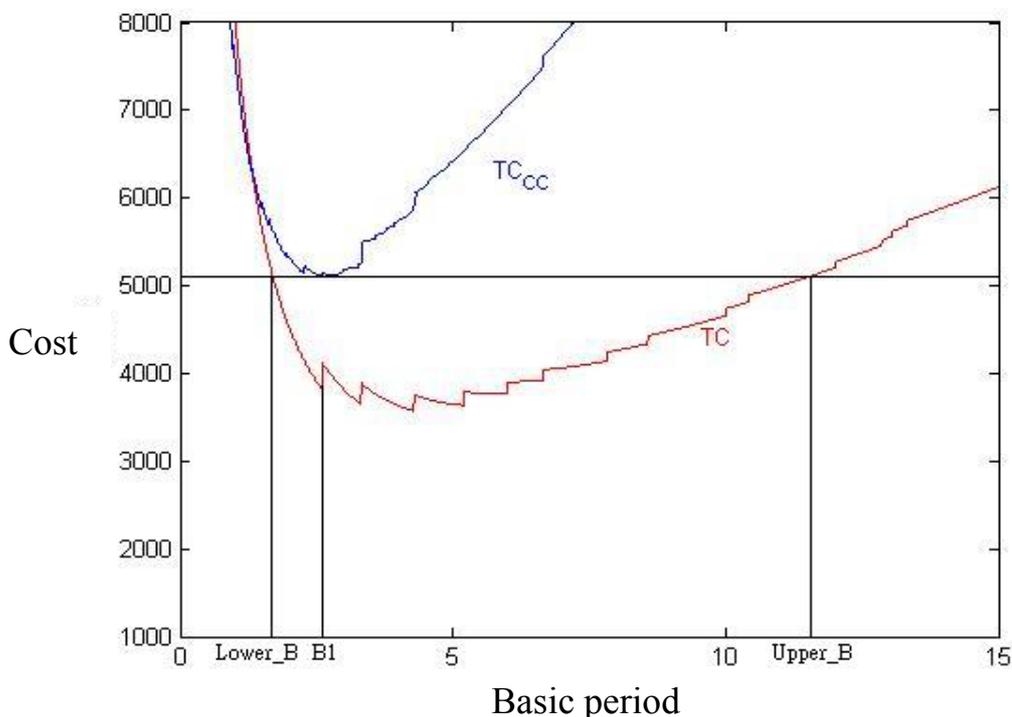


圖 4.5 利用共同間距法尋找基本週期之上下界

當獲得共同週期的總成本之後，則根據已知的路線間距尋找斷點(break point)，但是不需找出全部的斷點(break point)，只需尋找在上下界範圍內的斷點(break point)。如表 4.1，以第一條染色體為例，經過染色體解碼之後，可以獲得一隨機產生之間距。而在已知共同間距之總成本的情況下，則可將基本週期的上下界限制在  $[0.0314, 9.0909]$  範圍內，而超過此範圍之斷點(break point) 則是不列入搜尋考量。而其餘的染色體則是以相同的方式，在已知路線間距下，尋找最佳的基本週期與其適應值。經由共同週期下的總成本獲得成本上界後，亦得知基本週期之下界  $B$ ，故基本週期的範圍為  $[B_{lb}, B_{up}]$ 。

在此範圍內由於多條路線供給需求不相同，並且有多種貨車種類，使得在基本週期範圍內會產生許多斷點，因此將在此範圍內搜尋已知間距下的最佳基本週期，若是有產生斷點超過基本週期的上下界，則不考量這些超出範圍的基本週期。將在基本週期上下界內所有的斷點候選者帶入評估函數式，找到一個能夠使得成本最小化的  $B^*$ ，將  $B^*$  與獲得的最小成本進行紀錄，表示在此間距下，採取此  $B^*$  可以獲得最小成本。

表 4.1 第一條染色體與各基本週期之總成本

基本週期		總成本
$B_1$	0.0314	\$6176.31
$B$	0.0335	\$5833.10
⋮	⋮	⋮
$B^*$	0.0893	\$4246.04
⋮	⋮	⋮
$B$	6.8182	\$109785.5
$B$	7.2581	\$116758.3
Upper_B	9.0909	\$145856.8

當染色體經過解碼後，獲得路線間距，根據貨車類型的載重限制，可以獲得斷點(break point)，如  $B_1, B_2, \dots, B^*$ ，在這些所搜尋到的斷點(break point)一一代入成本函數，尋找最小成本的斷點(break point)，而在第一條染色體的最佳基本週期  $B^* = 0.0893$ ，在此最佳基本週期下的最小成本為 \$4246.036，而此成本將成為此條染色體的適應值，若是適應值越低，則越容易此條染色體被挑選到的機會更大。

#### 4.1.4 評估適應力函數值(Fitness Evaluation)

在已知各路線間距的情況下，將基本週期上下界內所有的斷點候選者代入評估函數式。適應函數是用來評估族群中每一個染色體適應力的指標，本研究運用目標函數值計算適應力函數值，目標函數值為最小成本，目標函數值越低代適應力越大，競爭力愈強，越有可能將本身的基因遺傳到下一代身上。運用 Marzouk 和 Moselhi (2003) 提出適應值的正規化 (fitness normalization)，可以更有效地解決最小化的問題。本研究採「線性排序正規化」(linear ranking normalization) 的方式，計算染色體的適應值。公式如式(4.3)所示。

$$\text{eval}_{i_{\text{temp}}} = 2 - SP + \frac{2(SP - 1)(i_{\text{temp}} - 1)}{PS - 1} \quad (4.3)$$

在線性排序正規化程序，將母體中所有的染色體根據其適應值函數排列至一表當中，其中 PS(Population Size)表示母體的數目，而表中染色體的 index 表示為  $i_{\text{temp}}$ ， $i_{\text{temp}}$  類似一排行榜，總成本越高的排在越前面，總成本

最小的染色體排行最後面，平均總成本最大的染色體（即最差的染色體）為  $i_{temp} = 1$ ，平均總成本最小的染色體為  $i_{temp} = PS$ 。例如：母體數為 5，利用 GA 求得最佳的五條染色體，目標函數值如表 4.2，適應值套用公式(4.3)可求得如表 4.1 第四欄所示。利用線性函數正規劃， $i_{temp}$  值越大，在選擇機制上所佔的比例也越大，該條染色體被選中的機率相對提高。此外，Pohlheim (1996) 定義「選擇壓力」(selection pressure; SP) 為選擇最好染色體的機率和平均機率之相對比率，SP 為介於  $[1.0, 2.0]$  的數值。

表 4.2 適應值函數

染色體	目標函數值	$i_{temp}$	適應值 $eval_{i_{temp}}$	累積機率
5	23723	1	0	0
2	21393	2	0.5	0.1
3	14114	3	1	0.3
1	13092	4	1.5	0.6
4	10684	5	2	1

#### 4.1.5 選擇

選擇(Selection)是依據每一個染色體(chromosome)之適應度(fitness)高低，決定其繁衍子代多寡的一種運算過程。適應度高的染色體在下一子代將被大量複製；適應度低的染色體在下一代中則被淘汰。一般常用的方法有輪盤式選擇法(roulette wheel selection method)及競爭式選擇(tournament selection method)。

- (1) 輪盤式選擇法：透過機率原理，將染色體適應值轉換為輪盤面積；在演化的過程中，會依各染色體適應值的大小來分割輪盤上的位置，適應值越大，在輪盤上的面積也就越大，相對被挑選出來的機率也就越大，即被挑選出來的機率等於輪盤上所佔面積的比例。
- (2) 競爭式選擇法：演化的過程中，隨機選取二個或更多個染色體，具有最大評估函數值，即被選中至下一個運算階段。

本研究採用輪盤式選擇法，以圖 4.6 為例，當隨機產生一組  $U(0,1)$  的變數值  $r = 0.9505$  (0.9,1)，選擇第 12 條染色體進行交配。

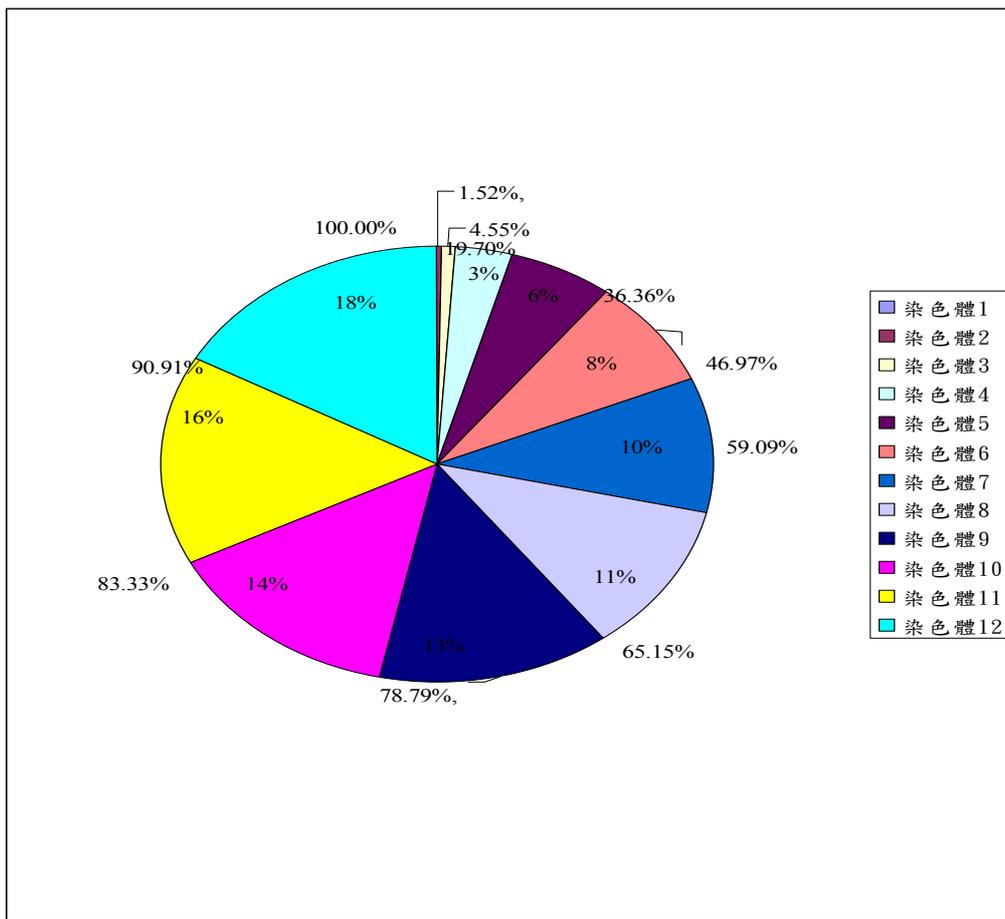


圖 4.6 輪盤選擇機制

#### 4.1.6 交配

交配(Crossover)是將經過選擇過程篩選出之成對染色體，以一定機率（交配率）決定是否進行配對。以隨機方式產生的 0-1 均勻分配（uniform distribution）亂數值若低於交配率，則進行交配，以新的子代取代母代；反之，被挑選之母代不進行交配，直接置入新的母體中。舉例來說，若有 9 條染色體，經過選擇機制後，選出第一條至第四條染色體進行兩兩交配，第五條至第九條染色體不進行交配，交配後得到新的染色體四條，新的母體如圖 4.7 所示。

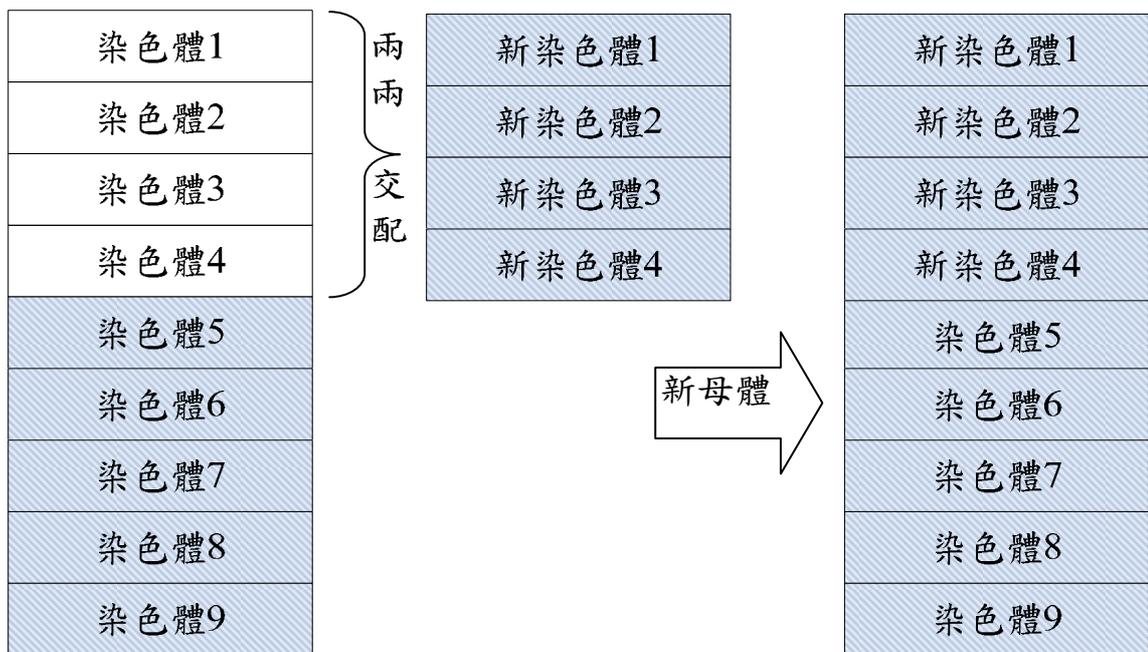


圖 4.7 選擇交配之染色體示意圖

藉由交配機制，子代可同時具有母代雙方之優良基因，組成更具適應能力之染色體。其處理方式主要有兩步驟：

- (1). 進行篩選後母體之隨機配對。
- (2). 隨機產生成對染色體之配對點，再依配對點之位置，進行基因互換。

交配的方法亦有數種，較常採用的方法有(a)單點交配(one-point crossover)、(b)雙點交配(two-point crossover)以及(c)均勻交配(uniform crossover)三種方式，交配方式介紹如下：

(a)單點交配

單點交配 (one-point crossover) 是以隨機方式，在所有可能切點 (crossover point) 中隨機選擇一切點，將切點左邊的基因予以保留，而切點右邊的基因予以互換，如圖4.8所示。

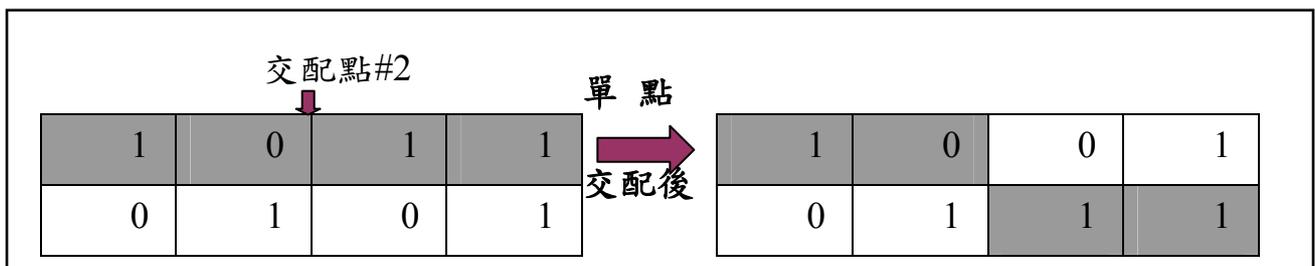


圖 4.8 單點交配示意圖

### (b) 雙點交配

雙點交配 (two-point crossover) 與單點交配極為類似，成對染色體中隨機挑選兩個交配點，將兩個交配點之間基因進行互換，如圖 4.9 所示。

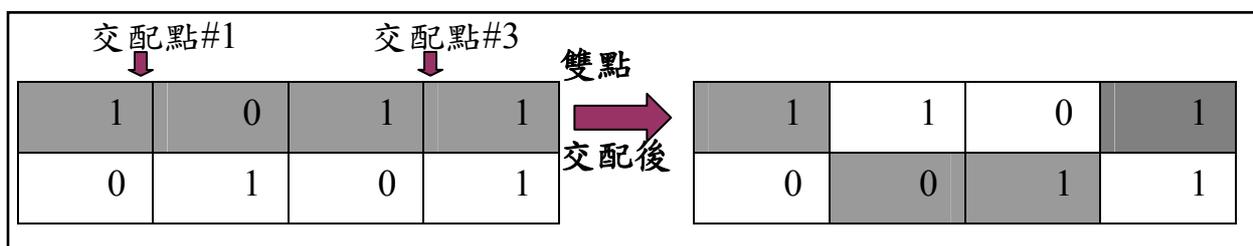


圖 4.9 雙點交配示意圖

### (c) 均勻交配

均勻交配 (uniform crossover) 是由 Syswerdy(1989) 提出的交配方法，針對基因字串中的每一個基因，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 0.5。隨機變數值  $< 0.5$ ，設其為 0，0 表示此基因不用互換；隨機變數值  $> 0.5$ ，設其為 1，1 表示要互換，如圖 4.10 所示。

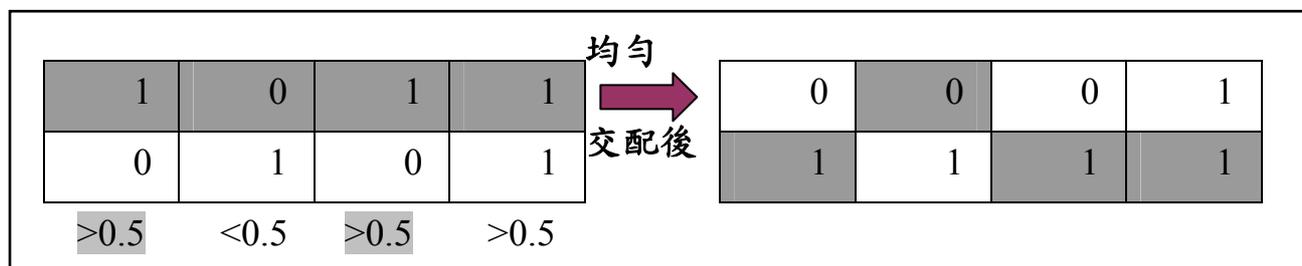


圖 4.10 均勻交配示意圖

而 Potts et al.(1994) 等一些相關研究報告顯示，使用均勻交配法，將有助於增加搜尋到最佳值的機率，故本論文採用均勻交配法。

### 4.1.7 突變

突變 (Mutation) 是針對子代之遺傳基因以一定機率 (突變率)，更動某一基因值 (0 變 1 或 1 變 0)。突變的目的在於防止複製和交配過程中，陷入區域最佳解。以隨機方式產生均勻分配  $U(0,1)$  亂數值，若亂數值低於突變率，該子代進行突變，再計算子代的適合度值；反之，子代不進行突變，可保留其交配結果的基因組合。突變的方式，首先，在染色體中，以隨機

方式選取欲產生突變的基因位置，決定基因位置後，將選取位置之基因值進行對調，如圖 4.11 所示。

1	0	0	1	0	0	1	1
							
1	1	0	1	0	0	0	1

圖 4.11 遺傳演算法之突變運作

#### 4.1.8 參數設定

遺傳演算法主要設定參數包括：母體大小(Population size)、交配率(Crossover rate；CR)、突變率(Mutation rate；MR)及繁衍的世代(Generation)等四項。這些參數的設定會影響求解時間及品質，根據問題的特性而定；依實驗結果調整，研究中之遺傳演算法參數設定如下：

1. 母體大小：即參與遺傳演算之染色體數目，本研究設為  $n+m$  條， $n$  為供應商數目， $m$  為零售賣場數目。
2. 交配率：本研究設為 0.7，低於此機率之染色體執行基因交換。
3. 突變率：本研究設為 0.25，低於此機率之基因需執行基因突變。
4. 終止條件 ( $\Gamma$ )：設定演算 100 代後，最佳解都沒有改變時，視為找到最佳解，停止搜尋。

## 4.2 遺傳演算法求解過程

本小節以第 4.1 節遺傳演算法架構為基礎，用一範例說明整個演算過程。範例中有四家供應商與八家零售商，已知各供應商與零售賣場間供給需求如表 4.3，相關運輸貨車載重限制與貨車費用如表 4.4，其餘成本需求項如表 4.5，遺傳演算法之參數設定如表 4.6 所示。

表 4.3 零售體系下各供應商與零售賣場之供給需求

零售賣場 供應商	1	2	3	4	5	6	7	8
1	15	20	35	40	35	40	20	20
2	20	25	30	50	25	50	40	30
3	10	20	40	70	45	70	45	35
4	20	40	35	30	25	25	30	50
備註	單位時間貨物重量 (Kg/天)							

表 4.4 貨車載重量與貨車費用

載重量 (公斤) ( $w_x$ )	貨車費用 ( $P_x$ )
300	\$800
800	\$1,600
2000	\$3,000
5000	\$4,000
6000	\$4,500

表 4.5 零售配銷系統之參數設定值

意義	參數	單位
單位時間-貨物庫存成本 ( $h'$ )	1.2	元/公斤-天
處理貨物固定成本 ( $A$ )	20	元
處理貨物變動成本 ( $X$ )	0.4	元/公斤

表 4.6 遺傳演算法之相關參數設定

母體大小(population size)	12
交配率(crossover rate)	0.7
突變率(Mutation rate)	0.25
繁衍世代(Number of generation)	20

#### 4.2.1 演算範例之產生初始母代與編碼

在範例當中，有 4 家供應商及 8 家零售賣場，因此在單一染色體當中會有 12 個變數。初始母體中矩陣由 0 和 1 位元組成，再根據染色體長度及族群數目隨機產生初始母代。在範例中，令進站間距乘數  $k_i^a$  長度為 3 個二元位元，而後端離站  $k_j^d$  減少  $k_i^a$  一個位元，即離站間距乘數 2 個位元，則染色體長度表示為  $4 \cdot 3 + 8 \cdot (3-1) = 28$  個位元，母體大小(Population Size)為 12，故隨機產生初始母代為 12 列 28 欄之矩陣，如表 4.7 所示。

表 4.7 初始母體矩陣

染色體	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	$k_1^a$			$k_2^a$			$k_3^a$			$k_4^a$			$k_1^d$		$k_2^d$		$k_3^d$		$k_4^d$		$k_5^d$		$k_6^d$		$k_7^d$		$k_8^d$	
1	1	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1	1
2	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	1	0	1	1
3	0	1	0	0	1	0	0	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	0	1	1
4	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0	1	1	1
5	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	0
6	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	0	0	0
7	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1
8	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	1	0	0	1	1	0	0
9	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1
10	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	1	1
11	0	1	0	0	1	0	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1
12	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1

4.2.2 演算範例之間距解碼

產生初始母體後，進行染色體解碼。進站路線間距  $k_i^a = 2^{p_i^a}$ ,  $i = 1, 2, 3, 4$  而離站路線間距  $k_j^d = 2^{p_j^d}$ ,  $j = 1, 2, 3, 4, 5, 6, 7, 8$ ，求得四家供應商進貨端路線發車間距冪次如下表 4.8 所示；以進站路線 1 之乘數冪次  $P_1^a$  為例，由 1, 0, 1 三個基因所組成，解碼後其值為  $2^0 \times 0 + 2^1 \times 0 + 2^2 \times 1 = 4$ ，而離站間距冪次亦以同樣方式求得。本研究為了避免零售賣場缺貨，提出進站貨車進站間距最大值必須小於或等於離站貨車的最小值，舉例來說，進站路線 1 間距乘數之冪次  $P_1^a = 4$  為最大值，而離站路線 3 之間距乘數冪次  $P_3^d = 1$ ，將 4 設定為離站路線可行之最小值，每一條離站路線乘數冪次加上此可行之最小值，可得  $P_3^d = 1 + 4 = 5$ ，其餘染色體則是以相同方式進行解碼。本研究採二冪策略，因此經解碼後得到的乘數值列表如 4.9 所示。

表 4.8 進站間距解碼

對應乘數	$P_1^a$			$P_2^a$			$P_3^a$			$P_4^a$		
染色體 1	1	0	0	1	0	0	0	0	0	0	0	1
間距幕次 $p_i^a$	4			4			0			1		

表 4.9 範例間距之染色體解碼

對應乘數	$k_1^a$	$k_2^a$	$k_3^a$	$k_4^a$	$k_1^d$	$k_2^d$	$k_3^d$	$k_4^d$	$k_5^d$	$k_6^d$	$k_7^d$	$k_8^d$
編碼值	4	4	0	1	3	1	1	2	1	0	1	3
												
乘數幕次	4	4	0	1	7	5	5	6	5	4	5	7
												
乘數值	16	16	1	2	128	32	32	64	32	16	32	128

### 4.2.3 演算範例之尋找基本週期

經過染色體編碼及解碼後，將針對這些隨機產生的染色體進行適應力評估。在獲得各染色體適應值之前，需要針對此組區間找出路線發車間距之最佳基本週期  $B^*$ 。尋找方式是根據運輸車輛類型的載運量限制，找尋此貨運成本曲線上之斷點，代入式(4.1)可以計算斷點位置。

假設進站路線 1 間距乘數為 32，貨車 A 載重量為 300kg/車，單位時間載運貨物量 225kg/天，則第一個斷點位置  $B_1 = \frac{300}{32 \times 225} = 0.0417$ 。表示在

$[0, 0.0417]$  之間需要一輛載重 300kg 的貨車；同理，第二個斷點位置

$B_2 = \frac{300 \times 2}{32 \times 225} = 0.0833$ ，表示在  $[0.0417, 0.0833]$  之間需要兩輛載重 300kg 的貨車，以此類推。

在已知各路線間距乘數的情況下，將基本週期上下界內，所有的斷點候選者代入目標函數式 3.15，再將目標函數式轉換為適應力函數值。

### 4.2.4 演算範例之評估適應值與選擇染色體

透過適應值評估後，各染色體可得一個適應值，根據此適應值挑選適應力最大之染色體。本研究欲追求成本最小化，因此目標函數越低者，其適應力越大，進行交配之機率亦越大。

以表 4.10 為例，說明如何運用正規排序適應值作為輪盤選擇機制之基礎。母體大小 ( $PS$ ) 為 12，假設選擇壓力 ( $SP$ ) 為 2，第 10 條染色體的平均總成本最大，令  $i_{temp} = 1$ ；第 9 條染色體的平均總成本最小，令  $i_{temp} = 12$ ，將相關

參數代入公式(4.3)，可求出適應值  $eval_{i_{temp}}$ 。例如：隨機產生一組  $U(0,1)$  的變數  $R = 0.6$  (0.5,0.65152)，選擇第 7 條染色體；接著產生第二組隨機變數  $R = 0.9056$  (0.8636,0.9697)，故選擇第 11 條染色體，依此類推。

表 4.10 母體中各染色體適應值函數

染色體	目標值	$i_{temp}$	適應值 $eval_{i_{temp}}$	$eval_{i_{temp}} / eval_{i_{temp}}$
10	15187	1	0	0
6	14897	2	0.18182	0.015152
12	11098	3	0.36364	0.045455
8	8027	4	0.54545	0.090909
1	7964.3	5	0.72727	0.151515
5	6084.9	6	0.90909	0.227273
2	4747	7	1.0909	0.318181
11	3359.9	8	1.2727	0.424239
3	3346.6	9	1.4545	0.545448
4	3128.9	10	1.6364	0.681814
7	3125.8	11	1.8182	0.833331
9	2459.4	12	2	1

#### 4.2.5 演算範例之交配

完成選擇程序之染色體，準備進行交配。研究中令交配率為 0.7，根據交配率決定挑選進行交配的染色體。以隨機方式產生一  $U[0,1]$  數值，當此隨機產生的數值低於交配率時，此染色體需要進行交配；若未低於交配率，此染色體不需進行交配，如示意圖 4.12 所示。

選擇出要進行交配之母體後，本範例採用用均勻交配方式。以染色體 1 及染色體 2 為例：首先判斷第一個位元，假設隨機產生一變數  $0.1813 < 0.5$ ，設其為 0，因此不進行交配。接著判斷第二個位元，假設隨機產生一變數  $0.59 > 0.5$ ，設其為 1，故進行交配，所以將第二個位元的值進行交換，如圖 4.13 所示，其餘染色體以此類推。

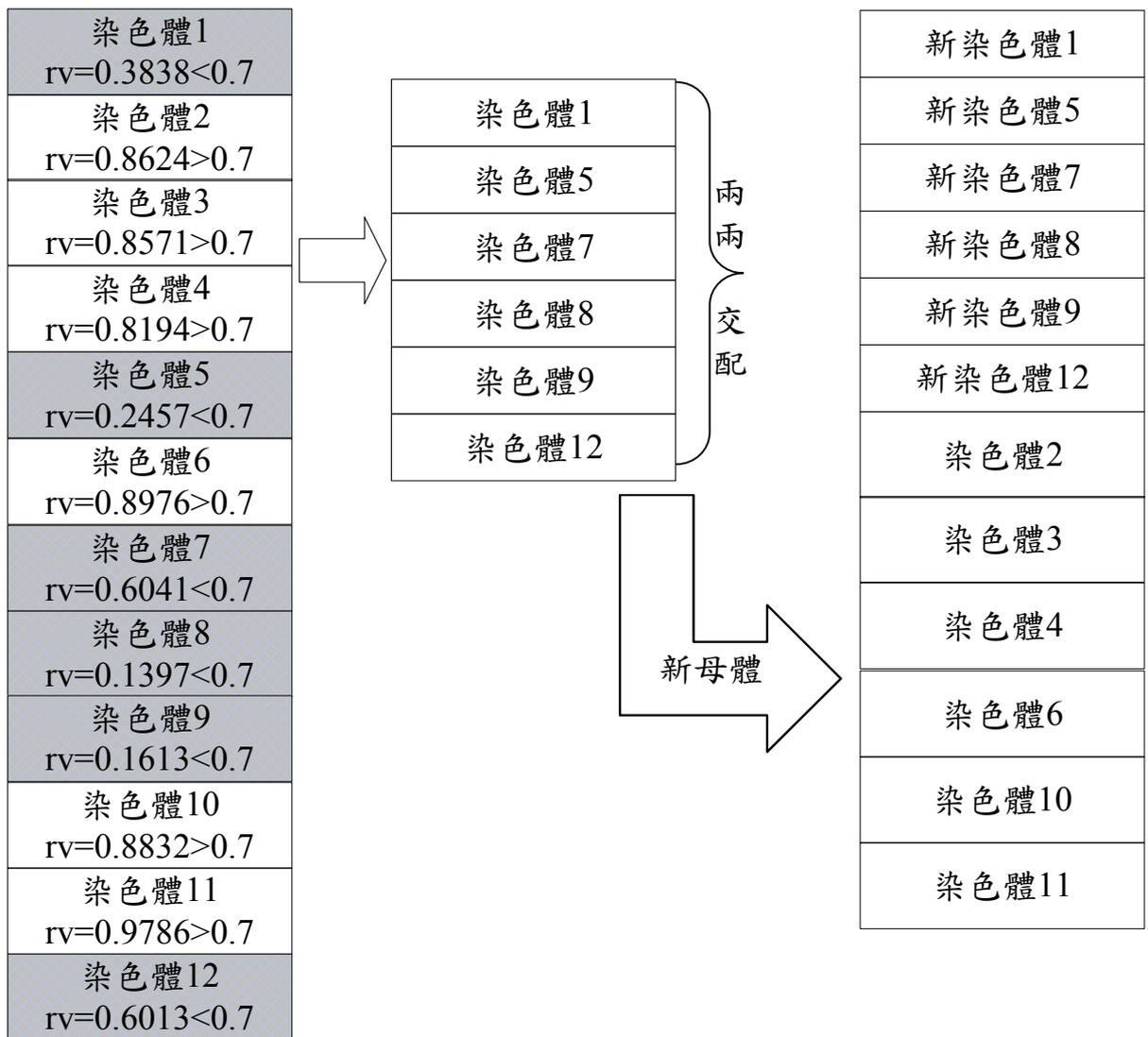


圖 4.12 範例中選擇交配之染色體示意圖

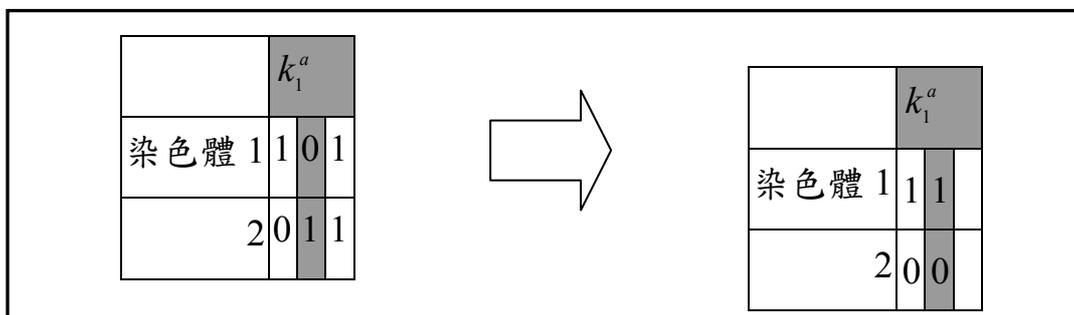


圖 4.13 均勻交配

#### 4.2.6 演算範例之突變

交配結束之後，為了避免陷入局部最佳解，運用突變跳脫局部解。突變之方式，同樣以隨機產生  $U[0,1]$  數值方式與突變率比較，當隨機值小於突變率

0.25 時，此二元位元需要改變。

以圖 4.14 為例，隨機產生一變數 0.0527 小於突變率 0.25，因此第一個位元需進行突變；同理，第二個位元產生隨機變數  $0.94 > 0.25$ ，故不進行突變。

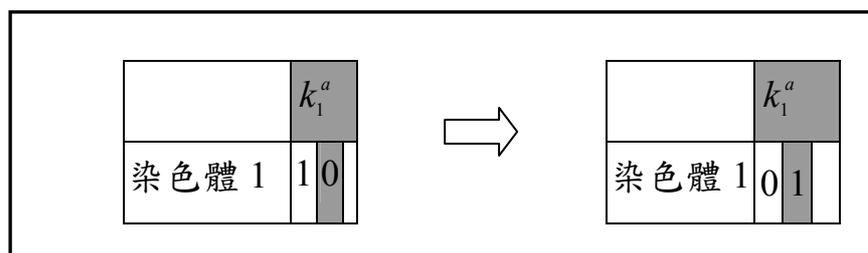


圖 4.14 突變機制

當完成上述運算，會獲得一個新的子代，再將此子代進行適應值評估，將最佳的染色體與對應之基本週期紀錄，同時將此子代複製為母代，若是尚未到達終止條件(本範例設定演化 100 代)，則此子代就可以持續進行演化，直到終止條件成立為止。

表 4.11 經 100 代演化後之目標函數值與基本週期

	TC	B
1	\$6597.1	0.008681
2	\$4613.6	0.046296
3	\$17445	0.092593
4	\$6324.3	0.009191
5	\$3911.4	0.011574
6	\$4629.7	0.005787
7	\$12035	0.034722
8	\$5119.8	0.046296
9	\$8190.7	0.023148
10	\$4254.5	0.023148
11	\$27884	0.023148
12	\$14263	0.028935

### 4.3 小結

在追求平均總成本為最低的目標下，本研究運用遺傳演算法求得(1)符合二冪策略的進站路線  $i$  的最佳進站間距乘數  $k_i^a$  與離站路線  $j$  的最佳離站間距乘數  $k_j^d$  的組合及(2)零售配銷體系之最佳基本週期  $B^*$ 。但是若要作為零售賣場經理人作相關決策之參考，仍需進一步考慮物流中心作業人員的產能資源限制，否則容易造成經理人的錯誤決策導致嚴重損失，因此，在下一章提出一啟發式方法，搭配於本研究的遺傳演算法，在滿足顧客需求條件下，進行物流中心分裝作業排程的加班工時最小化。

## 第五章 分裝作業排程的啟發式解法

本章提出加班工時最小化程序(Procedure for Overtime Minimization, Proc OTM)的啟發式解法，可有效藉由降低尖峰負載量，以求得物流中心總加班工時的人工成本最小。過去一般求解整數線性規劃法 (ILP) 的商用軟體，如 LINGO，所使用的分支界線法(branch and bound)常造成過度資料簿記負載問題導致求解較無效率。而運用使用基因演算法(GA)需進行選擇、交配、突變等運算過程，且經過多世代的演進，亦需耗費較多時間求得一組分裝作業排程解。第 5.1 節簡單介紹 ProcOTM，第 5.2 節為 ProcOTM 之初步排程程序，以產生一組初始作業排程。第 5.3 節介紹產品排程程序，將初始分裝作業排程一步步建構完成。在產生初始排程後，第 5.4 節利用排程撫平程序撫平物流中心分裝作業排程之負載，藉以使加班工時最小。最後，在第 5.5 節為加班成本最小化之演算程序。

### 5.1 ProcOTM 介紹

本研究利用 Yao(2001)所提之合理解測試程序(Procedure for Feasibility Testing, ProcFT)的啟發式方法為基礎，以協助判斷所求是否具有合理的排程解。黃健雄(2002)運用合理解測試程序，藉由降低尖峰負荷解決損耗性經濟批量排程問題(economic lot scheduling problem; ELSP)。林晏妃(2004)也運用可行解測試程序搜尋補貨乘數與基本週期的最佳合理可行解。藉由調整產品的補貨排程，可有效利用倉儲空間，以降低最大空間需求，使得排程下的最大倉儲空間需求能滿足倉儲空間限制。與前述合理解測試程序不同的是，本研究的目的是將加班成本降到最低，而並非只將尖峰負荷(peak load) 撫平。

過去一般研究大多假設物流中心內部人員的產能無限，並未考量人員有基本工作時數，此與現實狀況並不相符。因此本研究以合理解測試程序(Proc.FT)為基礎，欲利用 Proc OTM 判斷經由遺傳演算法求得一組間距乘數  $(\{k_i^a\}, \{k_j^d\})$  (詳見第四章)，在物流中心的作業排程是否需要加班工時：若不需要加班工時，則將排程輸出；而若經過 Proc OTM 程序後仍未得到不需加班的排程，則將加班工時產生的額外人工成本加入目標函數值，並重新計算新的適應力函數值。

Proc OTM 演算法之主要輸出是一組分裝 (loading /unloading) 作業排程

解，而目標函數值最小化的情形下求得之一組  $(\{k_i^a\}, \{k_j^d\})$  解為其主要輸入。由於目標函數的最佳化並未考慮分裝作業產能資源的限制，因此必須調整物流中心作業排程，以避免過多的批量產品在同一時間進行物流分裝作業，造成人員產能超過負載，而增加不必要的加班人工成本。當產能充裕，作業排程皆在基本工時時間內時，可以輕易的找出沒有額外加班成本的作業排程；但若產能利用率較高時，運用 Proc OTM 演算法可有效率的產生一組初始的作業排程，並藉由重複調整產品開始的作業基期，改善產能負載情形，最後產出將總加班成本最小的作業排程，並提供予決策者做相關決策。

茲定義 Proc OTM 中運用使用到的符號如下；W 矩陣表示一種可能的排程結果。其中矩陣的列數決定於進站間距乘數  $k_i^a$  及離站間距乘數  $k_j^d$  的最小公倍數，矩陣的行數由供應商及零售賣場數目總和決定。例如：目前有一間供應商及三間零售賣場，已知供應商 1 的進站間距乘數  $k_1^a = 2$ ，而三間零售賣場的離站間距乘數  $k_j^d = \{4, 2, 8\}, j = 1, 2, 3$ ，矩陣的列數以進站間距及離站間距的最小公倍數決定之，即  $lcm\{2, 4, 2, 8\} = 8$ ；行數為供應商加上零售賣場的數量，即  $1 + 3 = 4$ ，故得到矩陣的列數為 8 列，行數為 4 行，如表 5.1 所示。

表 5.1 分裝作業排程矩陣示意圖

	供應商 1	零售賣場 1	零售賣場 2	零售賣場 3
基期 1				
基期 2				
基期 3				
基期 4				
基期 5				
基期 6				
基期 7				
基期 8				

$\Omega(W)$  表示在物流中心 W 排程的所有基期中，批量產品之作業加班時數。研究中藉以調整供應商  $i$  及零售賣場  $j$  的批量產品應在哪一個基本作業週期開始進行分裝作業，使得物流中心的加班成本能夠最小化，因此  $\Omega(W)$  為本研究檢測 W 排程是否需要額外加班成本的指標。以表 5.2 為例，物流中心作業人員一天基本工時為八小時(即 480 分鐘)，已知產品的需求時間  $\sigma_i$  及間距乘數  $K$ ，若將零售賣場 3 的作業排在基期 1 進行分裝作業，此時基期 1 之總作業時間為  $306 + 208 + 586 = 1100$ ，加班工時  $\Omega(W^1) = 620$ ，若將零售賣

場3之作業排在基期4，加班工時明顯降低，以本例而言，得到 $\Omega(W^1)=34$ ， $\Omega(W^4)=106$ 。

表 5.2 分裝作業排程

	供應商1	零售賣場1	零售賣場2	零售賣場3	Total	加班工時
需求時間 $\sigma_i$	306	208	381	586		
乘數 $K$	4	2	4	4		
基期1	306	208	0	0	514	34
基期2	0	0	381	0	381	0
基期3	0	208	0	0	208	0
基期4	0	0	0	586	586	106

## 5.2 初步排程程序(ProcIS)

初步排程(The Initial Schedule Procedure, Proc. IS) 以貪婪式的方法為基礎，快速的產生一組初始作業排程，主要使用在下列兩種情形中：

1. 產生一組開始區域搜尋的初始分裝作業排程  $W$
2. 產生一組供重複再改善  $L^*$  的種子分裝作業排程

定義  $N$  表示所有供應商及零售賣場  $n$  種產品的集合， $\mathcal{G}$  表示產品之子集合，因此  $|\mathcal{G}| \leq n$ 。  $W(\mathcal{G})$  表示產品子集合  $\mathcal{G}$  的部分作業排程。當一組  $\{k_i^a, k_j^d\}$  第一次使用 Proc IS 時，產品子集合初始狀況為空集合，即  $\mathcal{G}=0$ ，產品子集合的初始排程為空排程  $W(\mathcal{G})=0$ 。依據 Proc PS（定義於 5.3 節）將產品的分裝作業期間指派到  $W(\mathcal{G})$ ，並更新相對應的產品子集合，直到  $n$  個產品全部被指派完畢。最後我們可以得到一個初始分裝作業排程即為  $W = W(N)$ 。接下來將繼續探討 Proc IS 被用在第 2 種情況，即當一組物流中心排程的可能解並不是第一次被使用時，研究假設  $W_0$  為經由前一次 Proc OTM 程序而無法再改善的最終排程結果。為了藉由重複 Proc OTM 程序來達到改善物流中心的工作負載並得到一組無須加班工時的排程解，隨機選取一部份的產品子集合，固定其分裝作業週期，將之指派到一個新的分裝作業排程  $W'$  中。令  $F$  代表所選取的產品子集合且  $\bar{F} = (N - F)$  表示尚未加入  $W'$  排程的產品集合。因為 Proc IS 扮演一個產生一組供重複再改善  $L^*$  的種子分裝作業排程的角色。令初始產品的子集合  $\mathcal{G} = F$ ， $W'(\mathcal{G}) = W_0(F)$ 。且將產品集合  $N$  扣除已加入分裝作業排程的

產品子集合  $F$ ，因此  $\mathbf{W}'(F)$  為一個新的初始部分分裝作業排程。尚未加入排程的產品子集合  $F$ 。則由 Proc IS 呼叫 Proc PS (定義於 5.2 節)。依據 LPT 法則來完成分裝作業排程。ProcIS 流程如圖 5.1 所示。

所謂 LPT 法則即是將  $F$  產品子集合中佔有較少物流中心處理時間的供應商或零售賣場的產品優先指派到物流中心上處理，直到所有產品均被指派完畢。(LPT 詳細法則請參照 Graham,1969 及 Pinedo,1993)，支持 LPT 啟發式法則可行的主要基本理由敘述如下：假設產品  $j$  在物流中心具有較長的處理時間  $\sigma_j$  且產品  $j$  在較後面才被指派時，面對已經完成的產品排程在每個分裝作業的基本週期已有沉重負載下，它可能會帶來更大的物流中心負載。這方法並不能保證優於其他學者所提出來的方法(Boctor,1987 及 Geng and Vickson,1988)。但是它可以幫助我們得到一個合理且不錯的初始分裝作業排程。初始步驟程序的範例參閱附錄 A。而 Proc IS 演算步驟程序說明如下：

步驟 1：檢驗門檻值  $\gamma$ 。

1.1 判斷  $\gamma = 0$ ，是則令  $\mathcal{G} = \emptyset$  和  $W(\mathcal{G}) = \emptyset$ ，並進入步驟 2

1.2 判斷  $\gamma > 0$ ，是則令  $\mathcal{G} = F$  和  $\mathbf{W}(\mathcal{G}) = \mathbf{W}(F) = \mathbf{W}_0(F)$ ，即將原先  $\mathbf{W}_0$  排程中之  $F$  產品集合排程維持不變，並進入步驟 2。

步驟 2：優先指派  $k_i$  乘數為 1 的產產品進入排程。

2.1 對所有未指派排程的產品  $\bar{F}$ ，若其  $k_i$  乘數為 1，則優先指派  $\sigma_i$  到  $W(\mathcal{G})$  排程，並更新  $\mathcal{G} = \mathcal{G} \cup \{j | j \in \bar{F} \text{ and } k_j = 1\}$ 。假使所有  $\bar{F}$  中  $k_j$  均大於 1，則略過步驟 2。

步驟 3：將  $i \in (N - \mathcal{G})$  的  $\sigma_i$  做降冪排列，並設  $[\mathbf{I}]$  表排序後的產品集合識別子

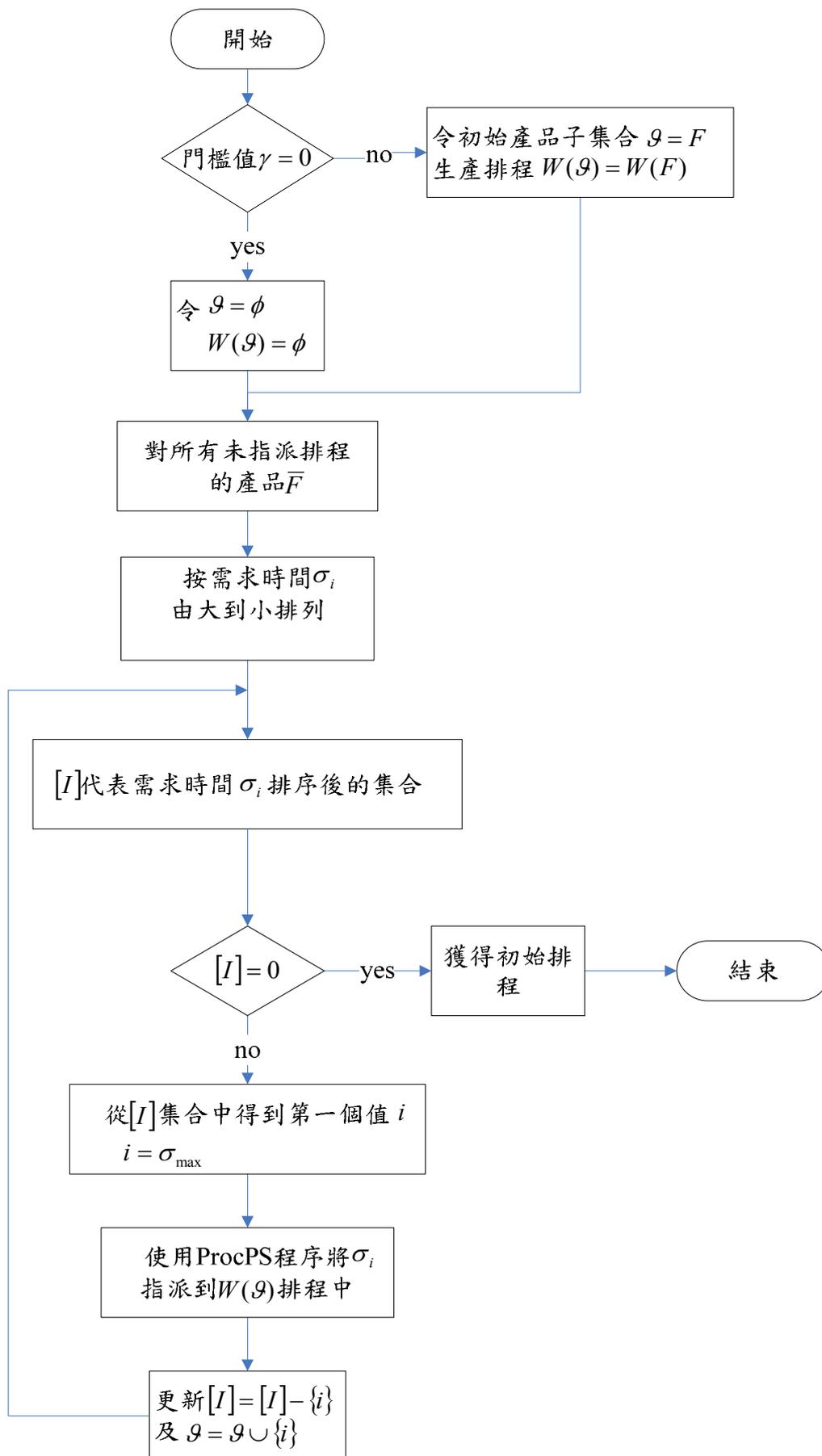


圖 5.1 初步排程程序流程圖

步驟 4：檢驗在  $[I]$  集合中的值：

4.1 判斷  $[I] = \phi$ ，是則進入步驟 4.1.1。

4.1.1 從  $[I]$  集合中得到產品  $\hat{i}$ ，且  $\hat{i} = \arg \max_{j \in [I]} \{\sigma_j\}$ 。

4.1.2 使用 Proc PS 程序將  $\sigma_{\hat{i}}$  指派到  $W(\mathcal{G})$  排程中。

4.1.3 將  $\mathcal{G}$  和  $[I]$  集合更新為  $\mathcal{G} = \mathcal{G} \cup \{\hat{i}\}$  和  $[I] = [I] - \{\hat{i}\}$ ，並回到步驟 4。

4.2 判斷  $[I] = \phi$ ，是則令  $W = W(N)$ ，並停止 Proc IS 演算程序。

### 5.3 產品排程程序(ProcPS)

產品排程程序(Product Scheduling Procedure, Proc. PS)主要的過程是在將初始分裝作業排程  $W$  一步一步地建構完成。ProcPS 的流程如圖 5.2 所示。

令  $\mathcal{G}$  表示初始產品的子集合，因此  $|\mathcal{G}| \leq n$ 。而  $W(\mathcal{G})$  則表示初始產品子集合  $\mathcal{G}$  的初始部分分裝作業排程。由於在合理解的測試過程中必須檢查每一個基本分裝作業週期的物流中心處理時間負載是否超過基本週期長度(基本工時)，因此必須決定在初始分裝作業部分排程中的基本分裝作業週期數目，其符號意義定義為  $k(\mathcal{G})$ 。 $k(\mathcal{G})$  可由初始產品子集合的  $\{k_i^a, k_j^d\}$  乘數之共同最小公倍數可決定之， $k(\mathcal{G}) = lcm(\{k_i^a\}, \{k_j^d\})$  且  $i, j \in \mathcal{G}$ 。假設批量產品  $i'$  下一個將被加入到初始產品的子集合，當產品  $i'$  的物流中心操做時間  $\sigma_{i'}$  指派到初始部分分裝作業排程  $W(\mathcal{G})$  時，必須先將基本分裝作業週期數目  $k(\mathcal{G})$  更新為  $k(\mathcal{G} \cup \{i'\}) = lcm\{k(\mathcal{G}), k_{i'}\}$ 。假設  $k(\mathcal{G} \cup \{i'\}) > k(\mathcal{G})$  時，新的初始部分分裝作業排程  $W(\mathcal{G} \cup \{i'\})$  必須將基本分裝作業週期數目增加為  $k(\mathcal{G} \cup \{i'\})$  而  $W(\mathcal{G} \cup \{i'\})$  在增加的基期中，產品子集合  $\mathcal{G}$  的排程則是複製原先  $W(\mathcal{G})$  分裝作業排程  $|k(\mathcal{G} \cup \{i'\})|/k(\mathcal{G})$  倍。則沿用  $W(\mathcal{G})$  分裝作業排程結構即基本分裝作業週期數目並不改變。由  $k_{i'}$  種將  $\sigma_{i'}$  指派到  $W(\mathcal{G} \cup \{i'\})$  得不同方法中，選擇可以使  $W(\mathcal{G} \cup \{i'\})$  有最小物流中心工作負載的方法來完成  $\sigma_{i'}$  的指派。Proc PS 演算步驟程序說明如下：

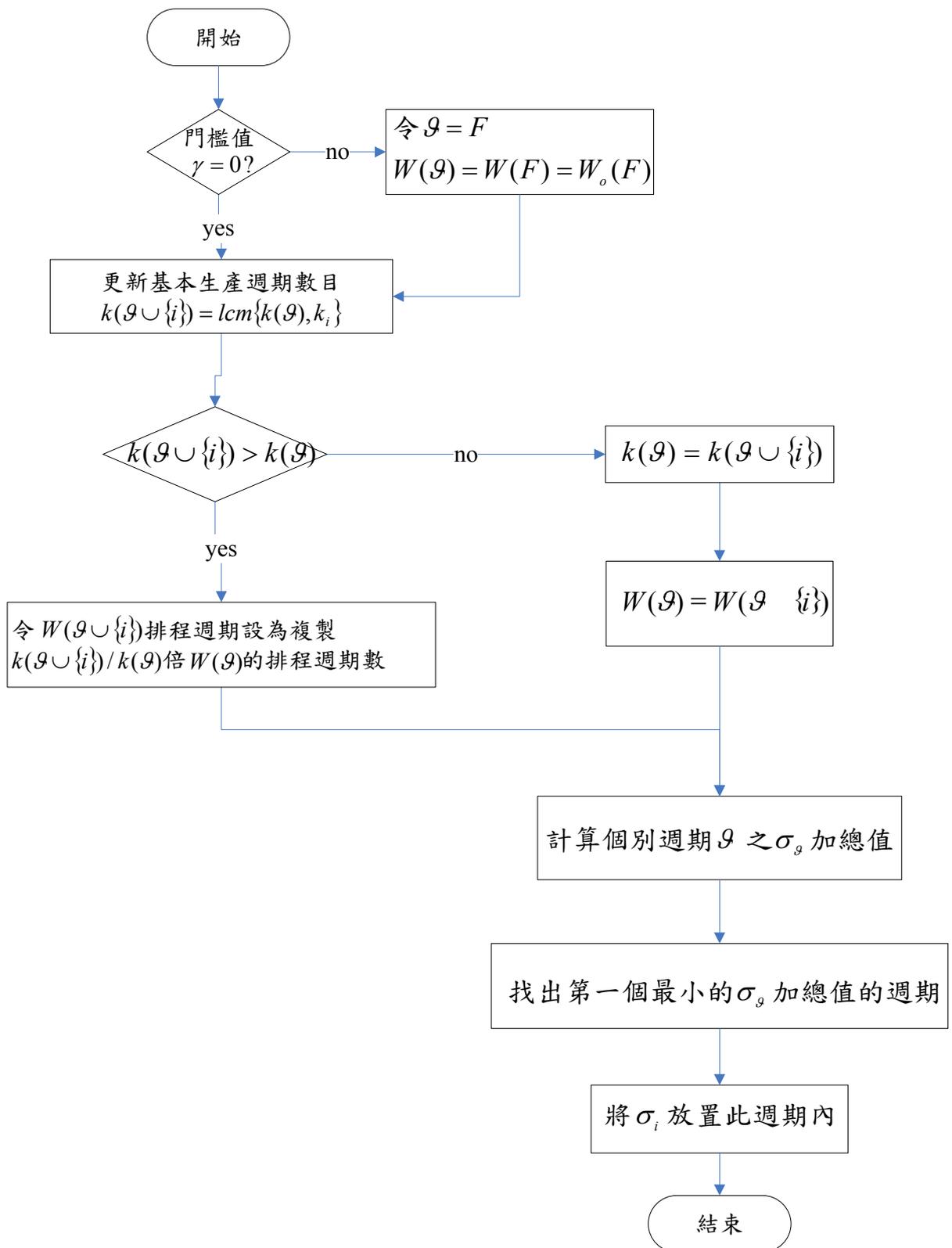


圖 5.2 產品排程程序流程圖

步驟 1：檢驗  $\gamma$  的值。

1.1 判斷  $\gamma = 0$ ，是則令  $\mathcal{G} = \phi$  和  $W(\mathcal{G}) = \phi$ ，並進入步驟 2。

1.2 判斷  $\gamma > 0$ ，是則令  $\mathcal{G} = F$  和  $W(\mathcal{G}) = W(F) = W_0(F)$ ，即將原先  $W_0$  排程中之  $F$  產品集合排程維持不變，並進入步驟 2。

步驟 2：將  $k(\mathcal{G} \cup \{i\})$  更新為  $k(\mathcal{G} \cup \{i\}) = lcm\{k(\mathcal{G}), k_i\}$

2.1 判斷  $k(\mathcal{G} \cup \{i\}) > k(\mathcal{G})$ ，是則令  $W(\mathcal{G} \cup \{i\})$  的排程週期設為複製  $k(\mathcal{G} \cup \{i\})/k(\mathcal{G})$  倍  $W(\mathcal{G})$  的排程週期數，並進入步驟 3。

2.2 判斷  $k(\mathcal{G} \cup \{i\}) = k(\mathcal{G})$ ，是則令  $W(\mathcal{G} \cup \{i\}) = W(\mathcal{G})$  並進入步驟 3。

步驟 3：計算  $R_t$  值。

$R_t$  為在初始部分分裝作業排程  $W(\mathcal{G})$  中的第  $t$  個基本分裝作業週期之目前物流中心負載量， $t = 1, \dots, k(\mathcal{G})$ 。

$$R_t = \sum_{i \in \mathcal{G}} w_{i\varphi}(k_i, t)(s_i + \beta_i)$$

$$l = \varphi(k_i, t) = \begin{cases} t \bmod k_i, & \text{if } t \neq \gamma k_i, \gamma \in N \\ k_i & , \text{if } t = \gamma k_i, \gamma \in N \end{cases} \text{ and } w_{il} = \{0, 1\} \text{ for all } i \text{ and } l$$

步驟 4：選擇  $w_{il}^*$

步驟 5：由步驟 4 所得的  $w_{il}^*$  更新  $W(\mathcal{G} \cup \{i\})$ 。

## 5.4 排程撫平程序(ProcSS)

排程撫平程序(Schedule Smoothing Procedure, Proc. SS)主要是被用來撫平由 Proc IS 程序所得到的物流中心初始分裝作業排程  $\mathbf{W}$  之負載。顧名思義，Proc SS 是用來調整分裝作業排程中的每一基本分裝作業週期中的物流中心負載，使其能夠最小化整個分裝作業排程範圍中最大的物流中心負載，並撫平每一個基本分裝作業週期中的物流中心負載以免超基本週期的長度。Proc SS 程序主要是經由下述 3 個子程序來完成，分別是移出程序、成對互換程序和 2 對 1 互換程序。移出程序(Smooth-out routine)主要的目地在從擁有最大物流中心負載的基期中，將某個  $k_i > 1$  的產品之  $\sigma_i$  指派到其他基本分裝作業週期。成對互換程序(Pair-exchange routine)則是從擁有最大物流中心負載的基期中，將某個  $k_i > 1$  的產品之分裝作業排程與非最大物流中心負載的基本分裝作

業週期中的某個  $k_i > 1$  的產品分裝作業排程互換。2 對 1 互換程序(Two-to-one exchange routine)則是從擁有最大物流中心負載的基期中，將某個  $k_i > 1$  的產品之分裝作業排程與非最大物流中心負載的基本分裝作業週期中的 2 個  $k_i > 1$  的產品分裝作業排程互換。排程撫平程序的流程如圖 5.3 至圖 5.5 所示，範例及詳細演算程序參閱附錄 A。

在執行 Proc SS 演算程序中，本研究使用一些新的參數，其定義如下所示：

$\tau$ ：在物流中心擁有最大工作負載的基期。

$W^m$ ：在每次個別執行 Proc OTM 程序中，及區域搜尋或重複改善  $\Omega^*$  的過程，所得到的最小化物流中心工作負載的分裝作業排程。

$\Omega(W^m)$ ： $W^m$  分裝作業排程中，最大的物流中心工作負載量。

$W^*$ ：第一次運用 Proc IS 所得初始分裝作業排程設為最佳分裝作業排程。

$\Omega(W^*)$ ： $W^*$  分裝作業排程中，最大的物流中心工作負載量。

$W$ ：一種可能的排程結果。

$\Omega(W)$ ：使用  $W$  排程，在所有基本作業週期中，物流中心最大工作負載量。

$\tau(\lambda, W^m)$ ： $\{i | w_{i, g(k_i, \tau)} = 1 \text{ and } k_i > 1\}$ ，即位在最大物流中心工作負載的基本作業週期上分裝作業且  $k_i$  乘數大於 1 的產品集合。

$\Lambda(\lambda, W^m)$ ： $\{i | w_{i, g(k_i, \tau)} = 0 \text{ and } k_i > 1\}$  即非位在最大物流中心工作負載的基本作業週期上分裝作業且  $k_i$  乘數大於 1 的產品集合。

$\chi$ ：連續使用 Proc OTM 程序且無法改善  $L^*$  的次數。

## 5.5 加班成本最小化演算程序

以下詳細介紹加班成本最小化程序(Proc OTM)的細部流程，流程如圖 5.6 所示。令  $\gamma$  為判別是否第一次使用 Proc IS 程序的標記， $\chi$  表示連續使用 Proc OTM 程序且無法改善  $\Omega^*$  的次數， $W^m$  表示每次個別執行的 Proc OTM 程序中即區域搜尋或重複改善  $\Omega^*$  的過程，所得物流中心工作負載的分裝作業排程最小化。 $\Omega(W^m)$  表示  $W^m$  分裝作業排程中，最大的物流中心工作負載量。令  $W^{FT}$  和  $\Omega(W^{FT})$  分別表示經由 Proc OTM 程序所得到合理分裝作業排程和基本分裝作業週期中的物流中心最大負載量。Proc OTM 演算步驟程序說明如下，演算細部流程圖則如圖所示：

步驟 1：設定初始參數

令  $\gamma=0$ ， $\chi=0$  且  $\phi=0$ 。並讀入 Y 參數

步驟 2：藉由 Proc IS 程序得到一組初始分裝作業排程  $W$  並開始區域搜尋。

2.1 令  $W^m = W$  和  $\Omega(W^m) = \Omega(W)$

2.2 判斷  $\gamma=0$ ，是則令  $\Omega^*(W^*) = \Omega(W^m)$  和  $W^* = W^m$

2.3 判斷  $\Omega(W^m) \leq B$ ，是則令  $\Omega(W^{FT}) = \Omega(W^m)$  和  $W^{FT} = W^m$ ，並進入步驟 4，為否則進入步驟 3。

步驟 3：應用 Proc SS 程序改善  $W^m$  的  $\Omega(W^m)$ 。

3.1 使用移出程序來改善。

3.1.1 判斷  $\Omega(W^m) \leq B$ ，是則令  $\phi=1$ 、 $\Omega(W^{FT}) = \Omega(W^m)$  和  $W^{FT} = W^m$ ，並進入步驟 4，為否則進入步驟 3.1.2。

3.1.2 判斷  $\Omega(W^m) > B$  和  $\Omega(W^m)$  是否已被改善，是則回到步驟 3，否則進入步驟 3.2。

3.2 使用成對互換程序來改善。

3.2.1 判斷  $\Omega(W^m) \leq B$ ，是則令  $\phi=1$ 、 $\Omega(W^{FT}) = \Omega(W^m)$  和  $W^{FT} = W^m$ ，並進入步驟 4，為否則進入步驟 3.2.2。

3.2.2 判斷  $\Omega(W^m) > B$  和  $\Omega(W^m)$  是否已被改善，是則回到步驟 3，否則進入步驟 3.3。

3.3 使用 2 對 1 互換程序來改善。

3.3.1 判斷  $\Omega(W^m) \leq B$ ，是則令  $\phi=1$ 、 $\Omega(W^{FT}) = \Omega(W^m)$  和  $W^{FT} = W^m$ ，並進入步驟 4，為否則進入步驟 3.3.2。

3.3.2 判斷  $\Omega(W^m) > B$  和  $\Omega(W^m)$  是否已被改善，是則回到步驟 3，否則進入步驟 3.4。

3.4 判斷  $\Omega^*(W^*)$  是否已經改善。

3.4.1 判斷  $\Omega(W^m) < \Omega^*(W^*)$ ，是則令  $\chi=0$ 、 $\Omega^*(W^*) = \Omega(W^m)$  和  $W^* = W^m$ ，並進入步驟 4，否則進入步驟 3.4.2。

3.4.2 判斷  $\Omega(W^m) \geq \Omega^*(W^*)$ ，是則令  $\chi=1$ ，並進入步驟 4。

步驟 4：判斷 Proc OTM 程序是否已經達到終止條件。

4.1 判斷  $\phi = 1$ ，是則停止 Proc OTM 演算，並輸出  $W^{FT}$  和  $\Omega(W^{FT})$ 。

4.2 判斷  $\chi > Y$  且  $\phi = 0$ ，是則停止 Proc OTM 演算，並計算加班工時的最小人工成本。

4.3 判斷  $\chi \leq Y$  且  $\phi = 0$ ，是則隨機選取  $[n/2]$  個產品並標記為  $\bar{F}$ ，令  $\gamma = 1$ ，並進入步驟 2 再次改善。

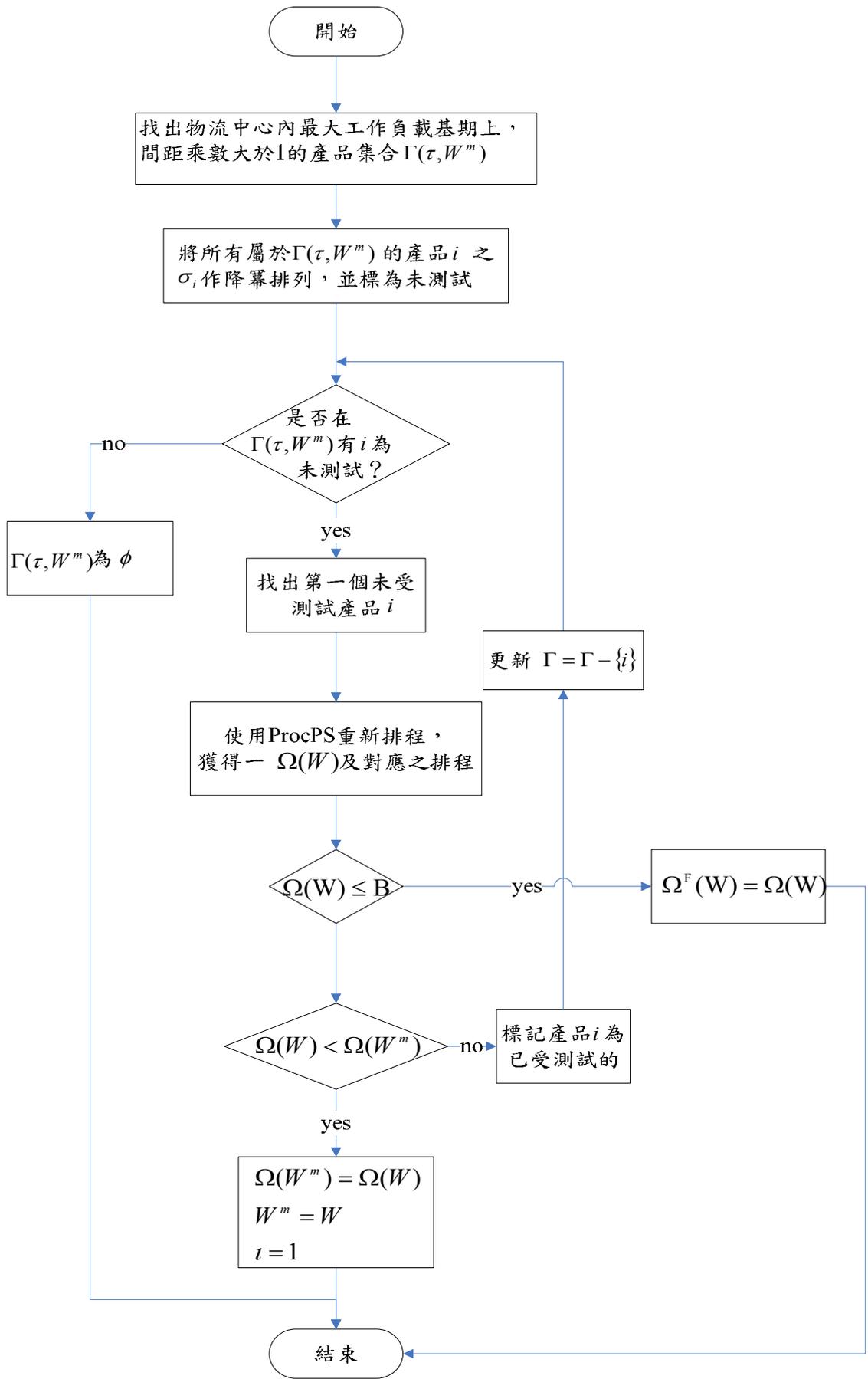


圖 5.3 移出程序流程圖

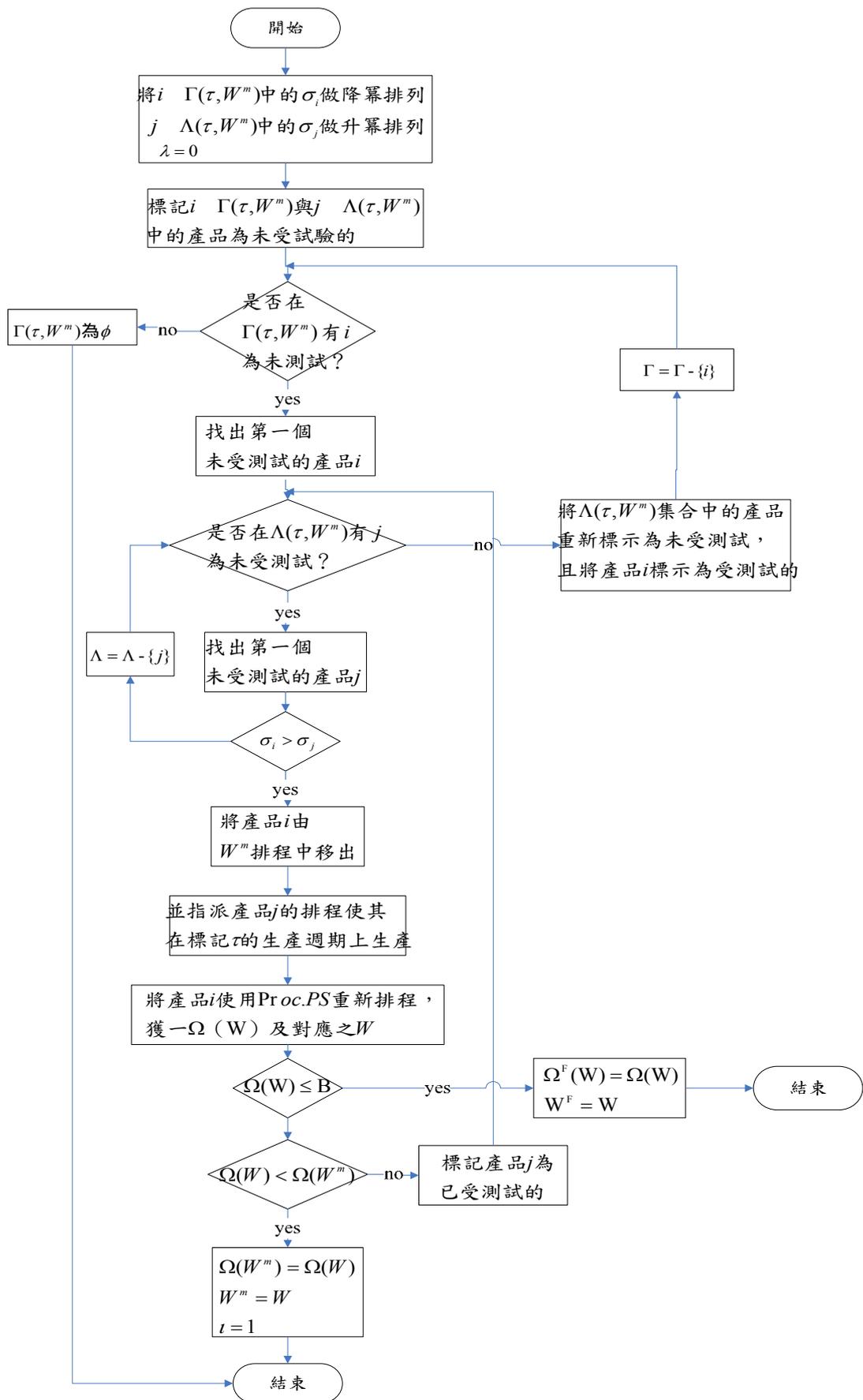


圖 5.4 成對互換程序流程圖

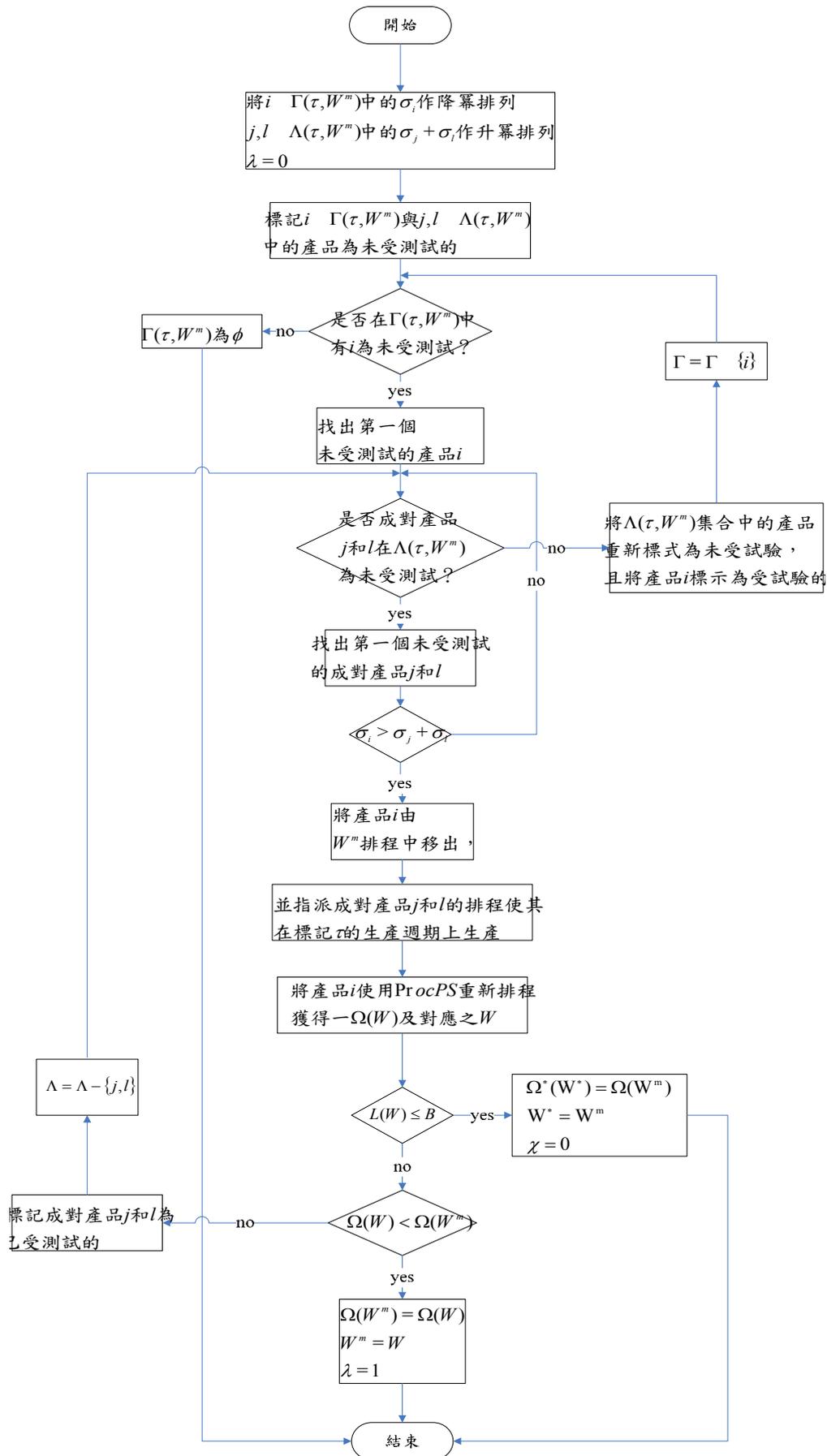


圖 5.5 二對一互換程序流程圖

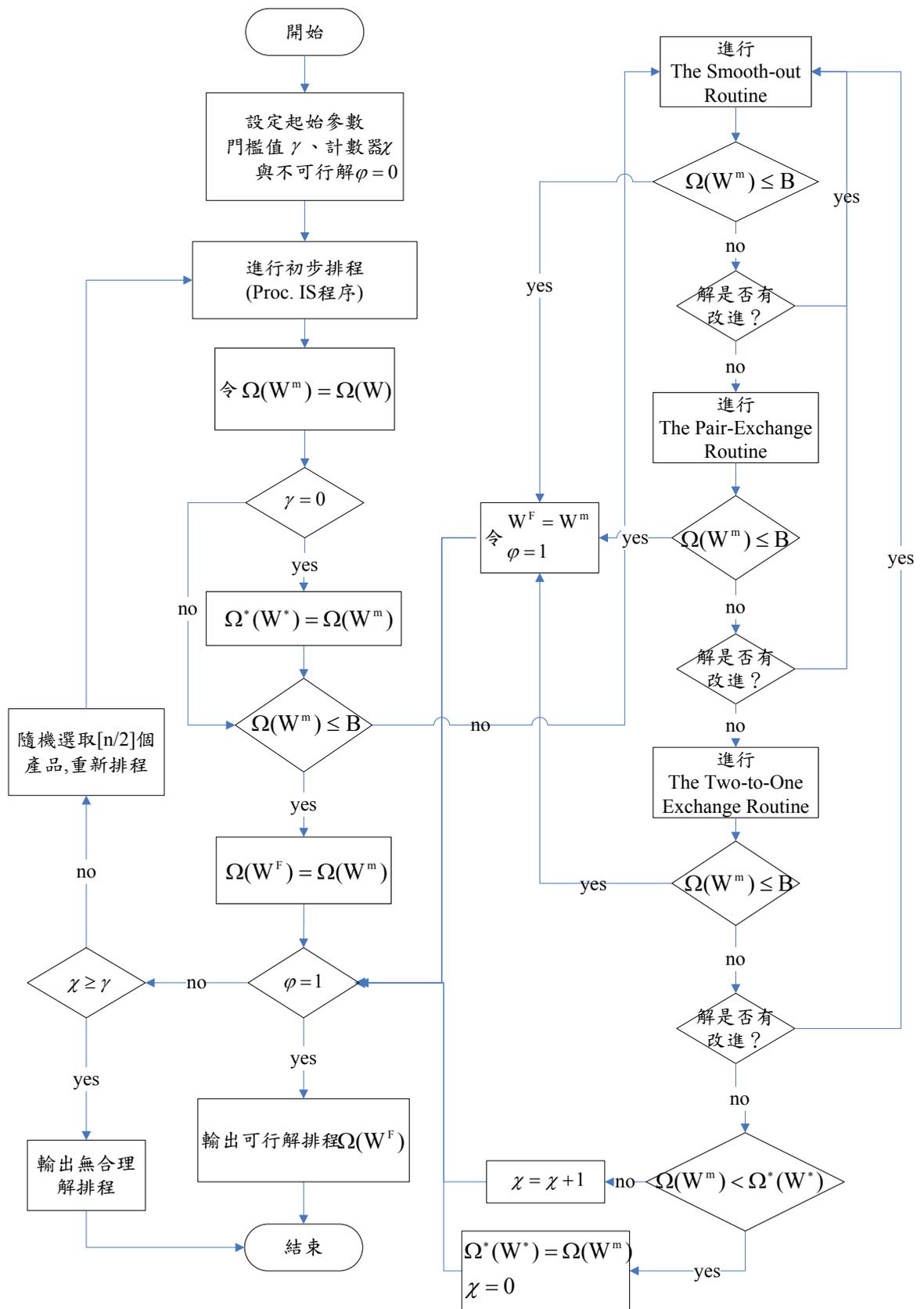


圖 5.6 加班成本最小化演算程序流程圖

## 第六章 數據範例

本章節主要目的在驗證第四章的遺傳演算法求得進站與離站間距乘數，以作為第五章加班成本最小化程序測試演算法的輸入參數。並將物流作業的加班成本加入目標函數式當中，經過 GA 演化的機制，求得最佳平均總成本。第 6.1 節介紹研究工具，第 6.2 節結合遺傳演算法與分裝作業排程之啟發式方法，以一個完整的例子進行數據實驗以介紹所提出之求解方式。

### 6.1 研究工具

Matlab 是 Matrix Laboratory 的簡寫，它是一個直譯式的語言程式，與其他的語言程式比較起來，因為語法較為單純，所以容易學習與了解。Matlab 主要特色在於他的數值分析、矩陣的運算與資料型態的轉換，它能夠讓使用者即使面對大量的資料以及冗長的運算問題時，也能有效的解決。因此，本研究選擇 Matlab7.0 作為演化最佳化問題的解決工具。另外，本研究的實驗模擬環境在軟、硬體部份，所使用的中央處理器(CPU)為 Pentium 1.6G HZ，而隨機存取記憶體(RAM)為 523,760KB。

### 6.2 數據實驗

本研究以四家供應商和八家零售商為範例進行實驗及流程說明，流程圖如圖 6.1 所示。本研究流程步驟詳述如下，

步驟 1：定義初始參數

1.1 遺傳演算法的相關參數，包括：交配率( $P_c$ )=0.7、突變( $P_m$ )=0.25、母體大小(Population size)=24、繁衍世代(generation)=100、選擇壓力(SP)=2

1.2 零售配銷體系相關參數：本研究假設四家供應商及八家零售商，供應商與零售賣場之供給需求參考如表 4.2 所示，貨車費用及可載運貨物量如表 4.3，系統成本參數如表 4.4 所示。

步驟 2：產生初始母體

設定母體大小為供應商加零售賣場數目之兩倍，染色體長度由供應商端及零售賣場端之間距乘數冪次方決定(詳見第 4.2.1 節)，產生染色體長度為 28，母體大小為 24 的初始母體。

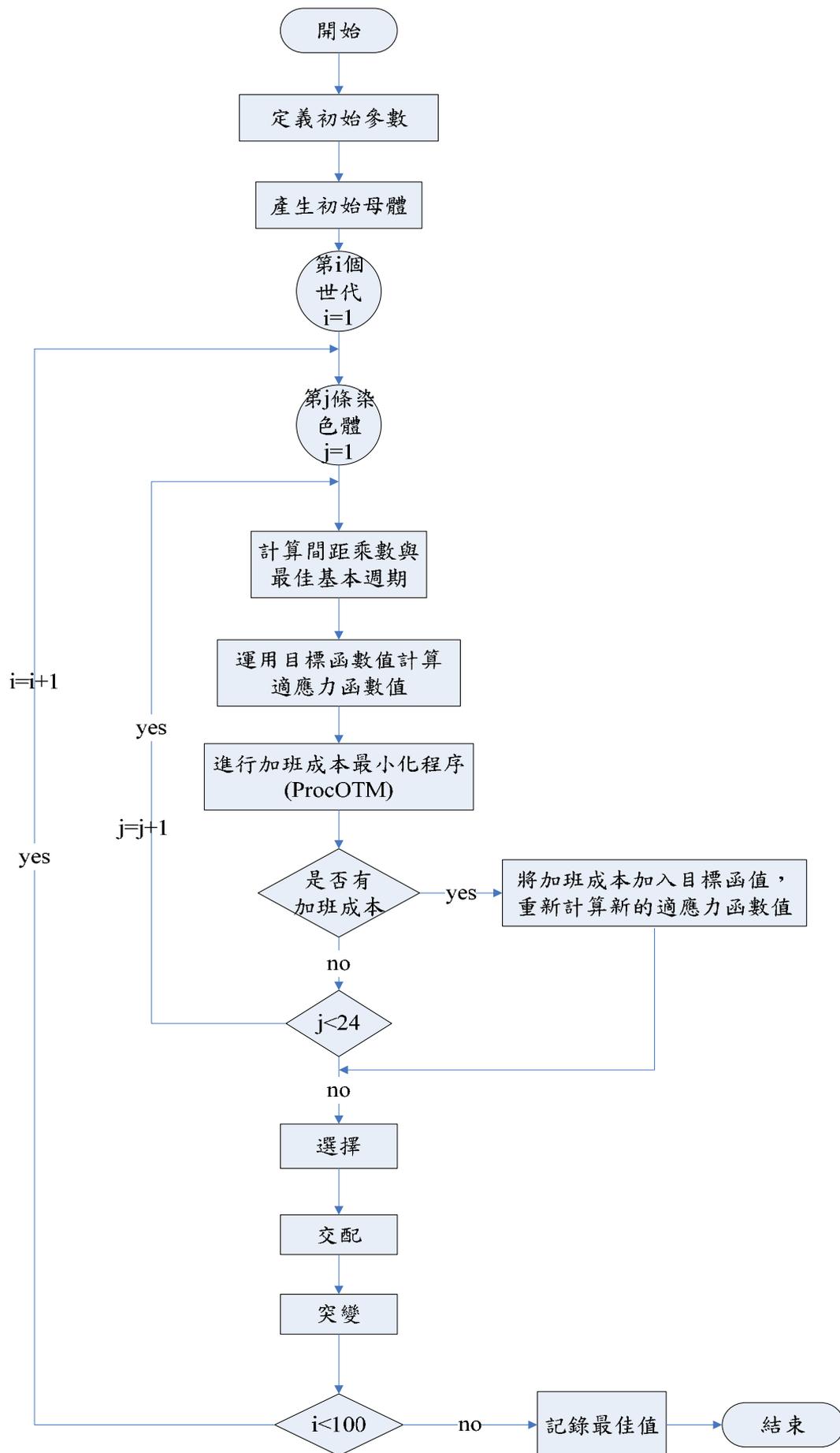


圖 6.1 結合遺傳演算法與分裝作業排程流程圖

### 步驟 3：計算間距乘數與最佳基本週期

以第一代的第九條染色體為例，將染色體基因進行解碼，計算供應商端與零售賣場端之間距乘數，如下表 6.1 所示。在已知一組間距乘數的狀況下，找出此組間距在最小成本下的最佳基本週期( $B^*$ )=0.017361，最小成本 \$11544。

表 6.1 第一代的第九條染色體解碼後之間距乘數

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
染色體 9	1	0	1	1	1	1	0	1	0	0	1	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0
	5		7		2		3		2		3		2		0		2		0		0		0		0			
乘數冪次	5		7		2		3		9		10		9		7		9		7		7		7		7			
間距乘數	32		128		4		8		512		1024		512		128		512		128		128		128		128			

### 步驟 4：進行分裝作業排程之啟發式演算法

遺傳演算法過程中並未考慮物流中心人員產能限制，物流中心作業人員一天工作時數為 8 小時，在基本假設式(5)中假設物流中心的產品一定會在當天完成分裝作業，當分裝作業時間超過一天基本工作時數，以加班成本計算之。因此，物流中心若有加班成本，將加班成本加入目標函數值當中；反之，目標函數值則為原本求得之目標函數值。物流中心對每一供應商及零售商的產品處理需求時間如表 6.2 所示。

表 6.2 單位產品平均花費時間(單位：分/kg)

供應商 1	供應商 2	供應商 3	供應商 4	零售商 1	零售商 2
0.016	0.0146	0.0094	0.0125	0.0061	0.0039
零售商 3	零售商 4	零售商 5	零售商 6	零售商 7	零售商 8
0.0032	0.0026	0.0031	0.0025	0.0034	0.0032

產品在物流中心內所耗用的時間為每單位產品平均花費時間乘上每一次之載運批量，以供應商 1 為例，在物流中心內進貨端所需時間為  $225 \cdot 0.016 \cdot 32 = 115.2$  分鐘，其餘依此類推。運用 ProcOTM，經過排程撫平程序之移出、成對及二對一互換三個子程序，撫平作業排程的尖峰負荷，希望使總加班時數最小。在輸入間距乘數及供給需求後，加班成本最小化程序的啟發式解法得到之總加班時數為 196.608 分鐘。根據行政院主計處

2004 年統計，服務業受雇員工每人平均薪資率為 3.2 元/分鐘，在此組間距乘數下，物流中心之總加班成本為  $3.2 \times 196.608 = \$629.149$ 。將加班成本加入原本求得之目標函數當中，在考量產能限制下，物流中心真正的成本函數值為  $11544 + 629.1496 = \$12173.1496$ ，其餘染色體以此類推，得到第一代目標函數值如表 6.3 所示。

表 6.3 第一代染色體之目標函數值

染色體	目標函數值	最佳基本週期
1	24448	0.0217
2	21459	0.09208
3	11269	0.01157
•	•	•
•	•	•
•	•	•
8	15062	0.04934
9	12173	0.01736
10	8915.2	0.02315
•	•	•
23	9621.2	0.02315
24	10803	0.00579

本研究在此加入一個參照的表格(look up table)，如表 6.4 所示，若求得之某組參數組合與之前求得之參數完全相同時，則不需要進行 ProcOTM 的計算流程，可以直接由表格當中擷取需要的平均總成本(TC)，如此可以節省排程之作業時間。舉例來說，若今求得之參數組合  $\{k_i^a, k_j^d, B\}$  比對後發現與第四條染色體完全相同，此時不需要再次進行加班成本最小化程序，即可得到平均總成本為 \$7,268。

表 6.4 參照表格

染色體	$k_i^a$				$k_j^d$								B	TC
1	64	1	8	128	128	512	256	512	128	512	256	1024	0.00579	\$2,771
2	1	16	16	8	16	32	16	128	128	64	16	128	0.0463	\$4,436
3	1	32	2	32	128	128	256	128	32	64	64	64	0.03472	\$5,564
4	32	8	2	8	256	32	256	64	32	128	32	64	0.09259	\$7,268
•	•				•								•	•
•	•				•								•	•
•	•				•								•	•
998	4	8	2	2	8	16	32	32	8	64	64	8	0.19737	\$8,704
999	8	32	128	16	256	512	1024	256	512	1024	1024	1024	0.01157	\$10,023
1000	32	1	128	16	1024	128	1024	1024	1024	512	512	1024	0.01157	\$10,526

步驟 5：適應值評估

求得各染色體適應值後，運用線性排序正規劃進行適應值評估。適應值評估方式如式(4.3)，得到之適應函數值如表 6.5 所示。適應值是作為選擇步驟之標準，適應力高的染色體在下一代將被複製，而適應力低的則被淘汰。

步驟 6：遺傳演算子

6.1 選擇：研究中採用輪盤式選擇法，透過機率原理，將適應函數值轉換為輪盤面積，適應值越大，在輪盤上面積越大，被挑選的機率也相對越高。舉例來說，產生隨機變數=0.6834，會選擇母代之第十五條染色體作為進行交配的第一條染色體，以此類推。選擇出的染色體如下表 6.5 所示。

6.2 交配：經由步驟 6.1，選出欲進行交配之染色體，進行判斷機制，當交配門檻值( $P_c = 0.7$ )超過所產生的機率值才進行交配運作。舉例來說，產生隨機變數  $rv = 0.3388 < 0.7$ ，故採第一條染色體進行交配，依此類推。將染色體兩兩進行交配以產生新的子代。本研究採用均勻交配。

6.3 突變：當產生的隨機(random)變數小於突變率，該基因進行突變機制，反之則不進行突變的機制。

表 6.5 第一代染色體之適應力函數值

染色體	適應函數值	累積機率
1	1.0435	0.043478
2	1.4783	0.10507
3	0.34783	0.11957
4	0.17391	0.12681
5	0.086957	0.13043
6	1.6522	0.19928
7	0.26087	0.21014
8	0.6087	0.23551
9	1.7391	0.30797
10	1.8261	0.38406
11	1.3043	0.43841
12	2	0.52174
13	1.5652	0.58696
14	0.69565	0.61594
15	1.913	0.69565
16	0.78261	0.72826
17	0.95652	0.76812
18	0	0.76812
19	1.1304	0.81522
20	0.52174	0.83696
21	0.86957	0.87319
22	0.43478	0.8913
23	1.3913	0.94928
24	1.2174	1

步驟 7：重複

將突變過後的子代當作新的母代，判斷是否達到終止條件。研究中設立的終止條件為一固定的演化世代，設為 100 代。如判斷未達到終止條件，回到步驟 3 繼續進行演化，直到終止條件成立為止。經過 100 代演化後，得到最終目標函數值( $TC$ )、最佳基本週期( $B^*$ )及間距天數  $\{k_i^a, k_j^d\}$  如表 6.6 所示，得到最佳平均總成本為  $TC_{\text{本研究}} = \$5347.2$ 。

茲以同樣的母體大小=24，計算黃紹維(2006)執行 100 代之後之平均總成本，基本週期與間距乘數的基本資料如表 6.7 所示，其中，該模式並未考慮物流中心人員產能限制，此與現況不符合。將表 6.7 中的 K 值  $\{k_1^a, \dots, k_4^a, k_1^d, \dots, k_8^d\}$  代入 ProcOTM 當中計算是否有加班成本(詳見第五

章)，將黃紹維(2006)之原本的平均總成本、間距天數代入後所得之加班成本與考量加班成本後之真正平均總成本列如表 6.8 所示。原本平均總成本為\$5847，在考量物流中心人員產能限制後，得到之平均總成本為  $TC_{Huang} = \$6,476.15$ 。茲定義黃紹維(2006)所得結果與本研究所得結果，而本研究與 Huang 之平均總成本差異度為

$$I = \frac{TC_{Huang} - TC_{本研究}}{TC_{本研究}} \quad (6.1)$$

經計算，可得  $I = (6,476.15 - 5347.2) / 5347.2 = 21.11\%$ 。由此可知，若經理人未考慮超時作業成本，其不當之發車排程決策將導致企業重大的損失。

表 6.6 本研究執行 100 代之結果

	$TC$	$B^*$	$k_i^a$				$k_j^d$							
1	\$8642.3	0.034722	1	2	32	8	32	64	32	64	128	256	256	64
2	\$8291.7	0.092593	16	8	2	8	64	64	16	32	16	16	64	64
3	\$7969.7	0.023148	16	4	64	32	128	256	128	64	512	256	128	256
4	\$8908.9	0.033482	64	64	32	8	512	128	256	64	256	256	256	64
5	\$5347.2	0.069444	1	1	1	16	64	128	32	32	32	16	64	32
6	\$21616	0.052426	4	64	16	2	64	128	256	512	512	256	64	256
7	\$19167	0.046296	2	64	1	2	512	128	64	128	512	256	512	128
8	\$6203.7	0.029762	32	2	2	64	64	512	128	256	64	128	64	64
9	\$11108	0.023148	64	32	2	8	128	128	128	64	128	256	256	256
10	\$5602.6	0.017361	4	16	8	128	128	128	512	128	256	128	512	128
11	\$9772.4	0.011574	16	32	1	128	128	1024	128	1024	128	128	256	512
12	\$9732.5	0.032895	16	64	1	8	256	128	128	512	64	64	512	64
13	\$11870	0.014468	16	128	8	64	128	512	512	1024	256	256	128	1024
14	\$12640	0.014468	16	128	128	16	512	256	512	512	512	256	1024	1024
15	\$6464.7	0.023148	4	4	64	32	128	512	512	64	128	128	256	64
16	\$7084.5	0.009191	16	32	64	128	128	128	128	1024	512	1024	256	128
17	\$6793.4	0.014881	32	128	1	32	256	512	256	128	512	512	512	128
18	\$6346.3	0.017361	1	4	64	32	128	256	256	64	64	256	256	128
19	\$8390.2	0.017361	16	32	32	16	256	256	64	256	256	32	128	128
20	\$11005	0.034722	64	1	64	8	128	512	128	128	64	512	256	128
21	\$5848.2	0.007710	2	2	32	128	1024	256	256	1024	256	512	1024	256
22	\$5361.8	0.008681	4	2	16	128	256	256	256	128	256	512	256	256
23	\$6827.4	0.009191	64	4	4	128	128	128	256	1024	256	128	1024	128
24	\$6287	0.009191	1	128	2	128	512	1024	512	128	1024	128	128	128

表 6.7 黃紹維(2006)執行 100 代之結果(未考慮人員產能限制)

	TC	$B^*$	$k_i^a$				$k_j^d$							
1	\$20713	0.03084	8	128	64	4	1024	1024	256	256	128	512	128	1024
2	\$7012.4	0.00868	2	64	8	64	512	128	64	512	256	256	256	256
3	\$28957	0.05787	64	128	2	1	128	256	512	256	512	128	128	256
4	\$7534.5	0.06944	4	8	1	8	64	16	64	64	16	16	8	32
5	\$6980	0.00901	32	1	1	128	128	512	1024	1024	1024	512	128	128
6	\$14803	0.02894	64	8	1	8	512	64	256	128	512	64	64	512
7	\$13755	0.05208	4	32	32	4	256	256	256	256	128	64	128	256
8	\$5847	0.01584	128	128	32	64	1024	128	256	256	128	512	256	128
9	\$6939.2	0.00434	2	128	32	128	1024	256	128	512	512	512	1024	512
10	\$18362	0.04630	4	2	64	2	512	64	512	128	256	256	64	128
11	\$26823	0.03701	4	128	64	2	512	128	1024	1024	512	128	256	256
12	\$8034.2	0.18519	2	4	4	4	32	8	32	32	32	8	32	32
13	\$12976	0.01447	128	32	64	16	256	512	512	1024	512	256	512	1024
14	\$11414	0.07212	32	32	1	4	32	64	256	64	32	128	256	64
15	\$7745.4	0.01157	32	16	128	64	1024	1024	512	256	512	128	256	512
16	9294.2	0.01157	128	16	1	32	128	256	512	512	256	256	1024	512
17	\$11644	0.00579	128	32	128	64	256	1024	128	1024	1024	128	1024	1024
18	\$9708.9	0.02315	64	8	64	64	128	512	64	512	64	64	64	256
19	\$7951.4	0.00434	2	16	8	128	1024	128	1024	1024	128	256	1024	512
20	\$20600	0.09259	4	8	32	1	256	64	64	256	32	256	64	64
21	\$14195	0.01447	128	4	1	32	128	256	128	1024	128	128	512	1024
22	\$9697.5	0.00579	128	64	8	128	128	256	512	256	1024	1024	512	1024
23	\$9450.1	0.02315	4	1	1	32	128	64	128	64	256	32	256	256
24	\$9137.2	0.00868	2	2	64	64	64	64	128	512	512	512	64	256

表 6.8 將黃紹維(2006)成本項加入超時作業成本

	原本的平均總成本	Overtime cost	Huang 的平均總成本
1	\$20713	\$629.15	\$21,342.15
2	\$7012.4	\$0	\$7,012.40
3	\$28957	\$355.37	\$29,312.37
4	\$7534.5	\$0	\$7534.5
5	\$6980	\$105.43	\$7,085.43
6	\$14803	\$0	\$14,803.00
7	\$13755	\$0	\$13,755.00
8	\$5847	\$629.15	\$6,476.15
9	\$6939.2	\$629.15	\$7,568.35
10	\$18362	\$0	\$18,362.00
11	\$26823	\$711.88	\$27,534.88
12	\$8034.2	\$0	\$8034.2
13	\$12976	\$82.73	\$13058.73
14	\$11414	\$0	\$11414
15	\$7745.4	\$0	\$7745.4
16	\$9294.2	\$0	\$9294.2
17	\$11644	\$82.73	\$11726.73
18	\$9708.9	\$0	\$9708.9
19	\$7951.4	\$95.36	\$8046.76
20	\$20600	\$0	\$20600
21	\$14195	\$92.82	\$14287.82
22	\$9697.5	\$0	\$9697.5
23	\$9450.1	\$0	\$9450.1
24	\$9137.2	\$0	\$9137.2

### 6.3 隨機實驗

本節主要是利用隨機產生數據來驗證本研究方法之求解品質較黃紹維(2006)之方法佳。首先，研究中選取四種供應商數目( $S = 2, 4, 6, 8$ )與二種零售商數目( $R = 4, 8$ )，一共產生 8 種供應商與零售商之組合進行比較。

本研究之解經過 ProcOTM 之排程撫平程序(包括移出程序、成對互換程序與二對一互換程序)步驟之調整，能有效降低加班成本，並將加班成本加入目標函數值之後，運用目標函數值計算適應力函數值，因此此時選擇要進行交配的染色體都是真正最佳解，在經過遺傳演算機制(包括選擇、交配與突變等動作)所得之解設為  $TC_{\text{本研究}}$ ，與將黃紹維(2006)在最小成本下之間

距乘數值  $K$  代入 ProcOTM 啟發式方法中求得之加班成本加入目標函數值得到之實際總成本  $TC_{Huang}$  相比較，其中，令平均總成本差異度為

$$I = \frac{TC_{Huang} - TC_{本研究}}{TC_{本研究}} \cdot 100\%$$

比較結果如下表所示，發現本研究方法所得之平均總成本明顯較黃紹維 (2006) 方法得到品質優良的解。且發現當零售賣場的數量固定時，隨著供應商數目越大時，成本差異度也有相對越大之趨勢。藉由如此較為貼切實務之考量，可提供零售賣場經理人作相關決策之依據。

S	R	代數	TC		I	runtime
2	4	500	Huang	1682.32	0.33%	9974.8 秒
			本研究	1676.87		
4			Huang	2689.433	7.82%	22303.7 秒
			本研究	2494.28		
6			Huang	4029	13.79%	40106.3 秒
			本研究	3540.62		
8			Huang	4818.927	21.01%	83974.3 秒
			本研究	3982.3		
S	R	代數	TC		I	runtime
2	8	500	Huang	3442.13	10.80%	142504.1sec
			本研究	3106.6		
4			Huang	4875.7	17.99%	158856sec
			本研究	4132.3		
6			Huang	7307	21.68%	165600.7sec
			本研究	6005.1		
8			Huang	7874	21.03%	176466.3sec
			本研究	6506		

## 第七章 結論與未來研究方向

### 7.1 結論與建議

本研究是在一連鎖零售體系下，建構系統總成本函數(包括供應商端平均成本、物流中心平均成本及零售賣場端之平均成本)，運用遺傳演算法，找出最佳基本週期與補貨排程。並考量物流中心因為有產能限制所產生之超時作業成本，利用排程撫平程序得到一成本最小化之加班工時成本，將超時成本加入目標函數值當中，利用目標函數值計算適應力函數值，並根據適應力函數值進行遺傳演算法之選擇、交配、突變等運算機制，以求得一組總成本最小的解。

並將本研究之模式與黃紹維(2006)做比較，利用其模式求得之間距乘數代入啟發式演算法當中，得到一超時成本，將超時作業成本加入至目標函數值中得到之總成本與本研究進行比較，發現本研究之方法明顯有較佳之求解品質。藉由如此較為貼切實務之考量，可提供零售賣場經理人作相關決策之依據。

另外，在獲得最佳的發車間距之後，決策者將可以利用此車輛排程要求第三方物流公司在時間點發車進行貨物運輸，而決策者可以根據排程資訊與第三方物流進行價格協商，可要求第三方物流以長期契約形式降低車輛租賃價格。此方式不僅是可以為企業有效降低營運成本，而也可以讓第三方物流公司可以根據本身的能力進行運輸規劃。

### 7.2 未來研究方向

本研究並未考量不同的貨物種類有不同的持有成本，僅考量不同供應商提供不同的貨物種類。由於單位庫存成本與產品之價值有關，因此發車間距排程亦需根據產品之價值作調整；另外，本研究所考量的運輸車輛為直線行駛，即中途無上貨或卸貨之情形，未來可考慮合併運輸方式，使運輸車輛的承載率可以大幅提高，將可更契合實務界之運作。

## 參考文獻

- [1] 黃惠民、黃正淡，「供應鏈管理策略聯盟透過 LLP 達成一汽車產業鏈之個案研究」，私立中原大學工業工程研究所，科技與管理學術研討會論文集，2001。
- [2] 黃建雄，「損耗性經記批量排程問題之研究」，碩士論文，私立東海大學工業工程與經營資訊研究所，2002。
- [3] 丁慶榮、翁偉倫，「接駁式轉運中心車輛排程之研究」，工業工程學刊，20 卷，6 期，636-650 頁，2003。
- [4] 張智星，MATLAB 程式設計與應用，清蔚科技，新竹市，2000。
- [5] 林晏妃，「配銷倉儲具空間限制下批量排程問題之研究」，碩士論文，私立東海大學工業工程與經營資訊研究所，2004。
- [6] 陳軍凱，「隨機需求下航空貨物收貨人員人力供給暨班次表設計規劃之研究」，國立中央大學，碩士論文，2004。
- [7] 黃紹維，「運用遺傳演算法求解接駁式轉運系統最佳車輛排程之研究」，碩士論文，私立東海大學工業工程與經營資訊研究所，2006。
- [8] 董福慶、陳明德編撰，物流中心揀貨作業，經濟部商業司1996。
- [9] 蔣明晃，「美國企業物流委外能夠盛行的因素」，取自物流新世紀，pp. 151-157，2001。
- [10] 周宣光，「物流資訊系統概論」，物流經營管理實務，經濟部商業司，1996。
- [11] Bard, J.F., "Weekly scheduling in the service industry: An application to mail processing and distribution centers", *IIE Transactions*, 37(5), 379-396, 2005.
- [12] Bartholdi, J.J., "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering", *Operations Research*, 29, 501-510, 1981.
- [13] Baumol, W.J., Vinod, H.D., "An inventory theoretic model of freight transport demand", *Management Science*, 16(7), 413-421, 1970.
- [14] Beasley, J.E., Cao, B., "A tree search algorithm for the crew scheduling problem", *European Journal of Operational Research*, 94(3), 517-526, 1996.

- [15]Blumenfeld, D.E., Burns, L.D., Diltz, J.D., Daganzo,C.F., “Analyzing trade-offs between transportation,inventory and production costs on freight networks”, *Transportation Research*, 19B(5), 361-380, 1985.
- [16]Boctor, F.F., “The g-group heuristic for single machine lot scheduling,” *International Journal of Production Research*, 25(3), 363-379, 1987.
- [17]Lee, C.Y., Jaruphongsa, W., Cetinkaya, S., “A dynamic model for inventory lot sizing and outbound shipment scheduling at a third-party warehouse”, *Operations Research*, 51(5), 2003.
- [18]Frazella, E.H., “World-Class Warehousing and Material Handling”, 2002.
- [19]Graham, R.L., “Bounds on multiprocessing timing anomalies,” *SIAM Journal on Applied Mathematics*, 17, 416-429, 1969.
- [20]Geng, P.C., Vickson, R.G., “Two heuristics for the economic lot scheduling problem: an experimental study,” *Naval Research Logistics*, 35, 605-617, 1988.
- [21]Higginson, J., “Modeling shipper costs in physical distribution analysis”, *Transportation Research*, 27(2), 113-124, 1993.
- [22]Hwang, H., Bake, W., Lee, M.K., “Clustering algorithms for order picking in an automated storage and retrieval systems,” *International Journal of Production Research*, 26(2), 189-201, 1988.
- [23]Jansson, J.O., “A simple bus line model for optimization of service frequency and bus size”, *Journal of Transportation Economics and Policy*, 14(1), 53-80, 1980.
- [24]Bartholdi J.J., Gue K.R., “Reducing labor costs in an LTL crossdocking terminal”, *Operations Research*, 48(6), 823–832, 2000.
- [25]Potts, J.C., Giddens, T.D., Yadav, S.B., “The development and evaluation of an improved genetic algorithm based on migration and artificial selection,” *IEEE Transactions on System, Man and Cybernetics*, 24(1), 73-86, 1994.
- [26]Lin, C.H., “Procedure of determining the order picking strategies in distribution center”, *International Journal of Production Economics*, 60, 301-307, 1999.
- [27]Marzouk, M., Moselhi, O., “Constraint-based genetic algorithm for earthmoving fleet selection,” *Canadian Journal of Civil Engineering*, 30, 673–683, 2003.

- [28] Watanabe, M., Furukawa, M., Mizoe, A., Watanabe, T., "GA applications to physical distribution scheduling problem", *IEEE Transactions on Industrial Electronics*, 48(4), 2001.
- [29] Yao, M.J., "The peak load minimization problem in cyclic production," *Computers and Operations Research*, 28, 1441-1460, 2001.
- [30] Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*. (Englewood Cliffs, NJ: Prentice-Hall), 1993.
- [31] Yao, M.J., Huang, J.X., "Solving the economic lot scheduling problem with deteriorating items using genetic algorithms", *Journal of Food Engineering*, 70, 309-322, 2005.

## 附錄 A 車輛選擇步驟

茲以表 4.3 說明如何依貨車載重量及貨車費用表進行車輛樣式與數量之選擇。表 4.3 中共有五種車輛，載重量及相對貨車費用分別為 300 公斤(\$800)、800 公斤(\$1600)、2000 公斤(\$3000)、5000 公斤(\$4000)及 6000 公斤(\$4500)。今假設求得供應商 1 的載重量  $k_1^a BQ_1^a = 7000\text{kg}$ ，其車輛選擇步驟如下：

步驟 1：判定目前的載重量(loading)超過最大載重車種(6000 公斤)，因此至少派遣一輛載重 6000kg 的車輛，此時剩下的載重量(left-loading)為  $7000 - 6000 \times 1 = 1000\text{kg}$ ，對剩餘的載重量進行車輛的選取。

步驟 2：找出是否可以只用一台超過貨物量但是最接近的貨車類型，因為目前載重量為 1000kg，找到的貨車類型是載重 2000kg 的車種，需要的車輛數(A)為 1 台。

步驟 3：當派遣 1 台載重 2000kg 之貨車，其載重量(loading)為  $1000 - 2000 \times 1 = -1000$ ，租車費用(price)為  $0 + 3000 \times 1 = \$3000$ 。一開始將此設為最佳成本，此時租車之最佳成本(best\_price)為 \$3000。接著令  $A=A-1$ ，(即選擇 0 台 2000kg)，因為  $A=0$ ，表示此種車輛已全部派遣完畢，將產品交由下一種車輛類行(800kg 之貨車)載運。

步驟 4：若將產品全部交由 800kg 之貨車載運，至少需要派遣車輛數(A)為 2 台，此時租車費用(price)為  $1600 \times 2 = \$3200$ ，因為成本並未較上一種方法佳，令  $A=A-1$ (僅有一台 800kg 之貨車)。當派遣一台 800kg 之貨車，因為產品總重量為 1000kg，此時將剩下之產品交由下一種貨車類型(載重 300kg 之貨車)進行載運，此時車輛費用為  $1600 \times 1 + 800 \times 1 = \$2400$ ，與之前所求得解進行比較，成本較低，故取其為目前之最佳解。

步驟 5：不斷重複步驟 3 及步驟 4，直到全部產品交由 300kg 之貨車載重為止，求得之解如下表 A.1 所示，

本例之總載重量為 7000kg，因為最大載重車種為 6000kg，故先將 6000kg 交由最大車種載運，而剩下之 1000kg 再根據上述之車輛選擇步驟可知，當選擇 1 輛 800kg 之貨車與 1 輛 300kg 之貨車時，此時車輛費用最低(\$2400)。故當載重量為 7000kg 時，租車費用為  $1600 \times 1 + 800 \times 1 + 4500 \times 1 = \$6900$

表 A.1 載重量 1000kg 之車輛選擇

車種 選擇	五 (6000kg)	四 (5000kg)	三 (2000kg)	二 (800kg)	一 (300kg)	載重	租車費用
1	0	0	1	0	0	2000	\$3000
2	0	0	0	2	0	1600	\$3200
3	0	0	0	1	1	1100	\$2400
4	0	0	0	0	4	1200	\$3200

## 附錄 B 物流中心之平均存貨成本

供應商 1 進站間距為 2 天，零售賣場 4 離站間距為 8 天，而零售賣場針對供應商 1 的單位需求貨物量為 250 公斤/天， $B=1$  天，因此其平均存貨水準為 375（公斤/天），平均存貨成本為 225 元/天，計算方式如下：

$$\begin{aligned}
 & \frac{2 \cdot \left[ 1 \cdot 2 \cdot 1 \cdot 250 + 2 \cdot 2 \cdot 1 \cdot 250 + \dots + \left( \frac{8}{2} - 1 \right) \cdot 2 \cdot 1 \cdot 250 \right] + \left( \frac{8}{2} \right) \cdot 2 \cdot 1 \cdot 250 - 8 \cdot 1 \cdot 250}{8} \\
 &= \frac{2 \cdot 1 \cdot 250 \left[ 1 \cdot 2 + 2 \cdot 2 + \dots + \left( \frac{8}{2} - 1 \right) \cdot 2 \right]}{8} \\
 &= \frac{2 \cdot 2 \cdot 1 \cdot 250 \left[ 1 + 2 + \dots + \left( \frac{8}{2} - 1 \right) \right]}{8} \\
 &= \frac{2 \cdot 2 \cdot 1 \cdot 250}{8} \left[ \frac{\left( 1 + \left( \frac{8}{2} - 1 \right) \right)}{2} \cdot \left( \frac{8}{2} - 1 \right) \right] \\
 &= \frac{2 \cdot 2 \cdot 1 \cdot 250}{8} \cdot \frac{4}{2} \cdot (4 - 1) \\
 &= \frac{1}{2} \cdot (4 - 1) \cdot 2 \cdot 1 \cdot 250 \\
 &= 750
 \end{aligned}$$

求得物流中心平均存貨水準為 750 公斤/天，在基本週期  $B=1$  天與庫存成本  $h^r = 0.6$  元/公斤-天假設下，供應商 1 對零售賣場 3 的平均庫存成本為  $750 \cdot 1 \cdot 0.6 = 450$  元/天。

## 附錄 C 成本函數簡化

$$\text{令}(k^a, \dots, k_n^a, k_1^d, \dots, k_m^d) = K$$

$$\begin{aligned} TC(K, B) &= \sum_{i=1}^n \frac{P_x}{k_i^a B} + h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n \left( \frac{A}{k_i^a B} + XQ_i^a \right) + \frac{1}{2} Bh^r \sum_{i=1}^n k_i^a \sum_{j=1}^m \left( \frac{k_j^d}{k_i^a} - 1 \right) q_{ij} \\ &\quad + \sum_{j=1}^m \frac{P_x}{k_j^d B} + h^s \sum_{j=1}^m V_j^d Q_j^d + h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d \end{aligned}$$

$$\begin{aligned} TC(K, B) &= \sum_{i=1}^n \frac{P_x}{k_i^a B} + h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n \frac{A}{k_i^a B} + \sum_{i=1}^n XQ_i^a \\ &\quad + \sum_{j=1}^m \frac{P_x}{k_j^d B} + h^s \sum_{j=1}^m V_j^d Q_j^d + h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d + \frac{1}{2} h^r B \left( \sum_{i=1}^n k_i^a \sum_{j=1}^m \frac{k_j^d}{k_i^a} q_{ij} - \sum_{i=1}^n k_i^a \sum_{j=1}^m q_{ij} \right) \end{aligned}$$

$$\begin{aligned} TC(K, B) &= \sum_{i=1}^n \frac{P_x}{k_i^a B} + h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n \frac{A}{k_i^a B} + \sum_{i=1}^n XQ_i^a + \sum_{j=1}^m \frac{P_x}{k_j^d B} + h^s \sum_{j=1}^m V_j^d Q_j^d \\ &\quad + h^r \sum_{j=1}^m \frac{k_j^d B}{2} Q_j^d + \frac{1}{2} h^r B \left( \sum_{i=1}^n \sum_{j=1}^m k_j^d q_{ij} - \sum_{i=1}^n k_i^a \sum_{j=1}^m q_{ij} \right) \end{aligned}$$

$$\begin{aligned} TC(K, B) &= \sum_{i=1}^n \left( \frac{P_x}{k_i^a B} + \frac{A}{k_i^a B} \right) + \sum_{j=1}^m \left( \frac{P_x}{k_j^d B} + \frac{k_j^d Bh^r Q_j^d}{2} \right) \\ &\quad + \frac{1}{2} h^r B \left( \sum_{i=1}^n \sum_{j=1}^m k_j^d q_{ij} - \sum_{i=1}^n k_i^a \sum_{j=1}^m q_{ij} \right) + h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n XQ_i^a + h^s \sum_{j=1}^m V_j^d Q_j^d \end{aligned}$$

$$\begin{aligned} TC(K, B) &= \sum_{i=1}^n \left[ \frac{P_x}{k_i^a B} + \frac{A}{k_i^a B} - \frac{h^r B k_i^a}{2} \sum_{j=1}^m q_{ij} \right] + \sum_{j=1}^m \left[ \frac{P_x}{k_j^d B} + \frac{k_j^d Bh^r Q_j^d}{2} + \frac{1}{2} h^r B k_j^d \sum_{i=1}^n q_{ij} \right] \\ &\quad + \left( h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n XQ_i^a + h^s \sum_{j=1}^m V_j^d Q_j^d \right) \end{aligned}$$

$$\text{令} h^s \sum_{i=1}^n V_i^a Q_i^a + \sum_{i=1}^n XQ_i^a + h^s \sum_{j=1}^m V_j^d Q_j^d = C$$

目標函數式：

$$TC(K, B) = \sum_{i=1}^n \left[ \frac{P_x}{k_i^a B} + \frac{A}{k_i^a B} - \frac{h^r B k_i^a}{2} \sum_{j=1}^m q_{ij} \right] + \sum_{j=1}^m \left[ \frac{P_x}{k_j^d B} + \frac{k_j^d Bh^r Q_j^d}{2} + \frac{1}{2} h^r B k_j^d \sum_{i=1}^n q_{ij} \right] + C$$

## 附錄 D ProcOTM 完整範例說明

### 步驟 1：設定起始參數

表 D.1 設定起始參數

	供應商 1	供應商 2	零售賣場 1	零售賣場 2
產品代號	1	2	3	4
間距乘數 K	4	2	4	16
需求時間 $\sigma_i$	306(分)	208(分)	381(分)	586(分)

共同週期  $K=16$  (取最小公倍數)

$B=480$

物流中心人員單位薪資率=3.2(元/分鐘)

$\text{chi}(\chi)$  累積連續無改善次數=0

$\text{phi}(\phi)$  不合理的解=0

### 步驟 2：藉由 Proc.IS 程序得到一組初始生產排程 W 並開始區域搜尋。

初始排程一開始，將分裝需求時間  $\sigma_i$  作降冪排列，此時  $\mathcal{g}=0$ ， $i \in N$  ( $N=4$ )，得到需求時間  $\sigma_i$  由大到小排序後的產品集合  $[I]=\{4,3,1,2\}$ 。因為判定  $[I] \neq \emptyset$ ，因此從  $[I]$  之取第一個值，即產品代號 4(零售賣場 2)，使用 Proc PS 程序將  $\sigma_4$  指派到  $W(\mathcal{g})$  排程中， $k(\mathcal{g} \cup \{i\}) = \text{lcm}\{k(\mathcal{g}), k_i\}$ ， $k_i = k_4 = 16$ ，因此  $k(4) = \text{lcm}\{16\} = 16$ ，基本週期數目為 16，生產排程如表 D.2。

表 D.2 部分初始生產排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1				586	586
基期 2				0	0
基期 3				0	0
基期 4				0	0
基期 5				0	0
基期 6				0	0
基期 7				0	0
基期 8				0	0
基期 9				0	0
基期 10				0	0
基期 11				0	0
基期 12				0	0

基期 13				0	0
基期 14				0	0
基期 15				0	0
基期 16				0	0

更新  $g = g \cup \{i\}$  及  $[I] = [I] - \{i\}$ ，得  $[I] = \{4, 3, 1, 2\} - \{4\} = \{3, 1, 2\}$ ， $[I] \neq 0$ ，因此從  $[I]$  之取第一個值，即產品 3(零售賣場 1)，使用 Proc PS 程序將  $\sigma_3$  指派到  $W(g)$  排程中， $k(3 \cup \{4\}) = lcm\{4, 16\} = 16$ ，得基本週期數目為 16，物流中心目前總需求時間第一個最小的基期是為基期 2，因此將  $\sigma_3 = 381$  放入到基期 2，其餘按照分裝作業時間依序排序產品 1 及產品 2，得到起始排程如下表 D.3：

表 D.3 利用 ProcIS 的初始排程

產品基期別	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0	0	586	586
基期 2	0	208	381	0	589
基期 3	306	0	0	0	306
基期 4	0	208	0	0	208
基期 5	0	0	0	0	0
基期 6	0	208	381	0	589
基期 7	306	0	0	0	306
基期 8	0	208	0	0	208
基期 9	0	0	0	0	0
基期 10	0	208	381	0	589
基期 11	306	0	0	0	306
基期 12	0	208	0	0	208
基期 13	0	0	0	0	0
基期 14	0	208	381	0	589
基期 15	306	0	0	0	306
基期 16	0	208	0	0	208

### 步驟 3：應用 Proc.SS 程序改善 $W^m$ 的 $L(W^m)$

Proc SS 程序主要是經由移出程序、成對互換程序和 2 對 1 互換程序 3 個子程序來完成，而其 3 個子程序之詳細演算步驟程序分別說明如下：

#### (一) 移出程序 (Smooth-out Routine)：

移出程序主要的目地在從擁有最大工作負載的基本生產週期中，將某個

$k_i > 1$  的產品之  $\sigma_i$  指派到其他基本生產週期。

步驟1：將所有屬於  $\Gamma(\tau, W^m)$  的產品  $i$  之  $\sigma_i$  做降冪排列且標示為未受試驗的產品。

步驟2：檢驗是否有任何在  $\Gamma(\tau, W^m)$  中的產品是未受試驗的

1.1 判斷  $\Gamma(\tau, W^m)$  是否有未受試驗的產品，是則找出第一個未受試驗的產品  $i$ ，且將產品  $i$  使用 Proc PS 重新排程以獲得一個較小的物流中心負載  $\Omega(W)$  和其對應之排程  $W$ 。更新  $\Gamma(\tau, W^m) = \Gamma(\tau, W^m) - \{i\}$ 。進入步驟3。

1.1 判斷  $\Gamma(\tau, W^m) = \phi$ ，是表示  $\Gamma(\tau, W^m)$  所有產品均已受過試驗，停止 Smooth-out Routine 演算程序。

步驟3：檢驗  $\Omega(W^m)$  是否已經改善：

1.1 判斷  $\Omega(W) < \Omega(W^m)$ ，是則令  $\Omega(W) = \Omega(W^m)$ 、 $W^m = W$  和  $l = 1$ ，並停止 Smooth-out Routine 演算程序。

1.1 判斷  $\Omega(W) \geq \Omega(W^m)$ ，是則標記產品  $i$  為已受試驗的，並回到步驟2。

(二) 成對互換程序 (Pair-exchange Routine)：

成對互換程序嘗試降低  $lcm\{k_i\}$  個基本作業週期中的最大物流中心工時負載，從擁有最大產品負載的基本生產週期中，將某個  $k_i > 1$  的產品之作業排程與非最大物流中心負載的基本作業週期中的某個  $k_i > 1$  的產品作業排程互換。成對互換程序的擬似程式碼敘述如下：

步驟1：將  $i \in \Gamma(\tau, W^m)$  中  $\sigma_i$  的做降冪排列，且  $j \in \Lambda(\tau, W^m)$  中的  $\sigma_j$  做升冪排列。並標記  $i \in \Gamma(\tau, W^m)$  和  $j \in \Lambda(\tau, W^m)$  中的產品為未受試驗的。

步驟2：檢驗是否有任何在  $\Gamma(\tau, W^m)$  中的產品是為未受試驗的。

2.2 判斷  $\Gamma(\tau, W^m)$  是否有未受試驗的產品，是則找出第一個未受試驗的產品  $i$ ，並進入步驟3。

3.3 判斷  $\Gamma(\tau, W^m) = \phi$ ，是則表示  $\Gamma(\tau, W^m)$  所有產品均已受過試驗，停止 Exchange Routine 演算程序。

步驟3：檢驗是否有任何在  $\Lambda(\tau, W^m)$  中的產品是為未受試驗的。

3.1 判斷  $\Lambda(\tau, W^m)$  是否有未受試驗的產品，是則找出第一個未受試驗的產品  $j$ ，進入步驟4。

3.2 判斷  $\Lambda(\tau, W^m)$  是否均受過試驗，是則將  $\Lambda(\tau, W^m)$  集合中的產品重新標記為未受試驗的且將產品  $i$  標記為已受試驗的，回到步驟2。

步驟4：比較  $\sigma_i$  和  $\sigma_j$ 。

4.1 判斷  $\sigma_i > \sigma_j$ ，是則進入步驟4.1.1。

1.1.1 將產品  $i$  從  $W^m$  排程中移出，並指派產品  $j$  的排程使其在標記  $\tau$  的生產週期上生產。

1.1.2 將產品  $i$  使用 Proc PS 重新排程以獲得一個較小的物流中心負載  $L(W)$  和其對應之排程  $W$ 。

1.1.3 檢驗  $\Omega(W^m)$  否已經改善：

A. 判斷  $\Omega(W) < \Omega(W^m)$ ，是則令  $\Omega(W) = \Omega(W^m)$ 、 $W = W^m$  和  $l = 1$ ，並停止 Pair-exchange Routine 演算程序。

B. 判斷  $\Omega(W) \geq \Omega(W^m)$ ，是則標記產品  $j$  為已受試驗的，並回到步驟3。

4.2 判斷  $\sigma_i < \sigma_j$ ，是則回到步驟3。

(三) 2 對1 互換程序 (Two-One Routine)

2 對1 互換程序嘗試降低  $lcm\{k_i\}$  個基本作業週期中的最大物流中心工時負載，從擁有最大產品負載的基本作業週期中，將某個  $k_i > 1$  的產品之作業排程與非最大產品負載的基本作業週期中的2 個  $k_i > 1$  的產品作業排程互換。2 對1 互換程序的擬似程式碼敘述如下：

步驟1：將  $i \in \Gamma(\tau, W^m)$  中的  $\sigma_i$  做降冪排列，且  $j, l \in \Lambda(\tau, W^m)$  中的  $\sigma_i + \sigma_j$  做升冪排列。並標記  $i \in \Gamma(\tau, W^m)$  和  $j, l \in \Lambda(\tau, W^m)$  中的產品為未受試驗的。

步驟2：檢驗是否有任何在  $\Gamma(\tau, W^m)$  中的產品為未受試驗的。

2.1 判斷  $\Gamma(\tau, W^m)$  是否有未受試驗的產品，是則找出第一個未受試驗的產品  $i$ ，並進入步驟3。

2.2 判斷  $\Gamma(\tau, W^m) = \phi$ ，是則表示  $\Gamma(\tau, W^m)$  所有產品均已受過試驗，停止 Two-One Routine 演算程序。

步驟3：檢驗是否有成對產品  $j$  和  $l$  在  $\Lambda(\tau, W^m)$  中是為未受試驗的。

3.1 判斷  $\Lambda(\tau, W^m)$  是否有未受試驗的成對產品  $j$  和  $l$ ，是則找出第一個未受試驗的成對產品  $j$  和  $l$ ，進入步驟4。

3.2 判斷  $\Lambda(\tau, W^m)$  是否均受過試驗，是則將  $\Lambda(\tau, W^m)$  集合中的產品重新標記為未受試驗的且將產品  $i$  標記為已受試驗的，回到步驟2。

步驟4：比較  $\sigma_i$  和  $\sigma_j + \sigma_j$ 。

4.1 判斷  $\sigma_i > \sigma_j + \sigma_j$ ，是則進入步驟4.1.1。

4.1.1 將產品  $i$  從  $W^m$  排程中移出，並指派成對產品  $j$  和  $l$  的排程使其在標記  $\tau$  的作業週期上生產。

4.1.2 將產品  $i$  使用 Proc PS 重新排程以獲得一個較小的物流中心負載  $L(W)$  和其對應之排程  $W$ 。

4.1.3 檢驗  $\Omega(W^m)$  否已經改善：

A. 判斷  $\Omega(W) < \Omega(W^m)$ ，是則令  $\Omega(W) = \Omega(W^m)$ 、 $W = W^m$  和  $l=1$ ，並停止 Two-One Routine 演算程序。

B. 判斷  $\Omega(W) \geq \Omega(W^m)$ ，是則標記成對產品  $j$  和  $l$  為已受試驗的，並回到步驟3。

4.2 判斷  $\sigma_i < \sigma_j + \sigma_l$ ，是則回到步驟3。

### 3.1 使用移出程序來改善

令基本週期  $B=480$ ，調整前最大的總需求時間  $L(W^m) = 589$ （即基期2）大於  $B$ ，所以該排程為需要加班工時，改善前的排程如上表 D.3 所示。將位於物流中心最大工作負載需求即基期2上裝卸且間距乘數  $K > 1$  的產品集合，按其產品需求時間  $\sigma_i$  降冪排序後，得  $\Gamma(\tau, W^m) = \{3, 2\}$ 。產品3優先從基期2移出，如表 D.4，再依 Proc.PS 重新排程，排列後的生產排程如表 D.5 所示。

表 D.4 移出程序之移出排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0		586	586
基期 2	0	208		0	208
基期 3	306	0		0	306
基期 4	0	208		0	208
基期 5	0	0		0	0
基期 6	0	208		0	208
基期 7	306	0		0	306
基期 8	0	208		0	208
基期 9	0	0		0	0
基期 10	0	208		0	208
基期 11	306	0		0	306
基期 12	0	208		0	208
基期 13	0	0		0	0
基期 14	0	208		0	208
基期 15	306	0		0	306
基期 16	0	208		0	208

表 D.5 移出程序排列後的排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0	381	586	967
基期 2	0	208	0	0	208
基期 3	306	0	0	0	306
基期 4	0	208	0	0	208
基期 5	0	0	381	0	381
基期 6	0	208	0	0	208
基期 7	306	0	0	0	306
基期 8	0	208	0	0	208
基期 9	0	0	381	0	381
基期 10	0	208	0	0	208
基期 11	306	0	0	0	306
基期 12	0	208	0	586	208
基期 13	0	0	381	0	381
基期 14	0	208	0	0	208
基期 15	306	0	0	0	306
基期 16	0	208	0	0	208

由表 D.5 改善的排程，最大的工時負載  $L(W^m) = 967$  (基期 1) 超出基本週期  $B = 480$ ，且解未改善(改善前的解為 589)，故恢復為表 D.3 的作業排程，進入成對互換程序。

### 3.2 使用成對互換來改善：

將位於物流中心中最大工作負載需求時間(即基期 2)上裝卸且間距乘數  $K > 1$  的產品集合，按其產品需求時間  $\sigma_i$  降冪排序後，得  $\Gamma(\tau, W^m) = \{3, 2\}$ ，而未在基期 2 上裝卸且間距乘數  $K > 1$  的產品集合，按其需求時間  $\sigma_i$  升冪排列後，得  $\Lambda(\tau, W^m) = \{1, 4\}$ ，並判斷移出產品的需求時間( $\sigma_3 = 381$ )是否大於將移入基期 2 的產品 1 的需求時間( $\sigma_1 = 306$ )，如「是」，先將產品 1 移入至基期 2 進行分裝作業如表 D.6，而產品 3 則由基期 2 移出後再依 Proc.PS 重新排程，經過成對互換程序排列後的生產排程如表 D.7 所示。

表 D.6 成對互換移入排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0		586	586
基期 2	306	208		0	514
基期 3	0	0		0	0
基期 4	0	208		0	208
基期 5	0	0		0	0
基期 6	306	208		0	514
基期 7	0	0		0	0
基期 8	0	208		0	208
基期 9	0	0		0	0
基期 10	306	208		0	514
基期 11	0	0		0	0
基期 12	0	208		0	208
基期 13	0	0		0	0
基期 14	306	208		0	514
基期 15	0	0		0	0
基期 16	0	208		0	208

表 D.7 成對互換排列後的排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0	0	586	586
基期 2	306	208	0	0	514
基期 3	0	0	381	0	381
基期 4	0	208	0	0	208
基期 5	0	0	0	0	0
基期 6	306	208	0	0	514
基期 7	0	0	381	0	381
基期 8	0	208	0	0	208
基期 9	0	0	0	0	0
基期 10	306	208	0	0	514
基期 11	0	0	381	0	381
基期 12	0	208	0	0	208
基期 13	0	0	0	0	0
基期 14	306	208	0	0	514
基期 15	0	0	381	0	381
基期 16	0	208	0	0	208

由表 D.7 改善的分裝作業排程，最大的工時負載  $L(W^m) = 586$  (基期 1) 雖然仍超出基本週期  $B = 480$ ，但解有改善(改善前的解為 589)， $\text{improved} = 589 - 586 = 3$ ，此時判斷  $L(W^m) > B$  且  $\text{improve} > 0$ ，重複進行上述移出程序及成對互換程序，直到在成對互換程序中  $L(W^m) > B$  且  $\text{improve} = 0$ ，進入二對一互換程序，目前作業排程如下表 D.8 所示：

表 D.8 欲進行二對一的初始作業排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0	0	586	586
基期 2	306	208	0	0	514
基期 3	0	0	381	0	381
基期 4	0	208	0	0	208
基期 5	0	0	0	0	0
基期 6	306	208	0	0	514
基期 7	0	0	381	0	381
基期 8	0	208	0	0	208
基期 9	0	0	0	0	0
基期 10	306	208	0	0	514

基期 11	0	0	381	0	381
基期 12	0	208	0	0	208
基期 13	0	0	0	0	0
基期 14	306	208	0	0	514
基期 15	0	0	381	0	381
基期 16	0	208	0	0	208

### 3.3 二對一互換(two to one exchange routine)

將位於物流中心最大工作負載需求時間（即基期 1）上裝卸且間距乘數  $K > 1$  的產品集合，按其產品需求時間  $\sigma_i$  降冪排序後，得  $\Gamma(\tau, W^m) = \{4\}$ ，而未在基期 1 上裝卸且間距乘數  $K > 1$  所形成的產品成對集合中，按成對產品需求總時間  $\sigma_i$  升冪排序後，得  $\Lambda(\tau, W^m) = \{2, 1, 3\}$ ，並判斷移出產品的需求時間 ( $\sigma_4 = 586$ ) 是否大於將移入基期 1 的成對產品 2 和產品 1 的總需求時間 ( $\sigma_1 + \sigma_2 = 306 + 208 = 514$ )，如「是」，則將產品 1 及產品 2 移入至基期 1 進行裝卸等動作，如表 D.9，而產品 4 則由基期 1 移出後再依 Proc.PS 重新作業排程，經過二對一互換程序排列後的分裝作業排程如表 D.10 所示。

表 D.9 二對一移入的排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	306	208	0		514
基期 2	0	0	0		0
基期 3	0	208	381		589
基期 4	0	0	0		0
基期 5	306	208	0		514
基期 6	0	0	0		0
基期 7	0	208	381		589
基期 8	0	0	0		0
基期 9	306	208	0		514
基期 10	0	0	0		0
基期 11	0	208	381		589
基期 12	0	0	0		0
基期 13	306	208	0		514
基期 14	0	0	0		0
基期 15	0	208	381		589
基期 16	0	0	0		0

表 D.10 二對一排列後的排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	306	208	0	0	514
基期 2	0	0	0	586	586
基期 3	0	208	381	0	589
基期 4	0	0	0	0	0
基期 5	306	208	0	0	514
基期 6	0	0	0	0	0
基期 7	0	208	381	0	589
基期 8	0	0	0	0	0
基期 9	306	208	0	0	514
基期 10	0	0	0	0	0
基期 11	0	208	381	0	589
基期 12	0	0	0	0	0
基期 13	306	208	0	0	514
基期 14	0	0	0	0	0
基期 15	0	208	381	0	589
基期 16	0	0	0	0	0

由表 D.10 改善的排程，最大的工時負載  $L(W^m) = 589$  (基期 3) 仍超出基本週期  $B=480$ ，且解並未改善(改善前的解為 586)，此時判斷  $L(W^m) > B$  且  $improve \leq 0$ ，累計無改善次數  $\chi = \chi + 1$ ，當尚未達到終止條件時，隨機選取  $[n/2]$  個產品重排。達到終止條件的最終排程如表 D.11 所示，由表中可以發現基期 1、基期 2、基期 6、基期 10 及基期 14 的作業時間超過一天基本工時 480 分鐘，針對這四個週期計算加班的人工成本，超過的基本工時\* 物流中心人員單位薪資率 =  $[(514-480)+(586-480)+(514-480)+(514-480)+(514-480)] \times 3.2 = 242 \times 3.2 = 774.4$

表 D.11 最終排程

	產品 1	產品 2	產品 3	產品 4	total
基期 1	0	0	0	586	586
基期 2	306	208	0	0	514
基期 3	0	0	381	0	381
基期 4	0	208	0	0	208
基期 5	0	0	0	0	0
基期 6	306	208	0	0	514
基期 7	0	0	381	0	381
基期 8	0	208	0	0	208
基期 9	0	0	0	0	0
基期 10	306	208	0	0	514
基期 11	0	0	381	0	381
基期 12	0	208	0	0	208
基期 13	0	0	0	0	0
基期 14	306	208	0	0	514
基期 15	0	0	381	0	381
基期 16	0	208	0	0	208