

國立交通大學
運輸科技與管理學系碩士班

碩士論文

生產一般產品與再生產品製造商利潤最大化經濟批量
排程問題之研究

The Economic Lot Scheduling Problem for a Profit-
maximizing Manufacturer Producing Regular and
Remanufactured Products

研究生：蔡季軒

指導教授：姚銘忠 教授

共同指導教授：林仁彥 助理教授

中華民國一百零一年七月

生產一般產品與再生產品製造商利潤最大化經濟批量排程問題
之研究

The Economic Lot Scheduling Problem for a Profit-maximizing
Manufacturer Producing Regular and Remanufactured Products

研究生：蔡季軒 Student：Chi-Hsuan Tsai

指導教授：姚銘忠、林仁彥 Advisor：Ming-Jong Yao, Jen-Yen Lin



國立交通大學
運輸科技與管理學系
碩士論文

A Thesis
Submitted to Department of Transportation Technology and Management
College of Management
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Transportation Technology and Management

June 2005

Hsinchu, Taiwan, Republic of China

中華民國一百零一年七月

生產一般產品與再生產品製造商利潤最大化經濟批量排程問題之研究

研究生：蔡季軒

指導教授：姚銘忠 教授、林仁彥 助理教授

國立交通大學運輸科技與管理學校碩士班

中文摘要

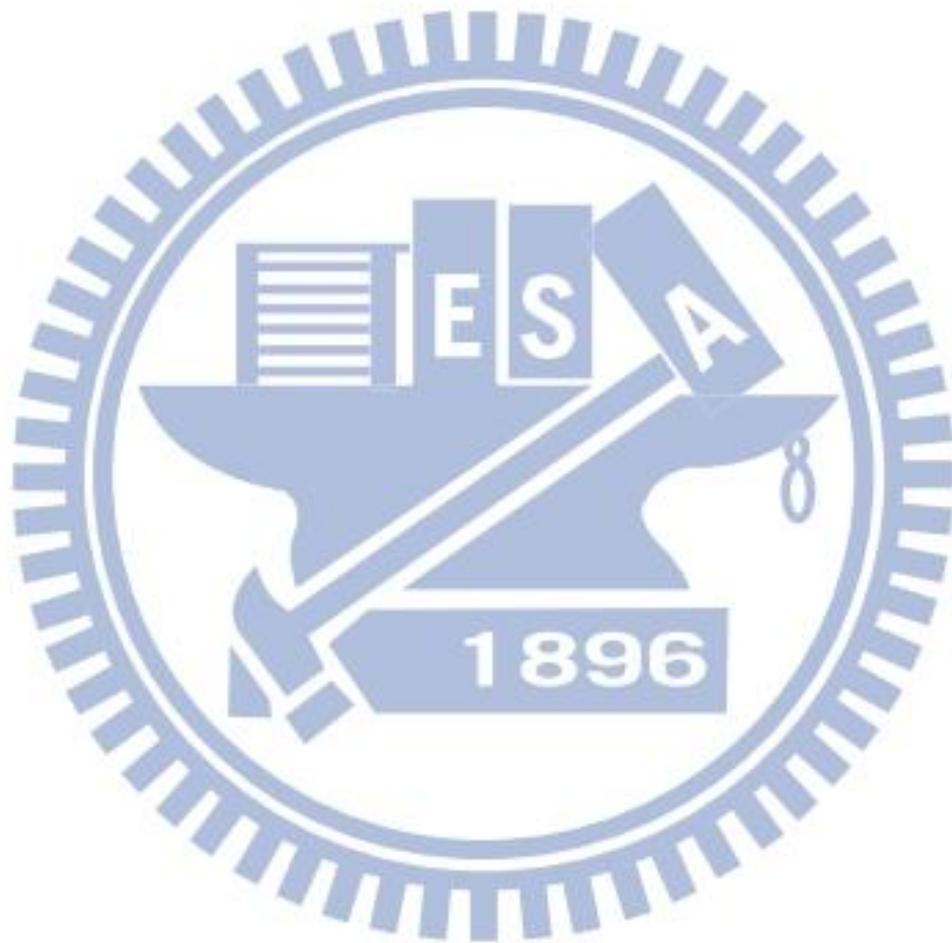
經濟批量排程問題(Economic Lot Scheduling Problem, ELSP)最早在 1958 年提出雛形後，至今近 50 年被學者持續鑽研及提出延伸研究議題，其目標在於決定出能使總成本最小化的生產批量大小和產品生產順序。而為了接近現實的生產情況，現今的學者又將 ELSP 細分成各種的情境來探討，其中一種是生產製造的同時，考量有逆物流並且回收再製造之行為。本研究探討不同於已有文獻研究之再製造情境，認為回收回來的原料製成之再生產品因為原料純度或新品有所耗損，造成再生之產品品質會劣於原產品(本研究稱為一般產品)，並且認定再生產品服務與一般產品不同的市場。

本研究使用時間變動批量大小方法(Time-varying lot-sizes approach, TVLS)架構出新的 ELSP 問題最佳解求解方法，分為三個部分：第一部分目的在於獲取基本週期及基本週期乘數，本研究建構 BP-based 模式並分別使用共同週期法、基因演算法，及接合點搜尋法三種方法進行求解之；第二部分使用最長操作時間派工法，將基本週期及基本週期乘數轉換為生產排程及生產總週期；第三部分在給定生產排程及生產總週期的情況下使用內點法(interior point algorithm)搜尋模式最佳解，改善文獻中使用投機演算法(quick-and-dirty algorithm)，本研究使用 MATLAB 程式進行內點法的搜尋。接著放鬆總週期，在給定排程下使用二分搜尋法搜尋最佳總週期，最後利用鄰域搜尋法進行改善排程，以求得近似最佳解。

在隨機數值分析中，本研究針對不同機台利用率進行求解，比較共同週期法、基因演算法，及接合點搜尋法三者的求解品質，其中以接合點搜尋法決定出之基本週期及基本週期乘數建構出的生產排程及生產總週期可以獲得

最好的目標值，而使用二分搜尋法搜尋最佳總週期的確能在有效率的時間內進一步找到更好之值，然而使用鄰域搜尋法進行改善排程的時間過長且改善的幅度較小。而運算時間的部分主要受排程長度的影響而非受限於求解基本週期乘數之方法，當排程長度越長，求解時間也越長。

關鍵字：經濟批量排程問題、再製造、共同週期法、基因演算法、接合點搜尋法、時間變動批量大小方法



The Economic Lot Scheduling Problem for a Profit-maximizing Manufacturer Producing Regular and Remanufactured Products

Student: Chi-Husan Tsai

Advisor: Ming-Jong Yao Jen-Yen Lin

Department of Transportation Technology and Management
National Chiao Tung University, Hsinchu, Taiwan, ROC

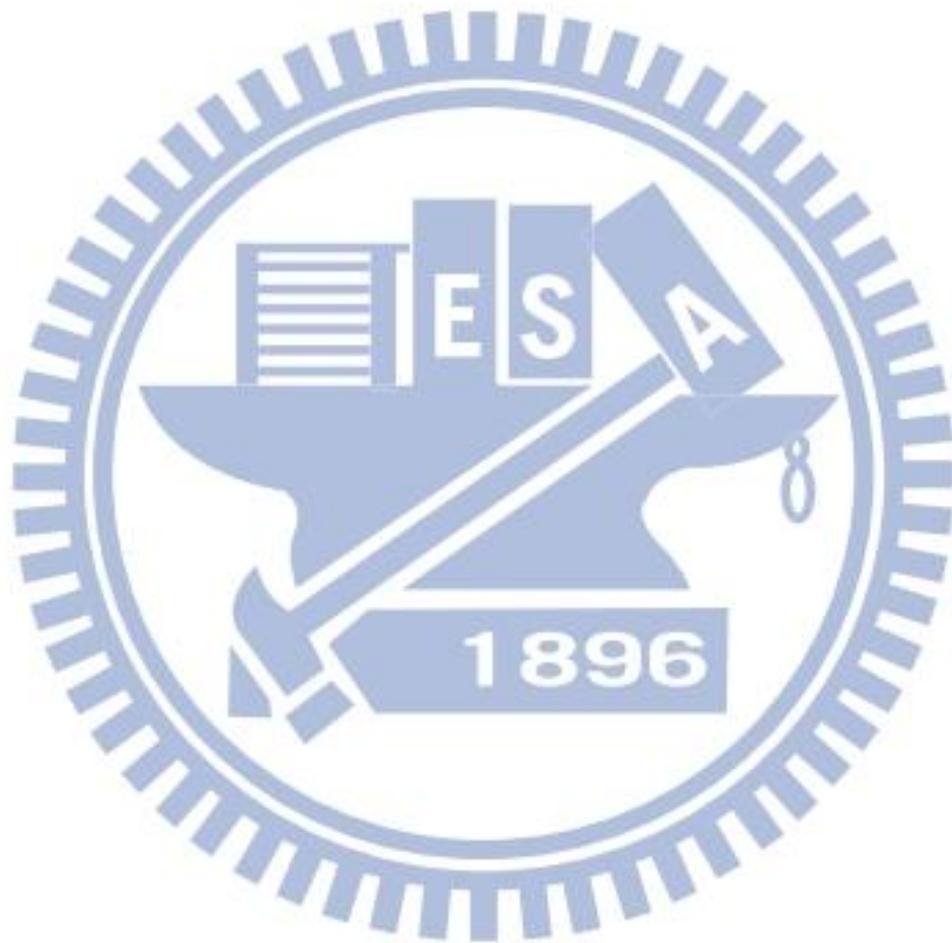
Abstract

Economic lot scheduling problem (ELSP) has been studied over 50 years. In this study, we are interested in the ELSP producing different types of products, namely, regular product and remanufactured product. In the concerned production system, we produce regular products to meet the customers' demand and also collect the recycled ones for remanufacturing. Because of using the recycled materials, the quality of the remanufactured products could be inferior to the regular ones. We assume that the remanufactured products are sold in different market segments.

To solve the concerned problem, we formulate a mathematical model using the time-varying lot-sizes (TVLS) approach. The TVLS approach assists our solution approach in generating feasible production schedules by allowing the lot sizes and cycle times for each product vary over time. We may divide the proposed solution approach into three phases: The first phase acquires the value of basic period and the set of multipliers. We propose three approaches, namely, a common cycle approach, a genetic algorithm, and a junction-point search algorithm for the purpose of the first phase. Taking the obtained basic period and the set of multipliers, the second phase generates a production sequence using a Longest-Processing-Time (LPT) heuristic. Given the production sequence and the value of cycle time, we solve the TVLS model, which is a nonlinear program, by an interior-point algorithm (that outperforms the quick-and-dirty heuristic in the literature) in the third phase. Different from the other approach based on the TVLS method, we employ a bisection search procedure to seek for the optimal cycle time, and also utilize a neighborhood search heuristic to look for a better production sequence to improve the solution quality.

Our numerical experiments compare three proposed solution approaches with different utilization rates. It shows that both the junction point search algorithm and the genetic algorithm outperform the common cycle approach. Also, the production sequence obtained from the junction point search algorithm performs better than that of the genetic algorithm. Furthermore, interestingly, the run time is majorly determined by the length of the production sequence rather than the number of items produced in the system.

Keywords: Economic Lot Scheduling Problem; Remanufacture; Common Cycle Approach; Genetic Algorithm; Junction Point Algorithm; Time Varying Lot Size Approach



致謝

兩年過得很快，終於是如願拿到有簽名的審查書了，過程中雖然充滿艱辛和挫折，但中間不只有老師的循循善誘，也有很多支持我的朋友們。

最先要感謝的當然是爸媽，謝謝你們把我拉拔到大，雖然很多時候小孩子的成長不如你們的期待，但終究是乖乖的拿了一個碩士學位回家。再來要感謝的是姚銘忠老師和林仁彥老師，兩位老師每次在論文的討論中，都能給我很多啟發，開拓我的視野，最後終於在困難中順利得完成了論文，真的是特別感謝兩位老師。另外也要謝謝兩位口試委員黃寬丞老師與陳文智老師，兩位老師在口試中的寶貴意見，讓這份論文能更臻完美。

還有在辛苦與煎熬之於陪伴我的朋友們，同一個 LAB 屋簷下的均宜、嘉珮和 Kiwy 三個陳氏姊妹花，兩年來一起成長一起學習，現在回想起來真能讓我掬起一瓢淚(?)；大學同學仍然感情不散的 David、金融、東翰、小雷、李卉、吉慧、菜瓜瓜、阿芝、阿昇、阿丹、森巴、兔寶、黑肥，運管 99 真的是感情長長久久；而珮慈、儀安和所有的碩二同學們，與大家一同修課、一同聚餐的日子真的是苦悶生活中的一大調劑，希望大家未來工作當兵都能順利；還有要感謝小朱、依潔、思賢和阿哲四位學長姐在我們碩一時的陪伴與學習相長，讓我們對碩士班的生活逐現熟悉，還能在忙碌閒暇之餘一起分享美食；而現在碩一的學弟妹瑜亮、怡今、朝援、小季和小恕，也是在日常生活中幫了我們很多忙，祝你們明年也會關關難過關關過。

還要謝謝撐起我另一個生活的社團朋友們，快可以當一家人的小 Q、超囧、Byes、賴 QQ、啊嘎、小俊和阿鈞，一起練舞的查理、土蛋、焦糖、派克、腿妹、瑞可、小冰和 OB，一起重訓的小石、大拜、Eric 和 Seed，一起出遊的小朱、雅喬、LoLo、垮垮、始棒、丹尼、小翔、會長、米腸，還有鮪魚、阿明、Fred、阿福、奇多、Daniel、阿樹哥、...，你們真的是我生活中很重要的一塊，謝謝大家的陪伴，希望大家都能平平安安，一直持續下去。

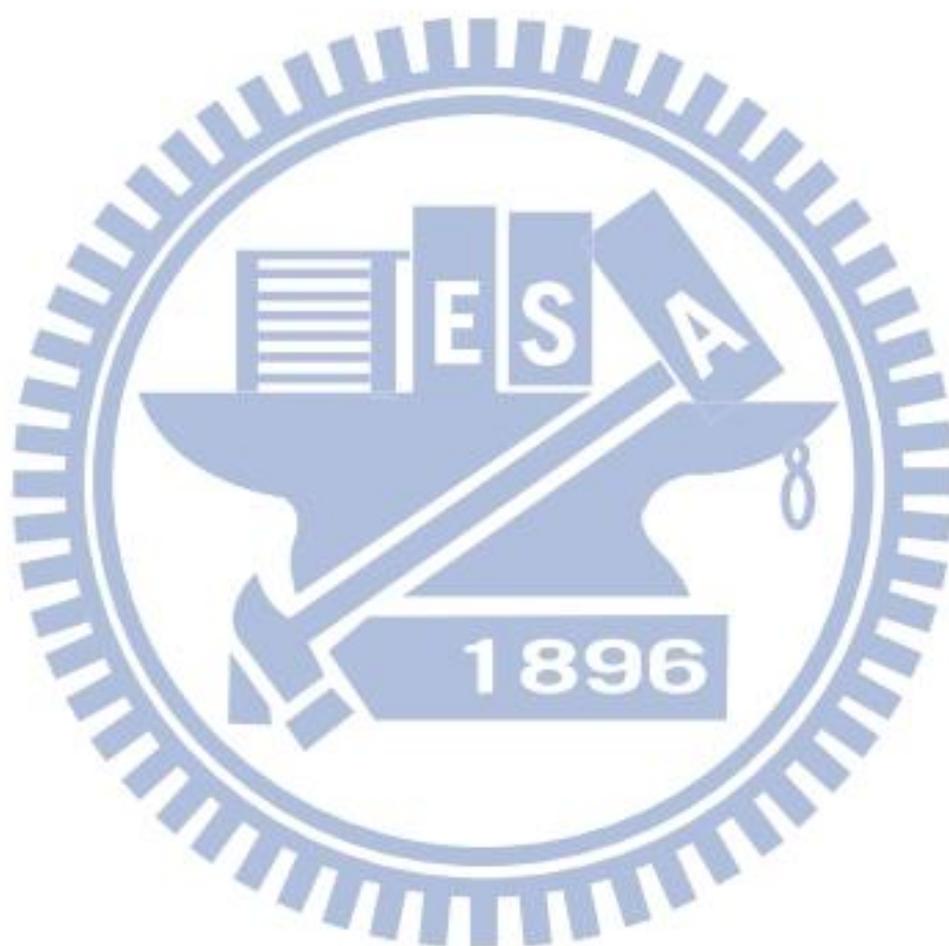
目錄

中文摘要.....	I
英文摘要.....	III
致謝.....	V
目錄.....	VI
表目錄.....	X
圖目錄.....	XI
第一章 緒論.....	1
1.1 研究動機與目的.....	1
1.2 研究範圍與假設.....	2
1.3 研究架構與流程.....	3
第二章 文獻回顧.....	5
2.1 經濟批量排程問題.....	5
2.1.1 獨立解法.....	8
2.1.2 共同週期法.....	8
2.1.3 基本週期法.....	9
2.1.4 延伸基本週期法.....	9
2.1.5 時間變動批量大小方法.....	11
2.2 再製造之生產存貨管理文獻.....	13
2.2.1 逆物流中的再製造.....	13
2.2.2 再製造生產存貨管理文獻.....	14
2.3 再製造經濟批量排程問題.....	14
2.4 小結.....	15
第三章 研究方法.....	16
3.1 情境描述與假設.....	16

3.2 TVLS 模式.....	18
3.2.1 TVLS 符號定義.....	18
3.2.2 TVLS 模式建立.....	19
3.2.2.1 回收時間與生產時間之關係.....	19
3.2.2.2 目標式說明.....	20
3.2.3 驗證本研究之 TVLS 目標式在給定排程週期 T 及排程 q 之情況下為 凹性函數.....	23
3.3 BP-based 模式.....	25
3.3.1 BP-based 模式符號定義.....	25
3.3.2 BP-based 模式建立.....	25
3.3.2.1 實際回收時間與生產時間之關係.....	25
3.3.2.2 縮放比例 θ_i	26
3.3.2.3 目標式說明.....	27
3.3.3 驗證本研究之 BP-based 目標式為凹性函數.....	29
第四章 模式求解.....	31
4.1 求解演算法架構介紹.....	31
4.1.1 縮放比例 θ_i 之設定.....	33
4.1.2 縮放比例之上下界及最佳縮放比例 θ_i^*	33
4.2 TVLS 最佳解搜尋法.....	35
4.2.1 建立排程.....	35
4.2.2 求解 TVLS 模式.....	36
4.2.2.1 排程固定下對 T 進行搜尋.....	36
4.2.2.2 排程改善.....	37
4.3 求解 BP-based 模式方法一：共同週期法.....	39
4.4 求解 BP-based 模式方法二：基因演算法.....	39

4.4.1 編碼及解碼	40
4.4.2 產生初始族群	40
4.4.3 適應值評估	40
4.4.4 選擇與交配	41
4.4.5 突變	41
4.4.6 菁英族群	41
4.5 求解 BP-based 模式方法三：接合點搜尋法	42
4.5.1 PoT 策略下無產能限制之 BP-based 問題	42
4.5.2 最佳利潤曲線之接合點及其性質	45
4.5.3 PoT 策略下無產能限制之 BP-based 模式理論分析	47
4.5.4 接合點搜尋演算法	48
第五章 範例說明與數值測試	49
5.1 範例說明	49
5.1.1 判斷缺貨產品及設定縮放比例	50
5.1.2 計算補貨週期乘數 k 與基本週期 B	50
5.1.3 建立排程(LPT 派工法)	54
5.1.4 求解 TVLS 模式	54
5.2 隨機數據測試	63
5.2.1 單次數值分析	64
5.2.2 多次數值分析	65
5.2.3 運算時間分析	67
5.2.4 小結	68
第六章 結論與建議	69
6.1 結論	69
6.2 建議	69

參考文獻.....71
簡歷.....74



表目錄

表 2.1—針對產品 i 生產批量排程與決策變數 x_{ij} 之關係	10
表 2.2—公式(11)展開結果表	10
表 4.1—接合點確定法之演算結果.....	46
表 5.1—範例說明之參數設定.....	49
表 5.2—判斷缺貨及縮放比例之結果.....	50
表 5.3—基因演算法執行 1000 個世代運算之結果.....	51
表 5.4—接合點搜尋法執行過程.....	52
表 5.5—接合點搜尋法之結果.....	53
表 5.6—三種方法之排程及週期結果.....	54
表 5.7—共同週期法結果代入 TVLS 模式求解(情況一).....	56
表 5.8—基因演算法結果代入 TVLS 模式求解(情況一).....	56
表 5.9—接合點搜尋法結果代入 TVLS 模式求解(情況一).....	57
表 5.10—情況一比較表.....	57
表 5.11—共同週期法結果代入 TVLS-search-T 模式求解(情況二).....	58
表 5.12—基因演算法結果代入 TVLS-search-T 模式求解(情況二).....	59
表 5.13—接合點搜尋法結果代入 TVLS 模式求解(情況二) (上半)	60
表 5.13—接合點搜尋法結果代入 TVLS 模式求解(情況二) (下半)	61
表 5.14—鄰域搜尋法改善排程與求解對 T 搜尋的 TVLS 模式數值比較.....	62
表 5.15—產品資料範圍.....	63
表 5.16—10 品項隨機產品資料.....	64
表 5.17—隨機產品範例求解結果.....	64
表 5.18—產品數為 10 項下對不同利用率求解結果(與共同週期法比較).....	65
表 5.19—產品數為 30 項下對不同利用率求解結果(與共同週期法比較).....	66

圖目錄

圖 1.1—研究情境圖.....	3
圖 1.2—研究流程圖.....	4
圖 2.1—傳統 ELSP 存貨曲線圖.....	7
圖 2.2—TVLS 存貨曲線圖.....	12
圖 3.1—問題情境圖.....	16
圖 3.2—本研究之 TVLS 存貨曲線圖.....	20
圖 3.3—本研究之 BP-based 存貨曲線圖.....	26
圖 3.4—以縮放比例 θ_i 調整之存貨曲線圖.....	27
圖 4.1—程式演算流程圖.....	32
圖 4.2—固定排程下對 T 搜尋 TVLS 目標值(1).....	36
圖 4.3—固定排程下對 T 搜尋 TVLS 目標值(2).....	37
圖 4.4—使用 LPT 派工法產生的排程失誤.....	38
圖 4.5—產品 1 之最大化利潤函數圖型.....	43
圖 4.6— $TP_{PoT}(B)$ 函數圖型.....	44
圖 4.7—接合點搜尋法執行過程.....	53
圖 5.1—10 品項運算時間散佈圖.....	67
圖 5.2—30 品項運算時間散佈圖.....	67

第一章 緒論

經濟批量排程問題(Economic Lot Scheduling Problem, ELSP)最早在 1958 年提出雛形後，至今近 50 年被學者鑽研及改良，其目標在於決定出能使總成本最小化的生產批量大小和產品生產順序。

而為了更接近現實的生產情況，現今的學者又將 ELSP 細分成各種的情況來探討，其中一種是生產製造的同時，考量有逆物流並且回收再製造之行為。本研究探討不同於已有文獻研究之再製造情境，架設出新的利潤最大化再製造 ELSP 模式，並提出求解辦法。

本章將對本研究之作概括性的描述，分別為以下三個部分：研究動機與目的、研究範圍、研究架構與流程。

1.1 研究動機與目的

過去經濟批量排程問題已經被套用在許多種生產情境上，例如：金屬工廠、汽車工業、電子產品等，而本研究之情境考量汽車產業之情境。

現今世界上，由於早期人類社會對環境的予取予求，資源匱乏的警訊早已被國際間強調，環境保護以及資源回收等議題也逐漸抬頭，各種產業都開始改良產品，除了能夠在有效率的情況下使用資源外，產品的廢棄物也要能進行回收。

汽車產業也在汽車產品內加入環保意識，如 VOLVO 汽車公司在 2009 年就提出了環保汽車的核心價值，用心選擇原料與設計零件，達到產品報廢時有 99% 的零組件可以被回收的成果；另外同年福斯汽車也與史丹佛大學合作設立「福斯汽車創新研究室」，積極發產安全綠色環保車；而近年來也有所謂油電混合車的出現，顯示出汽車產業與環保理念在未來的緊密結合。

但基於產品新舊以及再生品之原料純度較差之考量，汽車產品的零組件在回收再製造之後的價值與新品無法同等並列，原廠零件的售價通常偏高，故經過再製造或修理等復原劉成的零件通常進入的是二手或次級商品市場，與新品市場的顧客需求有所不同。

故本研究討論此情境架構之再製造 ELSP 問題，考量一個汽車工廠裡只有單一的生產線，生產的每種零件都可以回收後進行再製造或修理，變成另一種再

生產品，進入不同於新品市場的再生品市場。本研究考慮生產工廠在一次生產週期下的平均利潤最大化，使用時間變動批量大小方法(Time-varying lot-sizes approach, TVLS)模式進行建模，並使用 Basic-period-based (BP-based)模式求解、構建出排程及週期長度兩種資訊，代入本研究之 TVLS 最佳解求解演算法，其中分別使用共同週期法、基因演算法、接合點搜尋法等三種方式求解 BP-based 模式，並以共同週期法為比較基礎，比較各模式之解構建出之排程及週期長度資訊所帶來的求解品質。

1.2 研究範圍與假設

本研究之基本假設與 Rogers(1958)相同，並增加數項與再製造生產關聯之假設，分點敘述如下：

傳統 ELSP 假設：

1. 工廠生產多種產品且僅有一生產線，同一時間點下只能生產一種產品。
2. 產品生產時會發生整備成本與整備時間，為已知常數，與生產順序獨立。
3. 產品的需求率、製造率、單位存貨存貨成本為已知常數。
4. 期初有一定之存貨量使生產排程規畫可以實行。一般產品不允許缺貨。

情境假設：

1. 每一種「一般產品」(即新品)都可以被回收再製成另一種「再生產品」(即二手產品)，兩者視為不同之產品，服務不同的市場。
2. 假設再生產品原料之回收率與對應之一般產品需求率成一線性關係。
3. 再生產品原料之取得成本(已包含處理、清洗並恢復到可以作為原料利用所發生之成本)與回收量呈線性關係。
4. 再生產品原料的回收不需要整備時間，每次的回收只能用於當次的再生產品生產。
5. 當再生產品開始生產時，原料之消耗率與再生產品之生產率呈一線性關係，且考慮回收產品有所耗損，製造一個再生產品會消耗掉大於一單位的一般產品原料。
6. 再生產品則可以缺貨，而缺貨之銷貨損失(lost sales)僅考慮未能賺取的機會成本，不考慮無形之商譽損失。
7. 一般產品與再生產品之市場需求獨立，兩者顧客皆不會有上買(buy-up)或

下買(buy-down)之行為。

其他參數假設：

1. 由於再生產品原料純度較差，故假設一般產品之售價大於再生產品。
2. 考量原料有回收、分解等工序，故假設一般產品之生產率大於再生產品。
3. 考慮供應鏈產品上下游關係，再生產品為原料受過加工過後，故再生產品之存貨成本大於再生產品原料。

1.3 研究架構與流程

本研究之架構與流程如圖 1.2 所示，並詳述如下：

(1) 確定研究目標

本研究確立新的再製造 ELSP 情境，建立數學模式及提出求解方法。

(2) 文獻內容回顧

根據主題，針對再製造相關文獻、再製造 ELSP、ELSP 求解方法進行回顧。

(3) 建立 TVLS 模式及 BP-based 模式

依照問題情境進行符號定義及目標式、限制式等推導，並證明本研究之模式為凹性函數。

(4) 建立 TVLS 最佳解求解演算法與範例說明

根據 TVLS 模式提出最佳解求解演算法，並套入範例進行演算法說明。

(5) 數值分析與結論

由於為新的問題，目前無範本題庫，將採用文獻使用之隨機題庫建立參數，加入再製造之部分隨機產生題庫，分別進行小範例及大範例之測試。

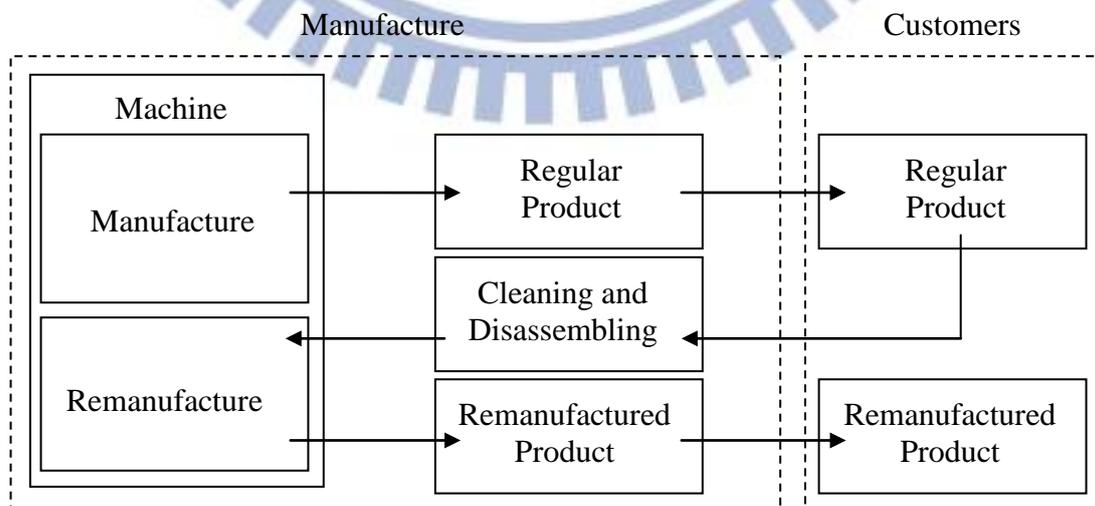


圖 1.1—研究情境圖

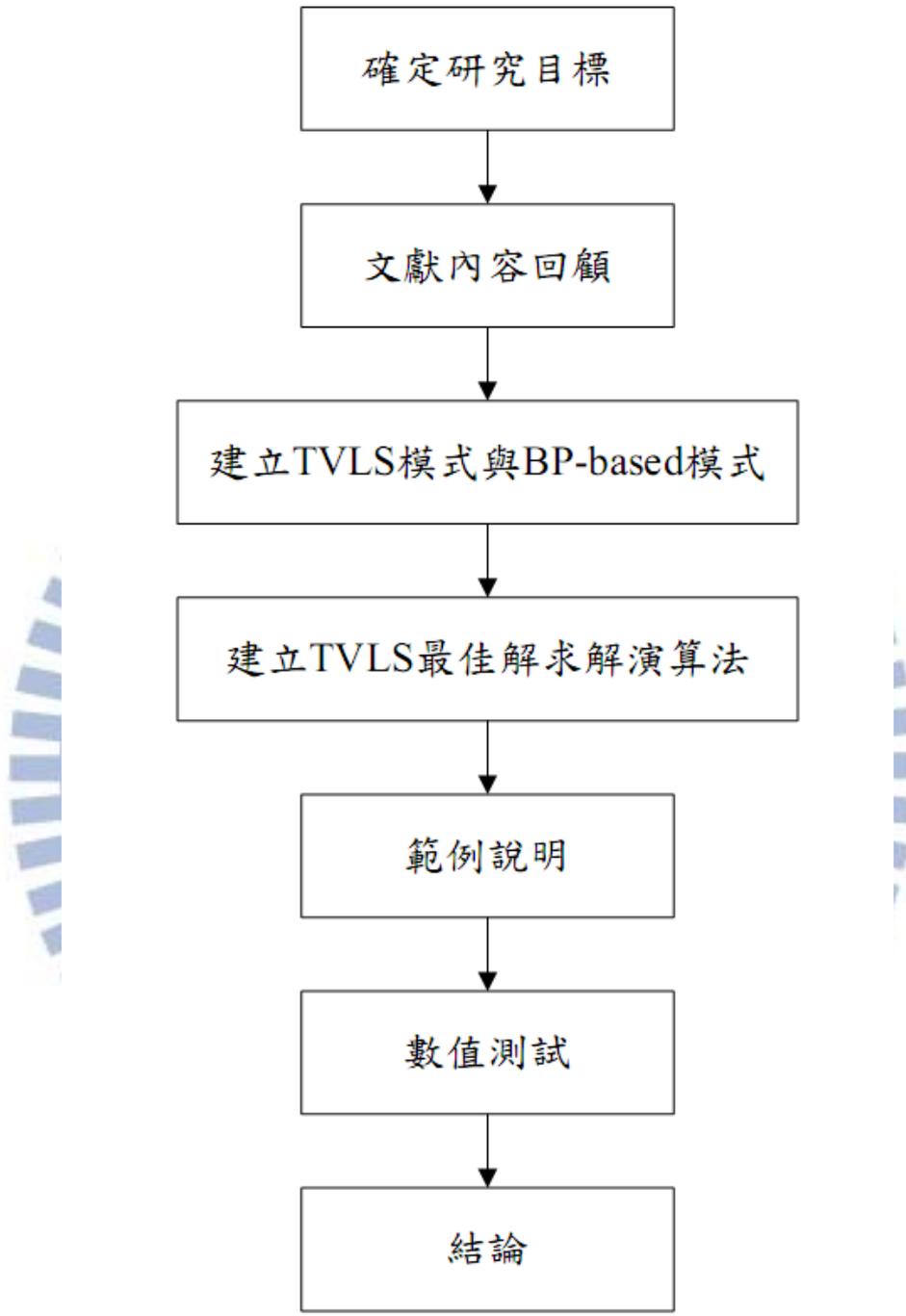


圖 1.2—研究流程圖

第二章 文獻回顧

本章節針對「再製造的經濟批量排程問題」，進行三個層面的歷史文獻回顧，分別是經濟批量排程問題及其求解方法、再製造有關之生產存貨管理文獻、再製造經濟批量排程問題，作為對於本次目標—情架架構與再製造經濟批量排程問題的求解方法之參考內容。

2.1 經濟批量排程問題

經濟批量排程問題的雛型最早由 Rogers(1958)提出。其情境假設在無限期規畫下，單一機器生產多產品時，決定出能使總成本最小化的生產批量大小和產品生產順序。按照 Boctor(1982)對於排程問題的分類，ELSP 屬於確定性無限期多品項單一機台之問題。此問題已由 Hsu(1983)證明為一 NP-Hard 問題。

對於 ELSP 問題的假設，綜合 Bomberger(1966), Davis(1990), Khouja(1998)的整理，有以下數點：

1. 所有產品皆由一個機台生產，且機台的能力可以滿足所有的需求量；
2. 所有產品的需求率、生產率、整備時間、整備成本、存貨持有成本都是已知且確定的，產品的需求是持續不斷的。
3. 同一時間點上，機器只能生產一種產品；
4. 整備時間及整備成本不會因為製造的順序或批量大小而有所增減；
5. 存貨持有成本與該產品之存貨數量為線性關係；
6. 每個產品的需求率為確定已知常數，且所有需求皆必須被滿足；

而 Elmaghraby(1978)整理相關文獻並將問題解法分類為：(1)分析式解法 (Analytic Approaches)及(2)啟發式解法(Heuristic Approaches)。分析式解法是將原有問題加入某一些限制或假設，來獲得該情況下的最佳解。此類解法以獨立解法(Indepert Solution; IS)、共同週期法(Common Cycle Approach; CC)、基本週期法(Basic Period)及延伸基本週期法(Extended Basic Period; EBP)最具參考性。而另外亦有人使用 Maxwell(1964)和 Delporte and Thomas (1978)之時間變動批量大小方法(Time-varying lot sizes approach)。

而啟發式解法的目的則在於縮小問題的求解範圍以增加解題的速利及求解

大型問題的能力，但通常只能求得較好的近似最佳解，並不保證能求得最佳解。其中備受關注且為後世廣為採納的是 McClain, Thomas and Mazzola (1992)提出的二冪策略 (Power of 2 policy; PoT)，且證明出採用此方法在啟發式解法獲得之近似最佳解與一般整數策略 (General Intigel; GI) 之近似解下界只會差 6%。

許多學者為了讓經濟批量排程問題更貼近生產的現實情況，也開始為 ELSP 延伸出各種方面的探討，Khoury (2001) 考量單一機台產能不足情況下的 ELSP、Moon et al. (1991) 使用共同週期法求解生產速率可改變的 ELSP、Giri et al. (2003)、蔣其軒 (2007) 考慮生產機台可能失控的不完美生產系統 ELSP、Leachman and Gascon (1988)、Gallego (1990) 和 Brander (2005) 將需求考量為隨機性需求等。

接著進行基本求解方法之介紹，先進行 ELSP 的變數定義，接著介紹現今較具參考性的五種 ELSP 解法：獨立解法 (Independent Solution Approach)、共同週期法 (Common Cycle Approach)、基本週期法 (Basic Period Approach) 和延伸基本週期法 (Extended Basic Period Approach)、時間變動批量大小方法 (Time-varying lot sizes approach)。

變數定義

n = 產品種類數目。

i = 產品類別， $i = 1, 2, \dots, n$

d_i = 產品 i 每單位時間的需求率 (units/year)

p_i = 產品 i 每單位時間的生產率 ($p_i > d_i$) (units/year)

a_i = 生產產品 i 的整備成本 (\$/setup)

s_i = 生產產品 i 的整備時間 (years)

h_i = 產品 i 每單位時間的存貨持有成本 (\$/unit/year)

T_i = 產品 i 的週期循環時間 (years)

T_i^* = 產品 i 的最佳獨立週期循環時間 (years)

B = 基本週期 (years)

k_i = 產品 i 的基本週期乘數

$\mathbf{K} = (k_1, k_2, \dots, k_n)$ ， k_i 的集合

ρ_i = 稼動率，即需求率與生產率之比值 d_i / p_i ，且 $\sum \rho_i \leq 1$

$PT_i = \rho_i T_i$ ，產品*i*每批量在機器上的處理時間

$CT_i = (1 - \rho_i) T_i$ ，產品*i*每批量在機器上的存貨消耗時間(或簡稱消耗時間)

$TPT_i = s_i + \rho_i T_i$ ，產品*i*每批量在機器上的總生產時間 (total production time)

$C_i = \frac{a_i}{T_i} + \frac{h_i}{2} d_i (1 - \rho_i) T_i$ ，產品*i*的總成本 (\$/units)

C_i^* = 產品*i*的最佳總成本(\$/units)

$TC = \sum_i^n C_i$ 所有產品的總成本(\$/units)

TC^* = 所有產品的最佳總成本(\$/units)

產品週期

ELSP 最重要之觀念在於如何界定一個產品的產品週期，如下方圖 2.1 為傳統 ELSP 的存貨曲線圖，產品*i*要生產時的時候會發生整備時間 s_i ，之後開始生產而有生產時間 $PT_i = \rho_i T_i$ ，結束生產之後會有存貨消耗時間 $CT_i = (1 - \rho_i) T_i$ (或簡稱消耗時間)，而產品週期則是從存貨從 0 開始增加到消耗至 0 的時間段，其中存貨成本即是存貨曲線下面積，利用三角型面積公式可以求得該值，三角型之高為生產時間 $\rho_i T_i$ 乘以存貨增加率 $p_i - d_i$ ，底即是產品週期 T_i ，兩者相乘除以 2 再乘上單位存貨成本 h_i 即得到，加上整備成本即是產品*i*的總成本：

$$C_i = \frac{a_i}{T_i} + \frac{h_i}{2} d_i (1 - \rho_i) T_i \quad (1)$$

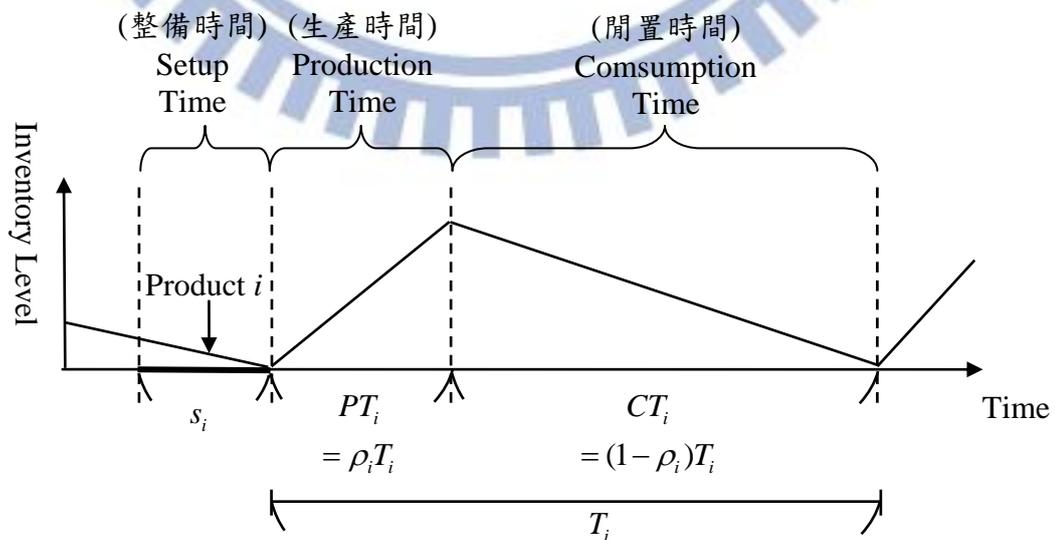


圖 2.1—傳統 ELSP 存貨曲線圖

2.1.1 獨立解法

Elmaghraby (1978)的獨立解法(Independent Solution Approach; IS)之數學模式利用每個產品總成本微分求極值的方式，求出產品*i*的最佳週期循環時間：

$$\text{產品 } i \text{ 的最佳週期循環時間 } T_i^* = \sqrt{\frac{2a_i}{h_i d_i (1 - \rho_i)}} \quad (2)$$

$$\text{產品 } i \text{ 的最佳總成本 } C_i^* = \sqrt{2a_i h_i d_i (1 - \rho_i)} \quad (3)$$

獨立解法中的每個產品只考慮本身的最佳解，所有的產品會按照求得的 T_i^* 進行生產，如果產品之間沒有發生兩產品在同一時間點下生產的情況，則該解為可行解，也必為最佳解，反之則為不可行解，然而要驗證產品之間是否有重疊現象並不容易。雖然獨立解法不能保證其解的排程一定合理，但此方法的解常被作為 ELSP 模式中可行解的下界，進行與其他方法的比較。

2.1.2 共同週期法

共同週期法(Common Cycle Approach; CC)由 Hanssmann (1962)所提出，乃方法中最簡單，且保證其解必為可行解的排程方法。其假設每個產品在每個週期週要生產，即把所有的產品都包含在一個共同的循環時間週期(T_{CC})內，所以 T_{CC} 必須夠大，之後所有產品按 T_{CC} 的時間循環生產。其模式如下：

$$\text{Min } \sum_{i=1}^n \left[\frac{a_i}{T_{CC}} + \frac{h_i}{2} d_i (1 - \rho_i) T_{CC} \right] \quad (4)$$

$$\text{s.t. } \sum_{i=1}^n (s_i + \rho_i T_{CC}) \leq T_{CC} \quad (5)$$

$$T_{CC} = \max \left(\sqrt{\frac{2 \sum_{i=1}^n a_i}{\sum_{i=1}^n h_i d_i (1 - \rho_i)}}, \frac{\sum_{i=1}^n s_i}{1 - \sum_{i=1}^n \rho_i} \right) \quad (6)$$

其限制式的概念是所有產品生產時間加總不可以超過循環時間，經過公式的化簡後可得循環時間之下界 T_{CC}^{LB} ，與最佳循環週期公式求得之值取較大者，即公式(6)。求解時先求得 T_{CC} ，再代入目標式中計算總成本，其值可作為 ELSP 模式中可行解的上界，常作為啟發式解法的判斷依據，也就是說如果搜尋到的解已經高於此上界，就不必花費較多的時間於判斷可行解上。

2.1.3 基本週期法

基本週期法(Basic Period Approach; BP)由 Bomberger (1966)提出，最早是使用動態規劃進行求解，現今其概念常用於發展啟發式解法。模式如下：

$$\text{Min } TC(\mathbf{K}, B) = \sum_{i=1}^n \left[\frac{a_i}{k_i B} + \frac{h_i}{2} d_i (1 - \rho_i) k_i B \right] \quad (7)$$

$$\text{s.t. } \sum_{i=1}^n (s_i + \rho_i k_i B) \leq B \quad (8)$$

$$T_i = k_i B \quad (9)$$

where k_i is integer, $k_i \in \{1, 2, \dots\}$

基本週期法假設所有產品在第一週都需要生產，且所有產品的生產週期為基本生產週期 B 的整數 k_i 倍，即每 k_i 週期生產一次，故也可以說共同週期法為所有產品的生產週期乘數 k_i 都為 1 的基本週期法特例。由此可見，基本週期法放鬆了共同週期法每週期都需要生產的限制。但也因為 B 要足夠長以容納所有的產品在第一週期的生產，往後的週期不是每個產品都要生產，所以機台往往會空下一段閒置的時間，造成產能上的浪費。

2.1.4 延伸基本週期法

Elmaghraby(1978)改良了 Bomberger(1966)求解 BP 方法，提出延伸基本週期法(Extend Basic Period Approach; EBP)的概念，其後 Haessler(1979)提出改善 EBP 方法之規劃程序，利用二冪次整數(Power of Two, PoT)策略來處理基本週期乘數。Yao(2001)建構出 PoT 下 EBP 之數學模式，並提出求解方法。EBP 的基本模式如下：

$$\text{Min } \sum_{i=1}^n \sum_{j=0}^{z_i} \sum_{l=1}^{2^j} \left\{ \frac{a_i}{B} 2^{-j} + \frac{h_i}{2} d_i (1 - \rho_i) 2^j B \right\} x_{ijl} \quad (10)$$

$$\text{s.t. } \sum_{i=1}^n \sum_{j=0}^{z_i} x_{ij\varphi(j,t)} (s_i + 2^j \rho_i B) \leq B, \text{ for } t = 1, \dots, 2^{\max\{z_i\}} \quad (11)$$

$$\sum_{j=0}^{z_i} \sum_{l=1}^{2^j} x_{ijl} = 1, \text{ for all } i = 1, 2, \dots, n \quad (12)$$

$$\text{where } \varphi(j,t) = \begin{cases} t \bmod 2^j, & \text{if } t \neq \gamma 2^j, \gamma \in N \\ 2^j, & \text{if } t = \gamma 2^j, \gamma \in N \end{cases}$$

and $x_{ijl} \in \{0,1\}$ for all i, j, l

EBP 中放鬆了 BP 中所有產品在第一週期皆需要生產的限制，其中 x_{ijl} 表示產品 i 的循環週期為 2^j 個基本週期 ($T_i = 2^j B$)，且從第 l 個時間階段開始生產。以下 $z_i = 3$ 作為例子作說明，則 2^j 的範圍為 $\{2^0, 2^1, 2^2, 2^3\}$ 之間，則產品 i 可能的生產排程可以以表 2.1 來呈現，如 $x_{i21} = 1$ 表示為產品 i 的生產週期為每 $2^j = 2^2 = 4$ 個基本週期生產一次，且在第 1 個時間階段開始生產。又因為公式(12)，所以表格內每列橫向加總的 x_{ijl} 會等於 1，確保一個產品只會有一種生產循環。

再來以下頁表 2.2 來表示公式(11)的計算，假設以 period 4 ($t=4$) 這一系列來進行說明，在第 1 行 ($j=0, t=4$)，因為 $4 = 4 \times 2^0$ ，所以 $\varphi(j, t) = \varphi(0, 4) = 2^0 = 1$ ，因此得到 $(s_i + 2^0 \rho_i B)x_{i01}$ ；而第 2 行 ($j=1, t=4$)，因為 $4 = 2 \times 2^1$ ，所以 $\varphi(j, t) = \varphi(1, 4) = 2^1 = 2$ ，因此得到 $(s_i + 2^0 \rho_i B)x_{i12}$ ；第 3 行 ($j=2, t=4$)，因為 $4 = 1 \times 2^2$ ，所以 $\varphi(j, t) = \varphi(2, 4) = 2^2 = 4$ ，因此得到 $(s_i + 2^0 \rho_i B)x_{i24}$ ；最後第 3 行 ($j=3, t=4$)，因為 $4 \neq \gamma \times 2^3, \forall \gamma \in N$ ，所以 $\varphi(j, t) = \varphi(3, 4) = 4 \bmod 2^3 = 4$ ，因此得到 $(s_i + 2^0 \rho_i B)x_{i34}$ 。

表 2.1—針對產品 i 生產批量排程與決策變數 x_{ijl} 之關係

$z_i = 3$								
$j = 0$	x_{i01}							
$j = 1$	x_{i11}	x_{i12}						
$j = 2$	x_{i21}	x_{i22}	x_{i23}	x_{i24}				
$j = 3$	x_{i31}	x_{i32}	x_{i33}	x_{i34}	x_{i35}	x_{i36}	x_{i37}	x_{i38}

表 2.2—公式(11)展開結果表

產品				
Period	2^0	2^1	2^2	2^3
1	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i11}$	$(s_i + 2^0 \rho_i B)x_{i21}$	$(s_i + 2^0 \rho_i B)x_{i31}$
2	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i12}$	$(s_i + 2^0 \rho_i B)x_{i22}$	$(s_i + 2^0 \rho_i B)x_{i32}$
3	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i11}$	$(s_i + 2^0 \rho_i B)x_{i23}$	$(s_i + 2^0 \rho_i B)x_{i33}$
4	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i12}$	$(s_i + 2^0 \rho_i B)x_{i24}$	$(s_i + 2^0 \rho_i B)x_{i34}$
5	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i11}$	$(s_i + 2^0 \rho_i B)x_{i21}$	$(s_i + 2^0 \rho_i B)x_{i35}$
6	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i12}$	$(s_i + 2^0 \rho_i B)x_{i22}$	$(s_i + 2^0 \rho_i B)x_{i36}$
7	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i11}$	$(s_i + 2^0 \rho_i B)x_{i23}$	$(s_i + 2^0 \rho_i B)x_{i37}$
8	$(s_i + 2^0 \rho_i B)x_{i01}$	$(s_i + 2^0 \rho_i B)x_{i12}$	$(s_i + 2^0 \rho_i B)x_{i24}$	$(s_i + 2^0 \rho_i B)x_{i38}$

2.1.5 時間變動批量大小方法

除了以上四種 ELSP 求解方法外，尚有一方法稱為時間變動批量大小方法 (Time-varying lot sizes approach, TVLS)。此方法與傳統 ELSP 之架構較不相同，由 Maxwell(1964) 和 Delporte and Thomas (1978) 提出，模式設計如下：

變數定義

n = 產品之品項數

l = 生產排程之總生產次數， $l \geq n$

i = 產品類別， $i = 1, 2, \dots, n$

d_i = 產品 i 每單位時間的需求率 (units/year)

p_i = 產品 i 每單位時間的生產率 ($p_i > d_i$) (units/year)

a_i = 生產產品 i 的整備成本 (\$/setup)

s_i = 生產產品 i 的整備時間 (years)

h_i = 產品 i 每單位時間的存貨持有成本 (\$/unit/year)

T = 整個生產順序的週期循環時間 (years)

q^k = 第 k 個生產產品之產品品項， $\forall k \in (1, 2, \dots, l)$

s^k = 第 k 個生產產品之整備時間

t^k = 第 k 個生產產品之生產時間

u^k = 第 k 個生產產品之閒置時間，即生產完到下一個產品開始整備之前的時間

$N_i = \{k | q^k = i\}$ 生產產品 i 之順序的集合

L_o = 從開始生產產品 o 到下次(但不包含)生產產品 o 之製造順序集合

產品週期

TVLS 的存貨圖如圖 2.2 所示，分別由整備時間(Setup Time)、生產時間(Production Time)、閒置時間(Idle Time)所構成，其中閒置時間的定義是本次生產完至開始整備下次生產中間的空閒時間，而 TVLS 對於產品週期的假設是五種方法中最為彈性的方法，假設每次生產產品的生產時間與閒置時間皆可以不同，故每次的產品週期亦不相同。

產品總成本的概念與傳統 ELSP 相同，為整備成本及存貨成本加總，存貨成本一樣使用三角型面積公式求出，故生產排程 q 中第 k 個產品的總成本如下：

$$C^k = a^k + \frac{1}{2} h^k (p^k - r^k) \left(\frac{p^k}{r^k} \right) (t^k)^2$$

故可以寫出 TVLS 模式之目標式：

$$\text{Min inf}_{k \in \mathcal{J}} \min_{\substack{t \geq 0 \\ u \geq 0 \\ T \geq 0}} \frac{1}{T} \left(\sum_{n=1}^l a^k + \sum_{n=1}^l \frac{1}{2} h^k (p^k - r^k) \left(\frac{p^k}{r^k} \right) (t^k)^2 \right) \quad (13)$$

$$\text{s.t. } \sum_{k \in N_i} p_i t^k = d_i T \quad (14)$$

$$\sum_{k \in L_o} (t^k + s^k + u^k) = \left(\frac{p^o}{d^o} \right) t^o \quad (15)$$

$$\sum_{k=1}^l (t^k + s^k + u^k) = T \quad (16)$$

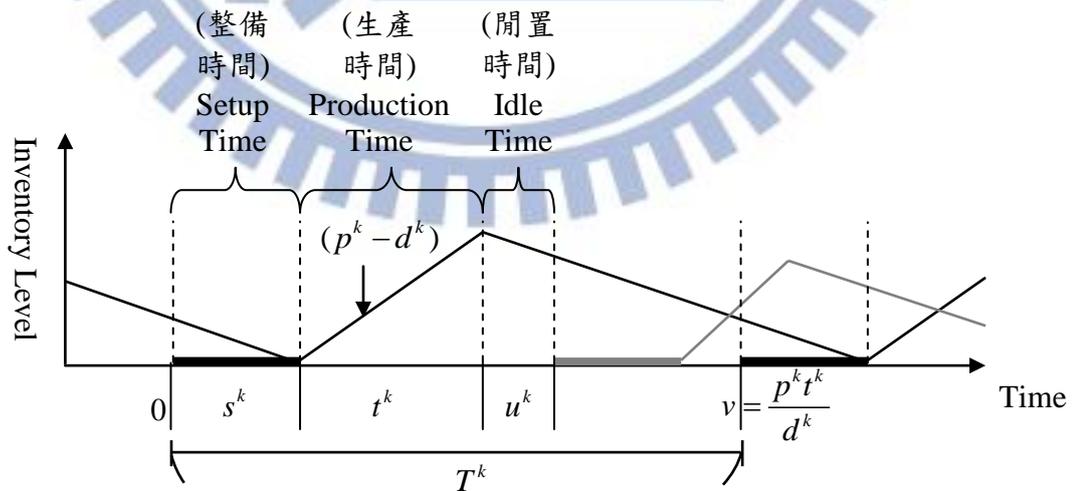


圖 2.2—TVLS 存貨曲線圖

公式(13)一樣考慮最小化在所有可行之解集合 \mathcal{S} 裡找到決策變數 (t^k, u^k, T) 使得目標式最小化，其中目標式包含存貨成本與整備成本。公式(14)定義將所有生產產品 i 的生產批次產量加總後必須等於週期時間產生的總需求量。公式(15)表示本次生產產品 i 之供給量必須要能夠提供到下次生產之前的總需求量。公式(16)表示所有的生產、整備及閒置時間加總。

而事實上公式(14)是多餘的(redundant)，因為公式(15)代入公式(16)，即可以表現出公式(14)。

此方法與其他方法最大的不同點在於，該模式之限制式交集即為本問題之可行解範圍，故不需要在搭配演算時逐步檢驗可行性，且因為方法的特性，可以求解出共同週期法較優品質之解。

2.2 再製造之生產存貨管理文獻

有關再製造的生產存貨管理，本節將先行定義逆物流中的「再製造」，再進入生產存貨管理文獻之研究。

2.2.1 逆物流中的再製造

再製造為逆物流眾多流程中的一環，美國逆物流協會 (Reverse Logistics Executive Council) 將逆物流定義為「逆物流是一種物品移動的過程中，從最終目的地搬運到其他地點，主要目的是為了獲得價值，或者是為了對產品做適當的處置。」且包括了以下活動：

1. 商品退回的處理，例如：物品的損壞、週期性的退貨 (Seasonal Inventory)、重新補貨 (Restock)、廢物利用 (Salvage)、產品召回 (Recall) 以及過量的存貨 (Excess inventory)。
2. 回收規劃 (Recycling Programs)
3. 有害物質的規劃 (Hazardous Material Programs)
4. 老舊設備的配置 (Obsolete Equipment Disposition)
5. 資產復原 (Asset Recovery)
6. 產品維修 (Reconditioning)、再製造 (Remanufacturing) 及再翻新 (Refurbishing)。

其中的第 6 點之層級即為本研究中討論之再製造。

2.2.2 再製造生產存貨管理文獻

近年來環保意識逐漸提升，學界也開始重視回收與再製造等議題。從是否有期限可以分為有期限與無期限兩種假設。有期限之假設會有週期數的區別，存貨的計算通常使用期末或期初盤存，傅俊源(民 100)即討論多品項下週期需求的批量製造與再製造之生產規劃問題，Zhou(2011)研究單一產品回收後可以分類為不同的損壞等級，在總成本最小的情況下選擇每種等級再製造之生產量。Absi and Kedad-Sidhoum(2009)考慮多品項的生產排程下可能面臨的缺貨問題及最佳安全庫存。Pan *et al.*(2009)考慮有限產能下的製造與再製造動態需求生產批量問題，並且允許回收品報廢、產品可缺貨待補，使用動態規劃法求解不同劇本下的最下化成本目標式。Li *et al.* (2007)使用基因演算法求解目標式為最小化成本的製造與再製造動態需求生產批量問題。

而本研究主要探討無限期之生產與存貨管理模式，是經濟生產批量(Economic production quantity, EPQ)以及其後延伸領域採用的解設，ELSP 即包含在其中，此段文獻內容至 2.3 節詳述。

其他亦有從多層供應鏈角度探討再製造問題，如鄧喻心(民 98)研究兩階層存貨系統下，透過最佳訂購量和最佳回收百分比將利潤最大化的補貨策略問題。亦有討論回收成本與市場關係之研究，如黃志強(民 100)探討再製造過程中最佳的回收努力值及最佳回收量，以及 Zhou(2011)假設回收成本為凸函數和產品之售價需求率呈線性關係來求解最佳回收量。

2.3 再製造經濟批量排程問題

目前已有一種探討再製造之經濟批量排程問題，為 Economic lot scheduling problem with rework，又稱 ELSPR，由 Tang and Teunter (2006)提出。該問題之情境假設產品有一定比例會經過回收再製造後成為新品，並且服務同一種市場。

Tang and Teunter (2006)在首度提出時採用共同週期法假設，再製造與新品製造使用同一機台之產能，使用混合整數規劃(Mixed integer programming)建立模式，搭配演算法求解。之後 Teunter *et al.* (2008)同樣使用共同週期法假設，但將新品生產與再製造切分為兩條產線，結果說明兩條產線可以大大減少存貨成本。Teunter *et al.* (2009)使用共同週期法假設，設計求解 ELSPR 的啟發式演算

法，並建立 120 個範例題測驗演算法效率。Zanoni *et al.* (2012)使用基本週期法假設求解 Tang and Teunter (2006)，並建立啟發式演算法增加求解效率，並與 Tang and Teunter (2006)之共同週期法最佳解與 2009 之 120 個範例題進行測試。

國內亦有李翔豐(2010)以共同週期法求解允許變動生產速率之製造與再製造經濟批量排程問題，王俊傑(2009)求解不完美系統下允許回收再製造之經濟批量排程問題。

2.4 小結

從文獻回顧來看，雖然已有很多新進探討再製造的生產存貨模式，經濟批量排程問題也有 ELSPR 在研究再製造的課題，前者通常假設有期限，使用產量而不是使用生產時間作為決策變數，後者使用共同週期法和基本週期法進行建模和求解，但兩者的情境與本研究仍然有非常明顯的差別：

1. 本研究之再生產品服務新市場，而文獻內容皆為服務原產品之市場
2. 本研究採用無限期生產，考慮平均利潤最大化模式
3. 本研究建立 TVLS 模式，先透過求解 BP-based 模式建置出週期長度及排程資訊，再利用 TVLS 最佳解求解演算法求解。

期待本研究能在 ELSP 領域建構出新的模式，並且提出有效之求解方法。

第三章 研究方法

本章節將詳述兩種模式之設計，先構建 TVLS 模式，詳述一般產品與再生產品在限制式上之差別，並驗證模式在給定排程與週期下為凹性函數(concave function)；再來以對應之關係列出 BP-based 模式，並修改簡化之。以下先描述情境、說明問題的假設及限制、進行符號定義、再呈現出修改之數學模式。

3.1 情境描述與假設

本研究之情境設計詳見下方圖 3.1，假定每種一般產品 (Regular product) 都可以進行回收再製造(Remanufacture)，生成對應之再生產品 (Remanufactured product)。現有一工廠同時有製造與再製造行為，首先工廠生產出一般產品，並按照穩定需求率提供給下游顧客，生產到出貨之前會發生整備成本及存貨成本，而產品收出會有銷貨收入；之後在工廠決定回收時，會從顧客端依照穩定回收率回收產品，進行清理和分解處理等動作 (Cleaning and disassembling)，轉變為再生產品之原料，而此過程發生之所有成本稱為取得成本，另外原料也依照儲存時間有原料存貨成本；最後再生產品開始製造時，再生產品會依照一定生產速率進行生產，但考量回收之原料純度較差或毀損不全之問題，故會消耗多於一單位的一般產品才成生產一單位再生產品。再生產品售出亦有銷貨收入，但由於為次等級產品，故價格較新品低，另外再生產品可以缺貨，缺貨成本只認定應賺而未賺得之利潤。由於生產成本在研究認定為常數，故在模式中不考慮。

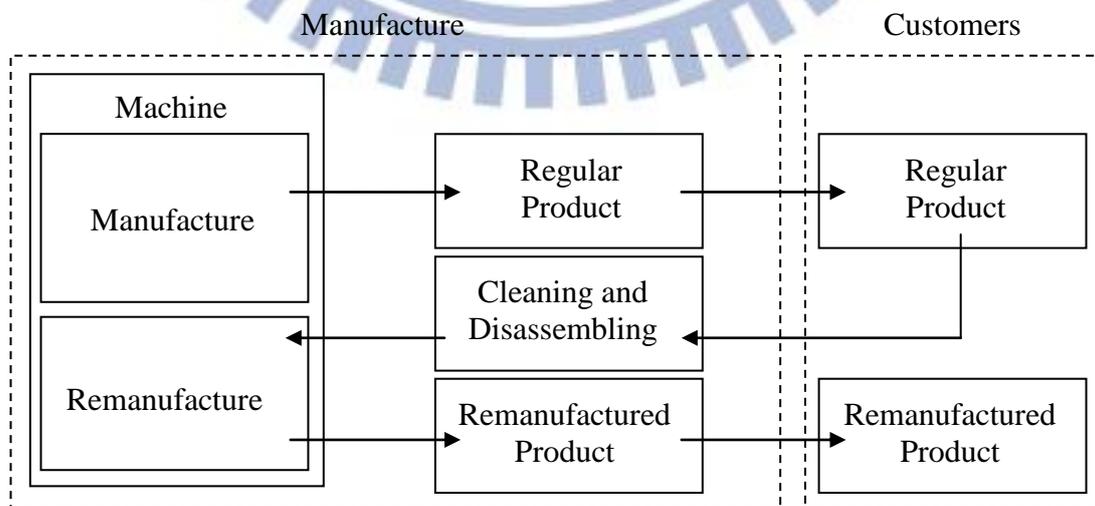


圖 3.1—問題情境圖

為了簡化問題之複雜度，將對以上情境做出數項假設，有利數學模式之建立。本研究之假設與如下：

傳統 ELSP 假設：

1. 工廠生產多種產品且僅有一生產線，同一時間點下只能生產一種產品。
2. 產品生產時會發生整備成本與整備時間，為已知常數，與生產順序獨立。
3. 產品的需求率、製造率、單位存貨存貨成本為已知常數。
4. 期初有一定之存貨量使生產排程規畫可以實行，且一般產品不允許缺貨。

情境假設：

1. 每一種「一般產品」(即新品)都可以被回收再製成另一種「再生產品」(即二手產品)，兩者視為不同之產品，服務不同的市場。
2. 假設再生產品原料之回收率與對應之一般產品需求率呈線性關係。此假設同王俊傑(2009)。
3. 再生產品原料之取得成本(已包含處理、清洗並恢復到可以作為原料利用所發生之成本)與回收量呈線性關係。此假設同 Li *et al.* (2007)。
4. 再生產品原料的回收不需要整備時間，每次的回收只能用於當次的再生產品生產。此假設同 Tang and Teunter(2006)的 ELSPR 假設。
5. 當再生產品開始生產時，原料之消耗率與再生產品之生產率呈一線性關係，且考慮回收產品有所耗損，製造一個再生產品會消耗掉大於一單位的一般產品原料。此假設為修改 Tang and Teunter(2006)對於 ELSPR 假設，該文獻中假設消耗率為 1。
6. 再生產品可以缺貨，而缺貨之銷貨損失(*lost sales*)僅考慮未能賺取的機會成本，不考慮無形之商譽損失。
7. 一般產品與再生產品之市場需求獨立，兩者顧客皆不會有上買(*buy-up*)或下買(*buy-down*)之行為。

其他參數假設：

1. 由於再生產品原料純度較差，故假設一般產品之售價大於再生產品。
2. 考量原料有回收、分解等工序，故假設一般產品之生產率大於再生產品
3. 考慮供應鏈產品上下游關係，再生產品為原料受過加工過後，故再生產品之存貨成本大於再生產品原料。

3.2 TVLS 模式

接著介紹本研究提出之 TVLS 模式，分別定義符號、說明模式，並按照研究架構，在給定排乘週期 T 和排程長度 l 的情況下，驗證模式為凹性函數 (concave function)，存在一極大值。

3.2.1 TVLS 符號定義

本研究將變數分為決策變數及已知變數做介紹，以方便讀者之閱讀。

決策變數：

T = 週期總長度(days)

$q = (q^1, q^2, \dots, q^k, \dots, q^l)$ 產品之生產順序， k 為生產順序之上標， l 為生產順序最大之上標值。

\bar{t}^k = 第 k 個生產之再生產品的原料回收至開始生產產品前的時間。在 TVLS 模式裡回收時間一樣與生產時間 t^k 相關聯，將於 3.2.2 詳述之。

u^k = 第 k 個生產產品之間置時間，即生產完到下一個產品開始整備前的時間

已知變數：

n = 一般產品種類數目，再生產品之數目亦同

i = 產品下標， $i = 1, 2, \dots, n$ 為一般產品， $i = n+1, n+2, \dots, 2n$ 為再生產品

d_i = 產品 i 每單位時間的需求率(units/day)

p_i = 產品 i 每單位時間的生產率($p_i > d_i$) (units/day)

ρ_i = 稼動率，即需求率與生產率之比值，且 $\sum \rho_i \leq 1$

a_i = 生產產品 i 的整備成本 (\$/setup)

s_i 生產產品 i 的整備時間 (hours)

h_i = 產品 i 每單位時間的存貨持有成本 (\$/unit/year)

\bar{h}_i = 產品 i 的原料存貨成本， $\bar{h}_i \leq h_i$ ；一般產品為 0

σ_i = 產品 i 的售價 (\$/unit)， $\sigma_{i+n} \leq \sigma_i$ ， $\forall i \in \{1, 2, \dots, n\}$

ξ_i = 產品 i 原料的回收參數， $0 \leq \xi_i \leq 1$ ， $\forall i$ ；一般產品為 0

r_i = 產品 i 的原料回收率(units/day)， $r_{i+n} = \xi_{i+n} \cdot d_i$ ， $\forall i \in \{1, 2, \dots, n\}$ ；一般產品為 0

λ_i = 產品 i 的原料消耗參數， $\lambda_i \geq 1$ ， $\forall i \in \{n+1, n+2, \dots, 2n\}$ ；一般產品為 0

γ_i = 產品 i 的原料消耗率，且與再生產品 i 的生產率為線性關係 $\gamma_i = \lambda_i \cdot p_i$

τ_i = 原料回收率與消耗率之比值，即 $\tau_i = \frac{\gamma_i}{r_i}$ ；一般產品為 0

α_i = 產品 i 的每單位變動取得成本 (\$/unit)；一般產品為 0

β_i = 產品 i 的每批次固定取得成本 (\$/unit)；一般產品為 0

$N_i = \{k | q^k = i\}$ 生產產品 i 之順序的集合

L_o = 從開始生產第 o 個產品到下次(但不包含)生產該產品之生產順序的集合

\bar{L}_o = 第 o 個產品生產前到(但不包含)上一次生產該產品之生產順序集合

v^k = 第 k 個生產的產品存貨從 0 開始增加至消耗為 0 的時間長數

3.2.2 TVLS 模式建立

接著進行收入項及成本項公式之說明，建構出本研究 TVLS 模式的目標式及限制式。

3.2.2.1 回收時間與生產時間之關係

本研究假設每次的原料回收只供應當次的再生產品生產，所以每次的原料回收會於該次生產全部用完，依此概念畫出下頁圖 3.2，令第 k 個生產產品的生產時間為 t^k 、回收時間為 \bar{t}^k ，原料按穩定回收率 r^k 不斷累積原料，到達開始生產時的回收量為 $r^k \bar{t}^k$ ，而開始進行生產時開始進行原料的消耗，但同時也繼續回收，故存貨減少率為 $(\gamma^k - r^k)$ ，故原料從開始消耗至完全消耗完的時間可以表示為 $\frac{r^k \bar{t}^k}{(\gamma^k - r^k)}$ ，然而此時間長度亦為第 k 個生產產品的生產時間，故針對再生產品的部分，可以寫出第 k 個生產的產品生產時間與回收時間之關係：

$$t^k = \frac{r^k \bar{t}^k}{(\gamma^k - r^k)} \Rightarrow \bar{t}^k = \frac{(\gamma^k - r^k) t^k}{r^k} \quad (16)$$

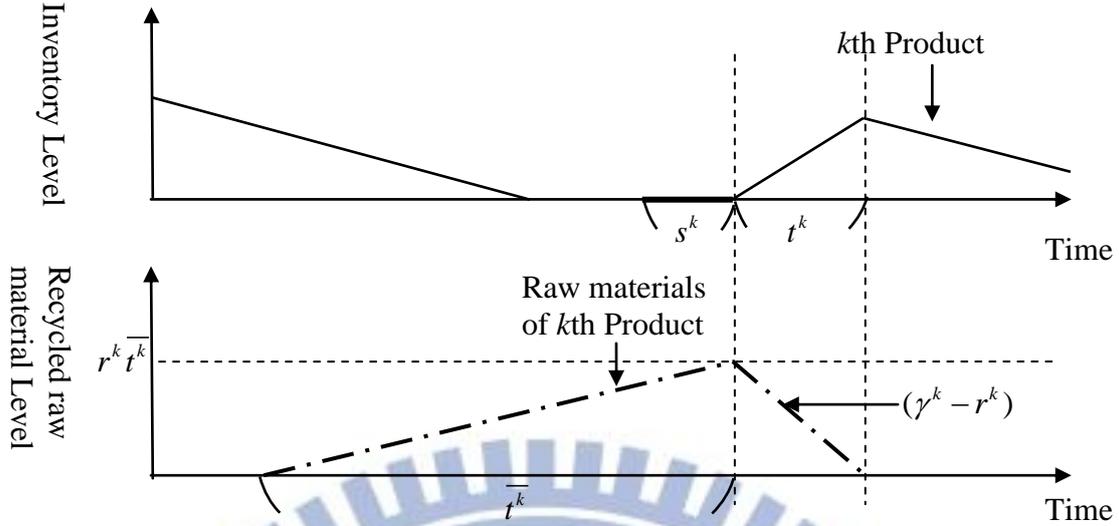


圖 3.2—本研究之 TVLS 存貨曲線圖

3.2.2.2 目標式說明

接著用求出之回收時間與生產時間的關係式改寫收入項及成本項公式，建構出本研究 TVLS 模式的目標式。

收入項說明

由於已知生產率、產品售價，使用生產時間表示銷貨收入為：

$$\sigma^k p^k t^k \quad (17)$$

成本項說明

成本項考慮整備成本、存貨成本、再生產品原料取得成本、再生產品原料存貨成本、再生產品缺貨成本，其中整備成本已知為 a^k 。

在 2.1.5 節已經介紹過，基本型 ELSP 模式的存貨成本公式，本研究的存貨成本公式亦同：

$$\frac{1}{2} h^k (p^k - r^k) \left(\frac{p^k}{r^k}\right) (t^k)^2 \quad (18)$$

再生產品原料取得成本為單位取得成本與回收量的線性關係，得到：

$$\alpha^k r^k (\bar{t}^k + t^k) + \beta^k = \alpha^k \gamma^k t^k + \beta^k \quad (19)$$

再生產品原料存貨成本可參考圖 3.2 之圖型，利用三角型面積公式，底乘以高除以 2 再乘上單位原料存貨成本 \bar{h}^k ，並改以生產時間表示，得到：

$$\frac{1}{2} \cdot \bar{h}^k \cdot (\bar{t}^k + t^k) \cdot r^k \bar{t}^k = \frac{1}{2} \cdot \bar{h}^k \cdot \left(\frac{(\gamma^k - r^k) t^k}{r^k} + t^k\right) \cdot r^k \bar{t}^k = \frac{1}{2} \bar{h}^k \gamma^k (\tau^k - 1) (t^k)^2 \quad (20)$$

再生產品缺貨成本以需求率乘以缺貨時間乘以產品售價得到，然而 TVLS

模式的缺貨時間並不容易定義，在此本研究採用排程週期 T 減去單項產品每次產品週期 v^k ，得到以下公式：

$$\sigma_i d_i \left[T - \sum_{k \in N_i} \left(\frac{p_i t^k}{d_i} \right) \right] \quad (21)$$

因為已經針對特定產品 i ，所以直接由上標 k 改為下標 i 即可。

以上為本研究之成本項說明，其中整備成本、存貨成本、再生產品原料取得成本和原料存貨成本依生產排程做加總，缺貨成本則是已經把次序考慮進公式，所以只針對產品別做加總。其中一般產品無再生產品原料取得成本和原料存貨成本。

TVLS 模式

根據上節成本項的推導，本研究之數學模式建構如下，：

$$\begin{aligned} \text{Max } TP(t^k) = & \frac{1}{T} \left\{ \sum_{k=1}^l \sigma^k p^k t^k - \left[\sum_{k=1}^l a^k + \sum_{k=1}^l \frac{1}{2} h^k (p^k - d^k) \left(\frac{p^k}{d^k} \right) (t^k)^2 \right. \right. \\ & + \sum_{i=n+1}^{2n} \sum_{k \in N_i} (\alpha^k \gamma^k t^k + \beta^k) + \sum_{i=n+1}^{2n} \sum_{k \in N_i} \frac{1}{2} \bar{h}^k \gamma^k (\tau^k - 1) (t^k)^2 \\ & \left. \left. + \sum_{i=n+1}^{2n} \sigma_i d_i \left(T - \frac{p_i}{d_i} \sum_{k \in N_i} t^k \right) \right] \right\} \end{aligned} \quad (22)$$

subject to

$$\sum_{k \in N_i} p_i t^k = d_i T \quad i = 1, \dots, n \quad (23)$$

$$\sum_{k \in L_o} (t^k + s^k + u^k) = \left(\frac{p^o}{d^o} \right) t^o \quad o \in N_i, i = 1, \dots, n \quad (24)$$

$$\sum_{k \in N_i} p_i t^k \leq d_i T \quad i = n+1, \dots, 2n \quad (25)$$

$$\sum_{k \in L_o} (t^k + s^k + u^k) \geq \left(\frac{p^o}{d^o} \right) t^o \quad o \in N_i, i = n+1, \dots, 2n \quad (26)$$

$$\sum_{k \in L_o} (t^k + s^k + u^k) + s^o \geq \bar{t}^o \quad o = 1, \dots, l \quad (27)$$

$$\sum_{k=1}^l (t^k + s^k + u^k) = T \quad (28)$$

限制式說明

限制式部分可以分為四個部分，分別是一般產品限制式、再生產品限制式、總週期限制式，以下逐步說明之。

1. 一般產品限制式：

公式(24)與(25)為一般產品限制式，為 TVLS 模式中固有之公式。公式(24)表示總週期裡之產品 i 的需求量必須要等於每次製造產品 i 的產量加總。公式(25)為本次生產產品 i 之產量，必須要足夠持續至下次生產該產品之前。兩者皆考慮產能限制，故說 TVLS 之解必定為可行解，不需要再進行可行性檢驗。

2. 再生產品限制式

公式(26)、(27)為再生產品限制式，兩者之概念與公式(24)和(25)相同，只是考慮再生產品允許缺貨，故將等於符號調整為大於與小於，即生產量可以不足滿足所有需求。

3. 回收時間限制式

公式(28)為回收時間限制式，在圖 3.2 已經呈現出再生產品原料之存貨圖，可以看到回收時間 \bar{t}^k 可用的最大時間長度為開始生產第 k 個產品之前的所有時間，直到上次生產相同產品，故使用 \bar{L}_o 進行 $\sum_{k \in \bar{L}_o} (t^k + s^k + u^k)$ 加總後再加上當次生產的整備時間 s^o ，作為 \bar{t}^o 之上界。

4. 總週期限制式

公式(29)將每個產品產品週期(整備時間、生產時間、閒置時間)按照生產順序加總後，等於總週期長度。

3.2.3 驗證本研究之 TVLS 目標式在給定排程週期 T 及排程 q 之情況下為凹性函數

本研究求解 TVLS 模式之情境是在給定排程週期 T 及排程 q 之情況下，故整個模式的決策變數就確定為生產時間 t^k 、閒置時間 u^k ，然而目標式中並沒有閒置時間 u^k 帶來的成本項，故僅須考慮生產時間 t^k 。現將目標式中獨立出一般產品除去缺貨成本的第 k 次利潤公式，令其為 $P_{regular}^k(t^k)$ ：

$$P_{regular}^k(t^k) = \sigma^k p^k t^k - a^k - \frac{1}{2} h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right) (t^k)^2 \quad (29)$$

再對 $P_{regular}^k(t^k)$ 對生產時間 t^k 做一次微分，得到：

$$\begin{aligned} \frac{dP_{regular}^k(t^k)}{dt^k} &= \sigma^k p^k - h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right) t^k = 0 \\ \Rightarrow t^k &= \frac{\sigma^k p^k}{h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right)} \end{aligned}$$

可知當 $t^k = \frac{\sigma^k p^k}{h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right)}$ 存在極值。對公式(45)做二次微分：

$$\frac{d^2 P_{regular}^k(t^k)}{d(t^k)^2} = -1 < 0 \quad (30)$$

由於公式(31)小於 0，故可說在給定排程週期 T 及及排程 q 之情況下，

$P_{regular}^k(t^k)$ 是 t^k 的凹性函數。用同樣的方法證明再生產品除去缺貨成本的第 k 次利潤公式 $P_{remanufacture}^k(t^k)$ ，發現 $P_{remanufacture}^k(t^k)$ 亦為 t^k 的 concave 函數：

$$\begin{aligned} P_{remanufacture}^k(t^k) &= \sigma^k p^k t^k - a^k - \frac{1}{2} h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right) (t^k)^2 - (\alpha^k \gamma^k t^k + \beta^k) - \frac{1}{2} \bar{h}^k \gamma^k (\tau^k - 1) (t^k)^2 \\ \frac{dP_{remanufacture}^k(t^k)}{dt^k} &= \sigma^k p^k - h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right) t^k - \alpha^k \gamma^k - \bar{h}^k \gamma^k (\tau^k - 1) (t^k) = 0 \\ \Rightarrow t^k &= \frac{\sigma^k p^k - \alpha^k \gamma^k}{h^k (p^k - d^k) \left(\frac{p^k}{d^k}\right) + \bar{h}^k \gamma^k (\tau^k - 1)} \\ \frac{d^2 P_{remanufacture}^k(t^k)}{d(t^k)^2} &= -1 - \bar{h}^k \gamma^k (\tau^k - 1) < 0 \end{aligned} \quad (31)$$

再來看到再生產品缺貨成本的部分，令 $P_{lost,i}(t^k)$ 為產品 i 的缺貨利潤(負)：

$$P_{lost,i}(t^k) = \sigma_i d_i \left[T - \frac{p_i}{d_i} \sum_{k \in N_i} (t^k) \right] = \sigma_i d_i T - \sigma_i p_i \cdot \sum_{k \in N_i} (t^k) \quad (32)$$

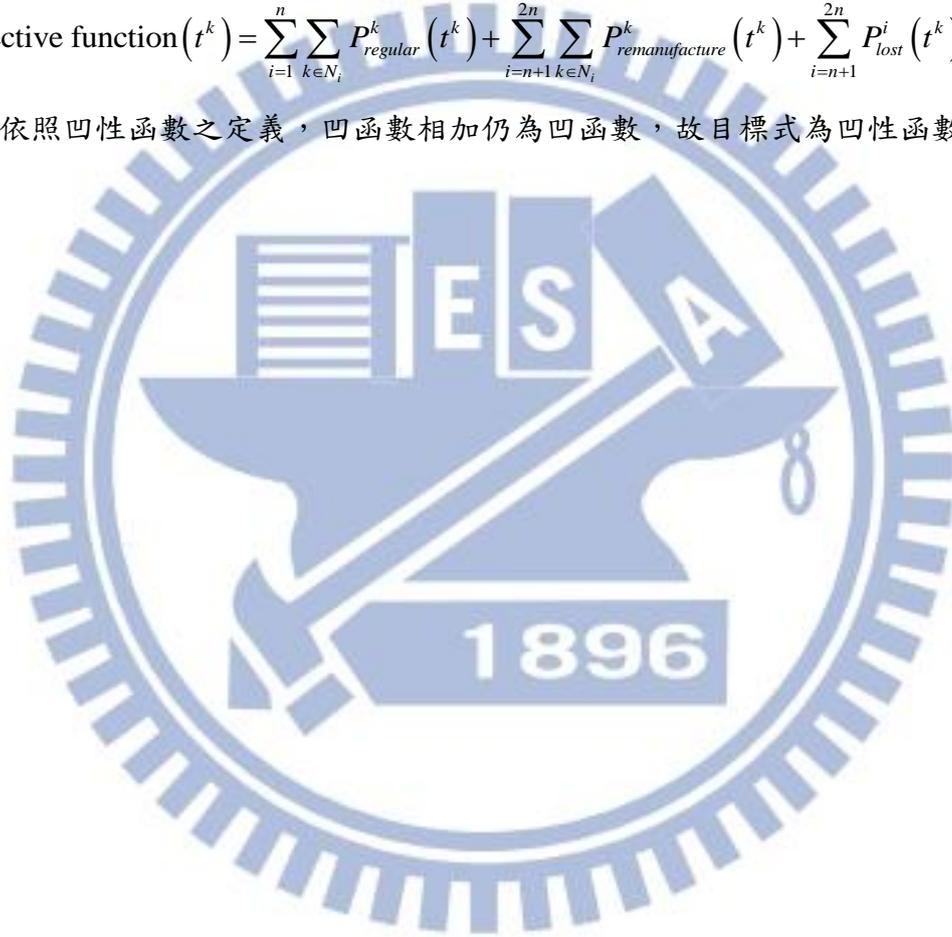
$P_{lost,i}(t^k)$ 為 $[0, \sigma_i d_i T]$ 之間的線性函數，亦為凹性函數。

而目標式可以表示成各次一般產品利潤 $P_{regular}^k(t^k)$ 、再生產品利潤

$P_{remanufacture}^k(t^k)$ 和缺貨利潤 $P_{lost,i}(t^k)$ 加總：

$$\text{objective function}(t^k) = \sum_{i=1}^n \sum_{k \in N_i} P_{regular}^k(t^k) + \sum_{i=n+1}^{2n} \sum_{k \in N_i} P_{remanufacture}^k(t^k) + \sum_{i=n+1}^{2n} P_{lost}^i(t^k) \quad (33)$$

依照凹性函數之定義，凹函數相加仍為凹函數，故目標式為凹性函數。



3.3 BP-based 模式

接著介紹本研究提出之 BP-based 模式，分別定義符號、說明模式，及驗證模式為生產時間的凹性函數，存在一極大值。

3.3.1 BP-based 模式符號定義

由於 BP-based 模式與 TVLS 模式之變數有部分相同，相同之部分就不在贅述，僅列出不同之部分。

決策變數：

B = 基本週期長度(days)

$K = \{k_1, k_2, \dots, k_{2n}\}$ 基本週期乘數，產品 i 之理想產品週期為 $T_i = k_i \cdot B$

θ_i = 產品 i 的生產時間縮放比例，此觀念為產品 i 的原料回收時間 \bar{t}_i 之長度會影響到回收原料量，轉而影響到是否能完成滿足整個週期需求的生產，而有一生產時間的縮放比例 $\theta_i \in [0, 1]$ ，調整實際的產品週期公式為

$\theta_i T_i = \theta_i (k_i \cdot B)$ ，於 3.3.2 節詳述

3.3.2 BP-based 模式建立

本節將建立在問題情境和假設之下的 BP-based 模式，先介紹回收時間與生產時間之關係、縮放比例 θ_i 之應用，再來逐步推導目標式之收入及成本項，最後進行限制式之說明。

3.3.2.1 實際回收時間與生產時間之關係

同 TVLS 模式，BP-based 模式也可以利用下頁圖 3.3 來探討生產時間與回收時間之關係，在傳統的 BP 模式中定義生產時間為 $\rho_i T_i$ ，然而本研究的情境中再生產品允許缺貨，故另定義產品 i 的實際生產時間為 t_i ，而回收時間為 \bar{t}_i ，原料按穩定回收率 r_i 不斷累積原料，到達開始生產時的回收量為 $r_i \bar{t}_i$ ，而開始進行生產時除了回收亦開始進行原料的消耗，故存貨減少率為 $(\gamma_i - r_i)$ ，即可推出 t_i 為回收量除以存貨減少率，得到：

$$t_i = \left(\frac{r_i}{\gamma_i - r_i} \right) \bar{t}_i \quad (33)$$

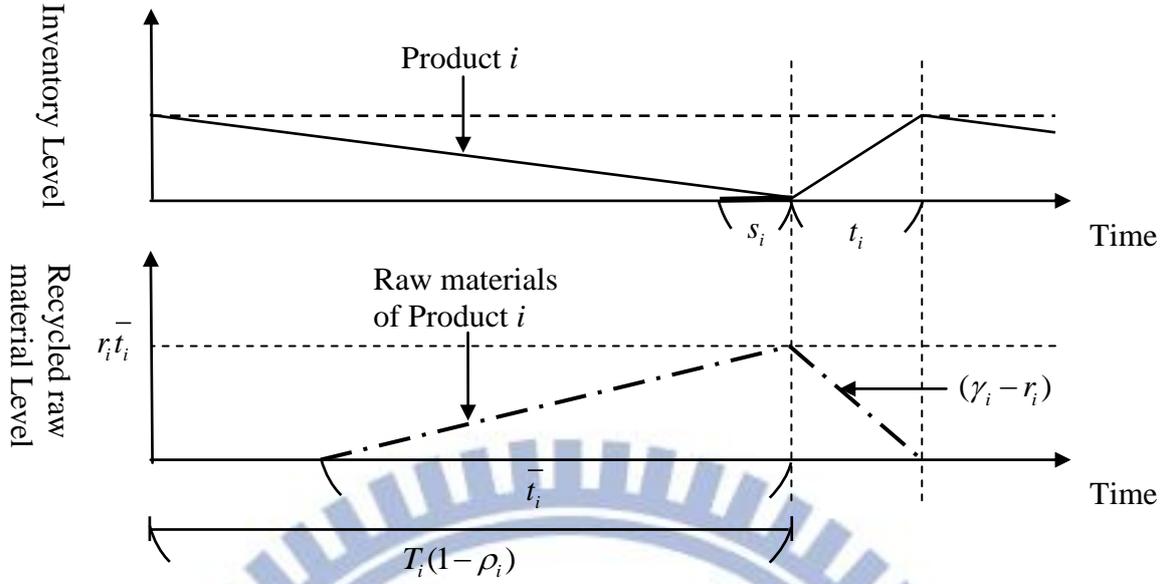


圖 3.3—本研究之 BP-based 存貨曲線圖

而從圖中也可觀察得知，因為每次的原料回收只供應當次的再生產品生產，所以回收時間的上界即是上次生產產品 i 的理想存貨消耗時間，長度即為 $T_i(1 - \rho_i)$ ，故可以寫出一回收時間之不等式：

$$\bar{t}_i \leq T_i(1 - \rho_i) \quad (33)$$

3.2.2.2 縮放比例 θ_i

本研究之再生產品需要考慮回收原料量是否足夠進行生產，因而影響到實際生產時間 t_i 未必等於理想生產時間 $\rho_i T_i$ ，但 t_i 會在證明目標式為凹性函數時產生困難，故於模式中增加一替代變數—「縮放比例」，以 θ_i 表示。其為實際生產時間理想時間 t_i 與理想生產時間 $\rho_i T_i$ 之比值：

$$\theta_i = \frac{t_i}{\rho_i T_i} \quad (33)$$

因此實際生產時間 t_i 與回收時間 \bar{t}_i 則可表示為：

$$t_i = \rho_i \theta_i T_i \quad (33)$$

$$\bar{t}_i = \frac{(\gamma_i - r_i) \rho_i \theta_i T_i}{r_i} \quad (33)$$

而調整理想生產時間 $\rho_i T_i$ 和理想產品週期 T_i 之結果如下方圖 3.4 所示。本模式將產品週期修改為 $\theta_i T_i$ ，表示以回收原料量生產後可服務到的週期長度，而剩餘的 $(1-\theta_i)T_i$ 即為缺貨時間，另外生產時間也從理想生產時間 $\rho_i T_i$ 調整為實際生產時間 $\rho_i(\theta_i T_i)$ 。

3.2.2.3 目標式說明

接著以新變數—縮放比例 θ_i 進行收入項及成本項公式之改寫及說明，建構出本研究 BP-based 模式的目標式。

收入項 R_i 說明

由於實際生產時間已經改寫為 θ_i ，則生產量可用生產時間乘以生產率得到 $d_i(\theta_i T_i)$ ，再乘上售價 σ_i 得到銷貨收入，除上週期 $(\theta_i T_i)$ 則為平均收入：

$$\frac{R_i}{\theta_i T_i} = \sigma_i d_i \quad (33)$$

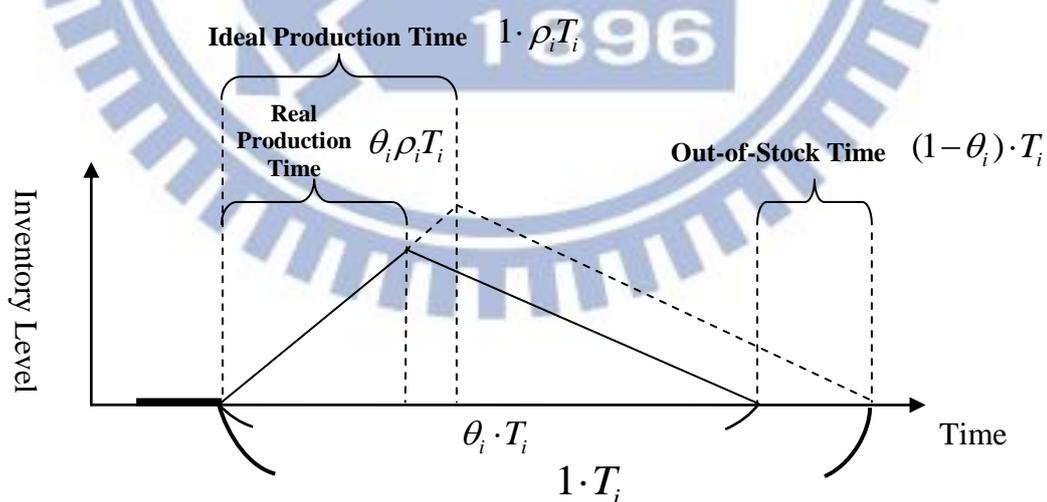


圖 3.4—以縮放比例 θ_i 調整之存貨曲線圖

成本項 C_i 說明

本研究之成本項一共討論整備成本、存貨成本、再生產品原料取得成本、再生產品原料存貨成本、再生產品缺貨成本五種成本，其中整備成本已知為 a_i ，剩下的部分將逐項說明。

存貨成本為傳統 ELSP 之成本項，原式為：

$$\frac{1}{2}h_i(p_i - d_i)\rho_i T_i \cdot T_i = \frac{1}{2}h_i d_i(1 - \rho_i)T_i^2$$

本研究以縮放比例 θ_i 調整為：

$$\frac{1}{2}h_i(p_i - d_i)\rho_i(\theta_i T_i) \cdot (\theta_i T_i) = \frac{1}{2}h_i d_i(1 - \rho_i)\theta_i^2 T_i^2 \quad (33)$$

再生產品原料取得成本、再生產品原料存貨成本、再生產品缺貨成本三者為本研究考慮之新成本項。依照研究假設，再生產品原料取得成本為單位取得成本與回收量的線性關係，又將回收時間改以縮放比例 θ_i 表示，得到：

$$\alpha_i r_i (\bar{t}_i + t_i) + \beta_i = \alpha_i r_i \cdot \left(\frac{(\gamma_i - r_i)\rho_i \theta_i T_i}{r_i} + \rho_i \theta_i T_i \right) + \beta_i = \alpha_i \gamma_i \rho_i \theta_i T_i + \beta_i \quad (33)$$

再生產品原料存貨成本則利用三角型面積公式，底為 $(\bar{t}_i + t_i)$ ，高為 $r_i \bar{t}_i$ ，兩者相乘再乘上單位原料存貨成本 \bar{h}_i 除以 2，一樣改以縮放比例 θ_i 表示，得到：

$$\begin{aligned} \frac{1}{2} \cdot \bar{h}_i \cdot (\bar{t}_i + t_i) \cdot r_i \bar{t}_i &= \frac{1}{2} \cdot \bar{h}_i \cdot \left(\frac{(\gamma_i - r_i)\rho_i \theta_i T_i}{r_i} + \rho_i \theta_i T_i \right) \cdot r_i \cdot \frac{(\gamma_i - r_i)\rho_i \theta_i T_i}{r_i} \\ &= \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i^2 T_i^2 \end{aligned} \quad (33)$$

其中 $\tau_i = \gamma_i / r_i$ 。

再生產品缺貨成本在本研究的假設下，只認列未賺得的銷貨收入，故可以用需求率乘以缺貨時間乘以產品售價得到：

$$\sigma_i d_i (1 - \theta_i) T_i \quad (33)$$

將以上成本項之結果加總，可以寫出總成本公式，除以週期得到平均成本：

$$\begin{aligned} \frac{C_i}{\theta_i T_i} &= \frac{a_i}{\theta_i T_i} + \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i T_i + \left[\alpha_i \gamma_i \rho_i + \frac{\beta_i}{\theta_i T_i} \right] \\ &\quad + \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \theta_i T_i + \sigma_i d_i \left(\frac{1}{\theta_i} - 1 \right) \end{aligned} \quad (33)$$

而對於一般產品不會發生的成本項(如原料取得成本、存貨成本，及缺貨成本)，則因為在最初給定之 $\alpha_i, \beta_i, \xi_i, \lambda_i, \bar{h}_i$ 為0，以及一般產品必定足夠供貨 $\theta_i=1$ ，所以該類成本項皆為0。

BP-based 模式

平均利潤 $\frac{P_i}{\theta_i T_i}$ 等於平均收入減去平均成本，可以得到產品 i 的平均利潤為：

$$\begin{aligned} \frac{P_i}{\theta_i T_i} &= \frac{R_i}{\theta_i T_i} - \frac{C_i}{\theta_i T_i} \\ &= \sigma_i d_i - \frac{a_i}{\theta_i T_i} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i T_i - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i T_i} \\ &\quad - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i T_i - \frac{\sigma_i d_i}{\theta_i} + \sigma_i d_i \end{aligned} \quad (33)$$

將全部的產品加總起來，即得到目標式，則模式可以寫成：

$$\begin{aligned} \max \quad TP(B, \{k_i\}) &= \sum_{i=1}^{2n} \frac{P_i}{\theta_i T_i} = \sum_{i=1}^{2n} \frac{P_i}{\theta_i k_i B} \\ &= \sum_{i=1}^{2n} \left\{ \sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} \right. \\ &\quad \left. - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} + \sigma_i d_i \right\} \end{aligned} \quad (33)$$

subject to

$$\begin{aligned} 0 &\leq \theta_i \leq 1, \quad \forall i \\ k_i &= 2^m, \quad m = \{0, 1, 2, \dots\} \end{aligned}$$

3.2.3 驗證本研究之 BP-based 目標式為凹性函數

在求解後續 BP-based 模式時，本研究將使用一個判斷缺料公式，進行縮放比例 θ_i 的設定，該假設之內容請參閱 4.1 節。故本研究是在給定縮放比例 θ_i 的情況下進行基本週期 B 的搜尋，為證明給定 θ_i 的 BP-based 模式存在一全域最大值，以下將逐步證明此前提下的模式為凹性函數 (concave function)。

首先從單項產品 i 的利潤公式推導起，在給定基本乘數組合

$\mathbf{K} = (k_1, k_2, \dots, k_{2n})$ 的情況下：

$$\begin{aligned} \frac{P_i(k_i, B)}{\theta_i k_i B} &= \sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} \\ &\quad - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} + \sigma_i d_i \end{aligned} \quad (33)$$

對該式做 B 的一次微分得到：

$$\begin{aligned} \frac{d \frac{P_i(k_i, B)}{\theta_i k_i B}}{dB} &= \frac{a_i}{\theta_i k_i B^2} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i + \frac{\beta_i}{\theta_i k_i B^2} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i = 0 \\ \Rightarrow \bar{B} &= \sqrt{\frac{2(a_i + \beta_i) / \theta_i k_i}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}} \end{aligned}$$

可以知道當 $\bar{B} = \sqrt{\frac{2(a_i + \beta_i) / \theta_i k_i}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}}$ 時存在極值。又對該式

做 B 的二次微分得到：

$$\frac{d \frac{P_i(B)}{\theta_i T_i}}{dB^2} = \frac{-2(a_i + \beta_i)}{\theta_i k_i B^3}$$

上式的 $(a_i + \beta_i) > 0$ ，且 θ_i, k_i, B 皆大於 0，故 $\frac{d \frac{P_i(B)}{\theta_i T_i}}{dB^2} < 0$ ，故平均利潤函

數為一基本週期 B 的凹性函數，在 $\bar{B} = \sqrt{\frac{2(a_i + \beta_i) / \theta_i k_i}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}}$ 時存在最大值。

而目標式為所有產品的平均利潤公式加總，因為凹函數之加總亦為凹函數，故在給定 θ_i 與固定基本乘數組合 $\mathbf{K} = (k_1, k_2, \dots, k_{2n})$ 之情況下，得證目標式亦為凹性函數。

第四章 模式求解

本章節將介紹本研究之模式求解方法，先介紹本研究之 TVLS 最佳解搜尋演算法的整體架構，再一一說明其細節。

TVLS 最佳解搜尋演算法從結構可以分為三部分，第一部分為 BP-based 模式求解，以獲得基本週期乘數 k_i 及基本週期 B 。本研究使用共同週期法、基因演算法、接合點搜尋法三種方法進行此步驟。第二部份是將 k_i 及 B 輸入進行排程之建構，本研究採用最長操作時間派工法，建構出排程 q 及週期 T 。最後一部分求解 TVLS 模式，並搭配鄰域搜尋法進行排程改善，且在固定的排程下對週期 T 進行進一步的搜尋，求得最佳解。

4.1 求解演算法架構介紹

本小節將介紹本研究之模式求解方法，如下圖 4.1 所示。在參數初始化之後，先使用演算法求解 BP-based 模式，此部分之關鍵點在於求出 PoT 策略下的基本週期乘數 k_i 及基本週期 B 。本研究分別使用共同週期法、基因演算法及接合點搜尋法三種方法進行求解，比較其求解品質及速率。

求得基本週期乘數 k_i 及基本週期 B 後，由於本研究提出之 TVLS 模式為給定生產週期 T 以及排程 q ，故要將基本週期乘數 k_i 及基本週期 B 透過轉換得到 T 及 q 的資訊，本研究採用最長操作時間派工法，建置基本週期下每一基期的生產排程，最後再將其依照順序整合為一條無分期的排程 q ，而生產週期 T 可用基本週期乘數中的最大值乘上基本週期 B 的長度來獲得。

將生產週期 T 以及排程 q 代入 TVLS 模式進行求解，並使用鄰域搜尋法進行排程的改善，且在固定的排程下對週期 T 進行進一步的搜尋，求出模式下的最佳解。計算使用之程式為 Matlab。

TVLS 最佳解搜尋演算法

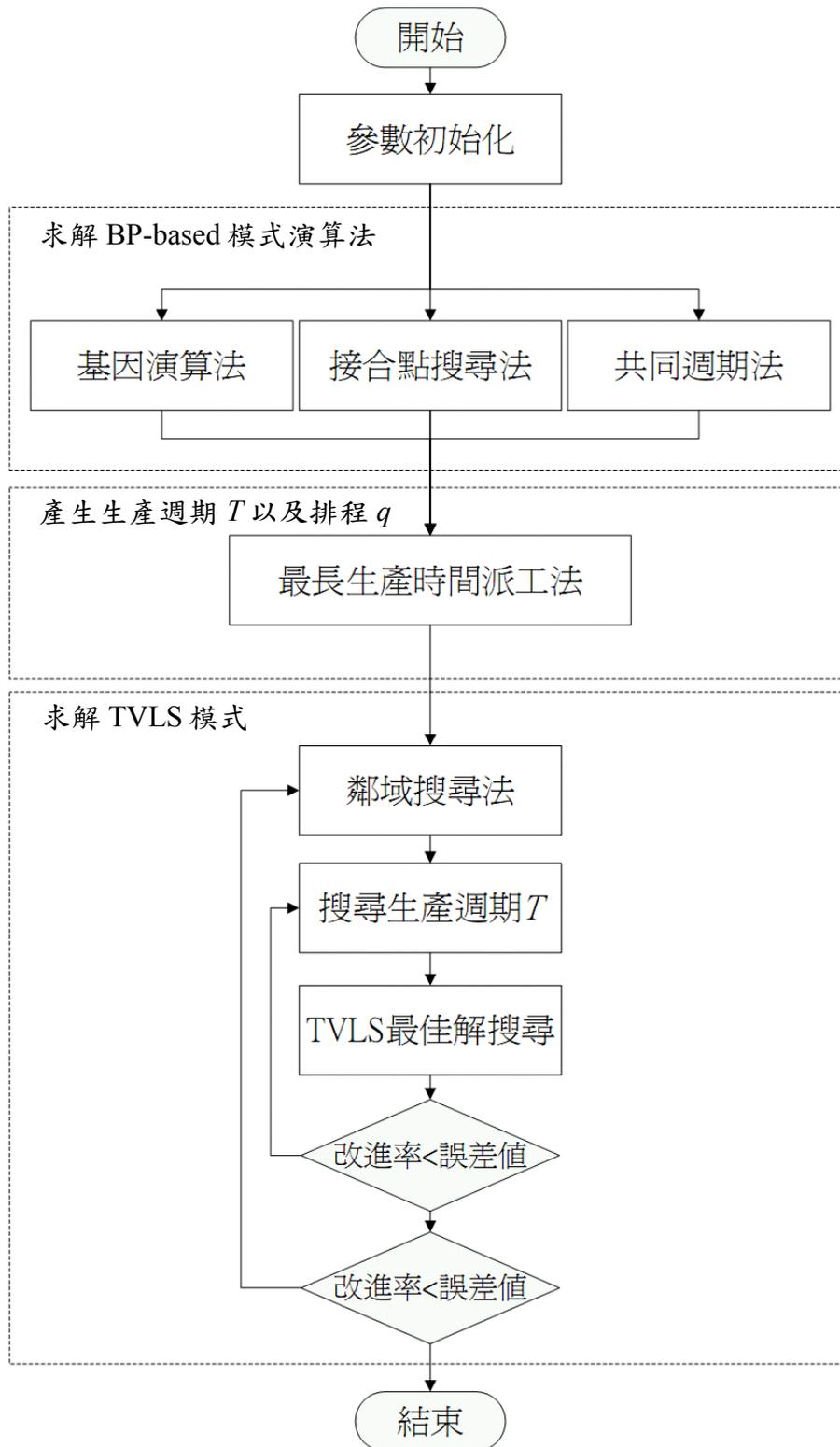


圖 4.1—程式演算流程圖

4.1.1 縮放比例 θ_i 之設定

在詳述該演算法之前，最初參數設定之部分，依照本研究之模式架構，必須在給定縮放比例 θ_i 的情況下，才能進行 BP-based 模式的求解。故須先透過「缺料條件式」判斷各品項是否發生缺料，於是對 θ_i 做了以下的推導：

缺料判斷式

在公式(18)已經說明，回收時間之上界為產品理想週期裡的消耗時間：

$$\bar{t}_i \leq T_i(1 - \rho_i)$$

又已知回收時間之公式可以表示如下：

$$\bar{t}_i = \frac{(\gamma_i - r_i)t_i}{r_i} = \frac{(\gamma_i - r_i)\theta_i \rho_i T_i}{r_i}$$

現今假設回收時間收集之產品足夠生產理想週期的需求(即 $\theta_i = 1$)，可得：

$$\begin{aligned} \frac{(\gamma_i - r_i)\rho_i T_i}{r_i} &\leq T_i(1 - \rho_i) \\ \Rightarrow \frac{(\gamma_i - r_i)\rho_i}{r_i} &\leq 1 - \rho_i \\ \Rightarrow \frac{\gamma_i}{r_i}\rho_i - \rho_i &\leq 1 - \rho_i \\ \Rightarrow \frac{\gamma_i}{r_i}\rho_i &\leq 1 \\ \Rightarrow \frac{d_i}{p_i} &\leq \frac{r_i}{\gamma_i} \end{aligned} \quad (34)$$

表示在給定參數下，當違反公式(49)時，即發生缺料。

4.1.2 縮放比例之上下界及最佳縮放比例 θ_i^*

此小節定義縮放比例 θ_i 之上下界，已知 $\theta_i \in (0, 1]$ ，但事實上回收時間會影響到 θ_i 之實際值。現假設回收時間用滿整個理想消耗時間，即 $\bar{t}_i = (1 - \rho_i)T_i$ ，又縮放比例為實際生產時間與理想生產時間之比值：

$$\theta_i = \frac{t_i}{PT_i} = \frac{\frac{r_i}{(\gamma_i - r_i)}\bar{t}_i}{\rho_i T_i} \quad (35)$$

將 $\bar{t}_i = (1 - \rho_i)T_i$ 代入公式(50)得：

$$\theta_i = \frac{\frac{r_i}{(\gamma_i - r_i)}(1 - \rho_i)T_i}{\rho_i T_i} = \frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)} \quad (36)$$

故重新將 θ_i 之範圍以更明確的方式定義：

$$\theta_i \in \left[0, \min \left(\frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)} \right) \right] \quad (37)$$

而在週期 $k_i B$ 給定之情況下，令平均利潤公式為 $f(\theta_i) = \frac{P_i(k_i B)}{\theta_i k_i B}$ ，將 $f(\theta_i)$

對 θ_i 做二次微分：

$$\frac{d^2 f(\theta_i)}{d\theta_i^2} = -2 \left(\frac{a_i + \beta_i}{\theta_i^3 k_i B} \right) - 2 \frac{\sigma_i d_i}{\theta_i^3} < 0$$

表示 $f(\theta_i)$ 為 θ_i 的 concave 函數，存在極大值，故用一次微分找出最佳值 θ_i^{opt} ：

$$\begin{aligned} \frac{df(\theta_i)}{d\theta_i} &= \frac{a_i + \beta_i}{\theta_i^2 k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) k_i B - \frac{1}{2} h_i \gamma_i (\tau_i - 1) \rho_i^2 k_i B + \frac{\sigma_i d_i}{\theta_i^2} = 0 \\ \Rightarrow \theta_i^{opt} &= \sqrt{\frac{2(a_i + \beta_i / k_i B + \sigma_i d_i)}{h_i d_i (1 - \rho_i) k_i B + h_i \gamma_i (\tau_i - 1) \rho_i^2 k_i B}} \end{aligned}$$

故將 θ_i^{opt} 考慮進 θ_i 之範圍裡，得到 θ_i^* ：

$$\theta_i^* = \begin{cases} \min \left(\frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)}, 1 \right) & \text{if } \theta_i^{opt} > \min \left(\frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)}, 1 \right) \\ \theta_i^{opt} & \text{if } \theta_i^{opt} \in \left[0, \min \left(\frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)}, 1 \right) \right] \\ 0 & \text{if } \theta_i^{opt} < 0 \end{cases} \quad (38)$$

然而在每個產品的生產週期 T_i 尚未決定之前， θ_i 仍然無法確定，故本研究做了一項較強的假設：

假設 1：於最初之參數設定中，所有產品皆生產到最大值，即

$$\theta_i = \frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)}$$

並在計算完生產週期 T_i 後反求 θ_i^* 進行驗證及調整。

4.2 TVLS 最佳解搜尋法

本研究之 TVLS 最佳解搜尋演算法從結構可以分為三部分，第一部分為 BP-based 模式求解，以獲得基本週期乘數 k_i 及基本週期 B 。針對此部分本研究使用共同週期法、基因演算法、接合點搜尋法三種方法進行求解，詳細內容將在 4.3、4.4 及 4.5 節進行敘述。

第二部份是將 k_i 及 B 輸入最長操作時間派工法，建構出排程 q 及週期 T 。最後一部分求解 TVLS 模式，以下分兩小節進行介紹。

4.2.1 建立排程

本研究使用最長操作時間派工法(Longest-Process-Time Dispatching, LPT)來建立排程，方法如下：

輸入資料：基本週期乘數 k_i 、基本週期 B

步驟一：找出 k_i 之最大值，建構出 $\max(k_i)$ 個的基期，每期的剩餘時間即為基本週期長度 B 。

步驟二：對每個產品計算出總生產時間(Total Process Time, TPT)，本研究之實際總生產時間計算公式如下：

$$RTPT_i = s_i + \rho_i(k_i B \cdot \theta_i)$$

步驟三：將產品以 k_i 由小到大排序，其中相同 k_i 之產品則按 $RTPT_i$ 由大到小排序，成為派工清單。

步驟四：按照派工清單之順序將產品放入基期，放入之頻率為 k_i ，若為 1 則每期皆要放入，若為 2 則每間隔一期放入一次，放入時選擇剩餘時間多者進行放入，而放入之後將剩餘時間減去該產品的 $RTPT_i$ ，並把該產品從派工清單中移除。

步驟五：持續放入直到派工清單為空。

但由於本方法會透過 TVLS 模式進行求解，且 TVLS 之解皆為可行解，故在此若產生剩餘時間為負的不可行解，仍能採用之。最後再將每個基期的生產順序按照期數順序接合成一條數列，即為輸入 TVLS 模式的排程 q ，而週期 $T = \max(k_i) \cdot B$ 即可得到。

4.2.2 求解 TVLS 模式

對於給定之排程順序下，未必在週期 $T = \max(k_i) \cdot B$ 可以找出最佳解，故週期 T 也需要進行搜尋，並且本研究透過設定搜尋之上下界來提升求解速率。然後使用 LPT 派工法建構出之排程 q 未必為最佳的排程順序，故需要透過演算法進行排程的改善，本研究使用之演算法為鄰域搜尋法。

4.2.2.1 排程固定下對 T 進行搜尋

已知在給定排程下，TVLS 模式為一個凹性函數，存在一個最大值，如圖 4.2 所示，雖然如圖中大約 $T=40$ 之後接近目標值持平，但仍然存在最大值的部分，再改動座標刻度後可以看出最佳解出現之位置，如下頁圖 4.3。

搜尋下界之設定為 $T^{LB} = \sum_{k=1}^l s^k$ ，其中 $\sum_{k=1}^l s^k$ 是給定排程下的整備時間加總，因為週期至少要大於該值，才會開始進行生產。

因為無法推導出合理且具管理意義之上界，故本研究僅使用數理方式求出上界，由於現有資訊僅有共同週期法週期 T_{CC} ，故每次取 T_{CC} 的整數倍 ($x \cdot T_{CC}, x \in N$) 作為測試週期 T 代入 TVLS 公式，運算後計算與前一次數值的變異量，若變異量為負，表示目標式曲線在該區間出現最大值後開始遞減，故可以作為二分搜尋法之上界。

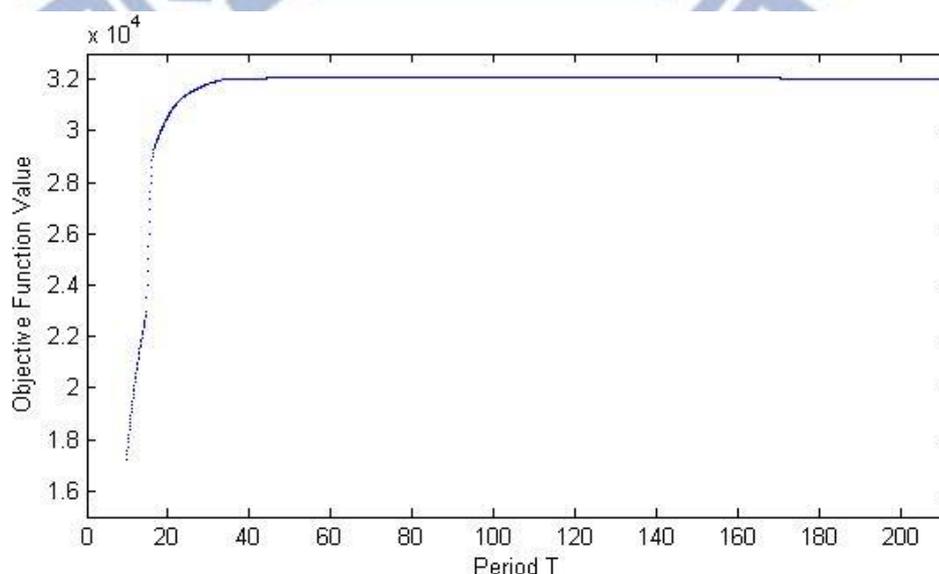


圖 4.2—固定排程下對 T 搜尋 TVLS 目標值(1)

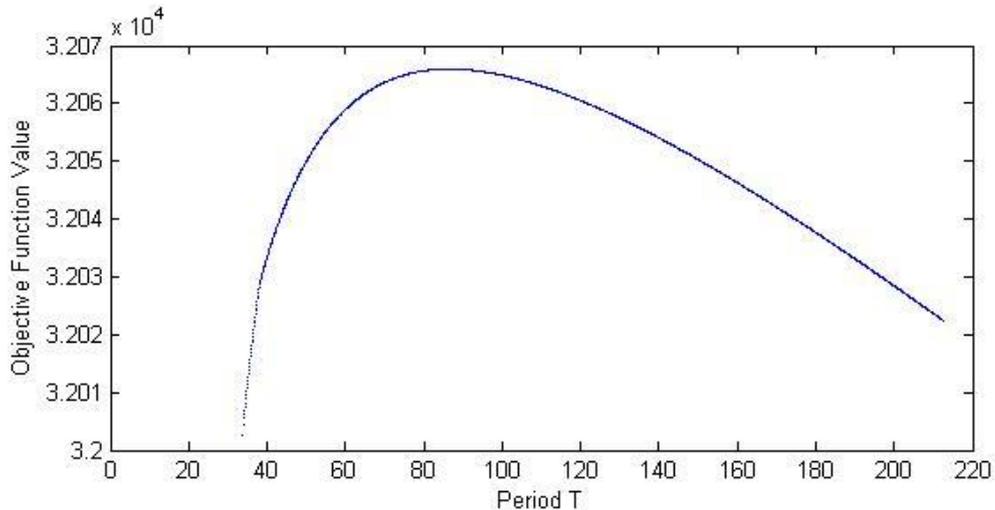


圖 4.3—固定排程下對 T 搜尋 TVLS 目標值(2)

但因為上下界的搜尋範圍還是過大，若使用微步搜尋法(Small-Step Search Procedure)會造成過久的搜尋時間，故本研究採用二分搜尋法(Bisection Search Algorithm)增加求解速率，步驟如下：

二分搜尋法(Bisection Search Algorithm)

步驟一：在最初的上下界中選擇兩者的中點 T ，計算該點目標值 $Obj(T)$ 。

步驟二：取 $T' = T + \Delta T$, ($\Delta T > 0$)，計算該點目標值 $Obj(T')$ ，並令

$\Delta Obj = Obj(T') - Obj(T)$ ，若 ΔObj 為正，表示最佳解在 T^{LB} 與 T 之間，則令 $T^{UB} = T$ ；若 ΔObj 為負，表示最佳解在 T^{UB} 與 T 之間，則令 $T^{LB} = T$ 。

步驟三：重複步驟一至二，若每期之目標值改進率小於誤差值，則停止。

該方法之求解時間會依據誤差值而有顯著的影響，考慮足夠的精確度與求解速率之間的權衡，本研究設定為 10^{-2} 至 10^{-3} 。

4.2.2.2 排程改善

在傳統 BP 模式裡，由於只定義將欲生產的產品放入基期後，剩餘時間大於 0 即為可行解，因此沒有生產順序上的疑慮。但 TVLS 模式已經無基期之概念，故即使是排程中的一個順序改變，也可能會造成整體生產時間與閒置

時間的重新調整，如圖 4.4 中產品 2 的 k_i 最小，所以每次都排在最前面，但其時還有其它的排序如 [4,1,2,9]、[1,2,9,4] 或其他可能，故對於 LPT 派工法生產出之排程，仍有最佳化之必要性。

本研究使用之排程改善方法為鄰域搜尋法(Neighborhood Search Algorithm, NS)，以下進行說明。

鄰域搜尋法

本研究提出的鄰域搜尋法(Neighborhood Search Algorithm, NS)之鄰域定義為每次隨機選定排程中的兩個位置，將先選的位置差入到後選位置之前，而後選位置及其後的排程將往後順延。但要注意若排程中有任一相鄰的兩個生產品項相同時，為不可行解。群體數設定為 20 個鄰域。

輸入資料：LPT 派工法產生之排程 q ，作為初始母體

步驟一：對母體建立鄰域群體，並計算各自之目標值(使用 TVLS-Search-T)。

步驟二：取鄰域群體中目標值最大者，與母體之目標值進行比較，若鄰域群體較母體為優，則取代成為新母體。

步驟三：重複步驟一至二，若每期之目標值改進率小於誤差值，則停止。

誤差值：設定與二分搜尋法相同。

BP _i	1 _i	2 _i	3 _i	4 _i	5 _i	6 _i	7 _i	8 _i
2 _i	2.409	2.409	2.409	2.409	2.409	2.409	2.409	2.409
4 _i	1.783		1.783		1.783		1.783	
7 _i		1.574		1.574		1.574		1.574
3 _i		1.640		1.640		1.640		1.640
1 _i	2.479				2.479			
5 _i		2.087				2.087		
9 _i	6.087							
8 _i			3.121					
6 _i		1.702						
10 _i			1.474					
σ	2419	27356	24810	273	241	2735	24	273

➔	排程 q :	2	4	1	9	2	7	3	5	6	2	4	8	10	2	7	3	2	4	1	2	7	3	5	2	4	2	7	3	
其它可能排序		4	2	1	9																									
		1	2	9	4																									

圖 4.4—使用 LPT 派工法產生的排程失誤

按照以上之方法及可以對排程 q 與週期 T 兩者皆進行最佳化，找出 TVLS 之最佳解。接著進行 TVLS 最佳解求解演算法第一部分的三個方法的介紹。

4.3 求解 BP-based 模式方法一：共同週期法

共同週期法在 ELSP 問題中是最常見的求解方法之一，多數的基本模式變型通常也是從共同週期法進行建模，才逐步提出基本週期法、延伸基本週期法的模式。共同概念週期法之概念請參閱本研究的 2.2.2 小節。本研究也提出 BP-based 模式下的共同週期求解法，共同週期的計算公式按照傳統方法，修改如下：

$$T_{CC} = \max \left(\frac{2 \sum_{n=1}^{2n} (a_i + \beta_i)}{\sqrt{\sum_{n=1}^{2n} [h_i d_i (1 - \rho_i) (\theta_i)^2 + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 (\theta_i)^2]}}, \frac{\sum_{i=1}^n S_i}{1 - \sum_{i=1}^n \rho_i} \right) \quad (39)$$

而由於共同週期法為每個產品生產一次，故可跳過 LPT 派工法，設定排程為 $q = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ ，週期 $T = T_{CC}$ ，代入 TVLS 最佳解求解演算法進行求解。

4.4 求解 BP-based 模式方法二：基因演算法

基因演算法(Genetic Algorithm, GA)由 John Holland 於 1975 年所提出，其過程是模擬大自然物競天擇的演化基製，來進行解的交配、汰換，從中淬煉出較佳之解。

基因演算法實際運算過程中，強調基因的轉變。首先將欲求解問題之解經過「編碼」而成為染色體，通常一段染色體為多段基因段的組合。接著利用模擬自然界演化過程，經過選擇(Selection)、交配(Crossover)，以及突變(Mutation)的遺傳演算機制，修改染色體的組成，進而使族群(Population)裡的染色體可求得之解能夠產生變化，進而搜尋問題的近似最佳解。

基因演算法具有下列幾點特性(Goldberg, 1989)：

1. 基因演算法的運算子(Operator)是直接作用在參數的編碼字串上，而非參數值本身。

2. 基因演算法為高維度搜尋，同時由參數空間中許多不同點開始搜尋而非點對點的單點搜尋法，因此可降低陷入區域最佳解之機率。
3. 基因演算法只需要適應值函數，沒有其他繁瑣的數學計算。
4. 基因演算法的搜尋方向是為「隨機(Probabilistic)」非「確定(Deterministic)」，因此可以適用於不同類型的問題上。

但由於演算法的隨機性，故不佳的參數設定仍可能會使解的品質不佳，或落於區域最佳解內。

接著介紹本研究對於基因演算法六項結構的設計：

1. 編碼及解碼
2. 產生初始族群
3. 適應值評估
4. 選擇與交配
5. 突變
6. 菁英族群

4.4.1 編碼及解碼

本研究將單一個基本週期乘數 k_i 設計為基因段，每個染色體則由所有產品的基因段結合在一起，即有 n 個產品，染色體就有 n 段基因。編碼採用二進位編碼，單段基因為3個位元，如[010]、[100]、[011]等，解碼方式則是使用二進位轉十進位，將基因轉換成數值 z_i ，乘數 k_i 即為 2^{z_i} ，故在3位元的編碼下，可以解碼出 2^0 、 2^1 至 2^8 ，而至於設定2的8次方為乘數上界則是因為該值等於256，相當於一年工作天240天。

4.4.2 產生初始族群

本研究採用隨機方式，建立若干條染色體作為初始母體。特別注意產生之染色體採用二冪次(PoT)策略，即乘數為 2^m ；其中 m 為包含零之正整數。

4.4.3 適應值評估

本研究使用模式目標值作為適應值，由染色體得到的乘數計算出基本週期 B 後，與乘數一起代入目標式，不過週期仍然要考慮產能限制帶來的下界

$$B^{LB} = \sum_{i=1}^n s_i / \left(1 - \sum_{i=1}^n \rho_i \right), \text{ 取其中大者。}$$

4.4.4 選擇與交配

本研究在選擇前使用「線性排序法」(Linear Ranking Method)得到每條染色體的被篩選機會，其公式計算如下：

$$prob = 2 - SP + 2 \cdot (SP - 1) \cdot \frac{rank}{Popsiz - 1} \quad (40)$$

其中 $rank$ 代表在目標式由大到小排序的順位數， $Popsiz$ 為族群大小。在「選擇」時先產生符合族群數量的隨機亂數，若隨機亂數落於某一個染色體的累積機率(染色體本身及目標式小於它的染色體 $prob$ 的加總)之區間即選入，故會選出符合族群數量的選擇族群，且裡面會有重複之染色體。

接著進入「交配」機制，對選擇族群內的每條染色體產生一個對應的隨機亂數，如果小於「交配率」，則選入為待交配族群，當待交配族群數量剩兩條以上時，則隨機選擇兩條染色體，使用單點交配，隨機決定染色體上的一個位置，在那個位置之後的基因段彼此互換。交配完之後該兩條染色體取代原族群內對應之兩條原染色體。

使用線性排序法的好處是可以避免使用基礎的輪盤法時，當目標式值差距不夠明顯，使得好染色體與差染色體被選入待交配群題的機率差不多，而使族群汰舊換新的速度不佳。

另外線性排序法使用「順序」作為選擇評價選擇機率的基礎，再使用 *Spanning (SP)* 進行效果的擴張，而擴張的效果過大的時候會造成高適應值染色體被選擇進行交配的機率過高，對於大部分的基因演算法設計未必是良好的結果。

4.4.5 突變

本研究採單點突變，在族群完成交配後，對族群內每條染色體的每個位元產生隨機亂數，若小於突變率，則由 0 轉為 1 或 1 轉為 0，將可能隨機改動解碼出來的乘數值，製造跳出區域最佳解的可能性。

4.4.6 菁英族群

基因演算法最容易為人詬病之處即為搜尋「廣度」夠，但「深度」不足，為了配合基因的組成，往往交配與突變未必能帶給族群提升的方向，故本研究使用菁英族群來確保每次的演化都會向著較佳之方向，增加求解的深度，即使未必能每次獲得全域最佳解，仍較容易停留在較佳的區域最佳解。

在此菁英族群的設計，在第一代的演化後，將整個族群內目標值較優的若干條選出，作為菁英族群，第二代以後，每當一代演化完，將本代的群體及上代的菁英族群中，按照優劣排序後，選出與起始族群數相同的最優族群，成為下一代之母代。

此方法可以確保新一代之成果不會退化，而且規律的淘汰一定數量的較差解，以維持群體整體的提升。

按照以上之設定，設定族群數為 40 條，菁英族群為 8 條，交配率 0.65，突變率 0.1 的情況下，進行 1000 個世代的演化，最後將最後一個世代的最佳解的 k_i 和 B 輸出，進入 LPT 派工法及後續的 TVLS 最佳解求解演算法進行求解。

4.5 求解 BP-based 模式方法三：接合點搜尋法

接合點搜尋法由 Yao(2001)提出，透過觀察不考慮產能限制下的二冪次乘數策略 ELSP 模式之性質，發現接合點(Junction Point)可以幫助求解 ELSP 問題，本研究參考並因應其中之概念修改出適用於本研究的接合點搜尋法，以下逐步介紹。

4.5.1 PoT 策略下無產能限制之 BP-based 問題

傳統考慮成本最小化的 ELSP BP-based 模式，在不考慮限制下的二冪次乘數策略的 ELSP 模式可以表示如下：

Unconstrained ELSP(PoT)

$$\text{Minimize } TC_{\text{PoT}}(B, \{k_i\}) = \sum_{i=1}^n \frac{a_i}{k_i B} + \frac{1}{2} h_i d_i (1 - \rho_i) k_i B$$

$$\text{subject to } k_i = 2^{m_i}; m_i \in \{0, 1, 2, \dots\}$$

而本研究的 BP-based 模式在給定 θ_i 後可以寫成：

Unconstrained ELSP_{RM}(PoT)

$$\text{Maximize } TP_{\text{PoT}}(B, \{k_i\}) = \sum_{i=1}^{2n} \left\{ 2\sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} - \frac{1}{2} h_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} \right\}$$

$$\text{subject to } k_i = 2^{m_i}; m_i \in \{0, 1, 2, \dots\}$$

若現在針對單一產品來考慮，可以寫出單一產品的利潤公式：

$$TP_{PoT,i}(B, k_i) = 2\sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} \quad (41)$$

而在不同的 B 下，能夠使 $TP_{PoT,i}(B, k_i)$ 最小的 k_i 也有所不同，故若要單一產品的定義最大化利潤函數，則可表示成下式：

$$\overline{TP}_{PoT,i}(B) = \max_{k_i} \{TP_{PoT,i}(B, k_i)\} \quad (42)$$

且藉由圖型可以看出，該函數為分段凹函數(piece-wise concave function)

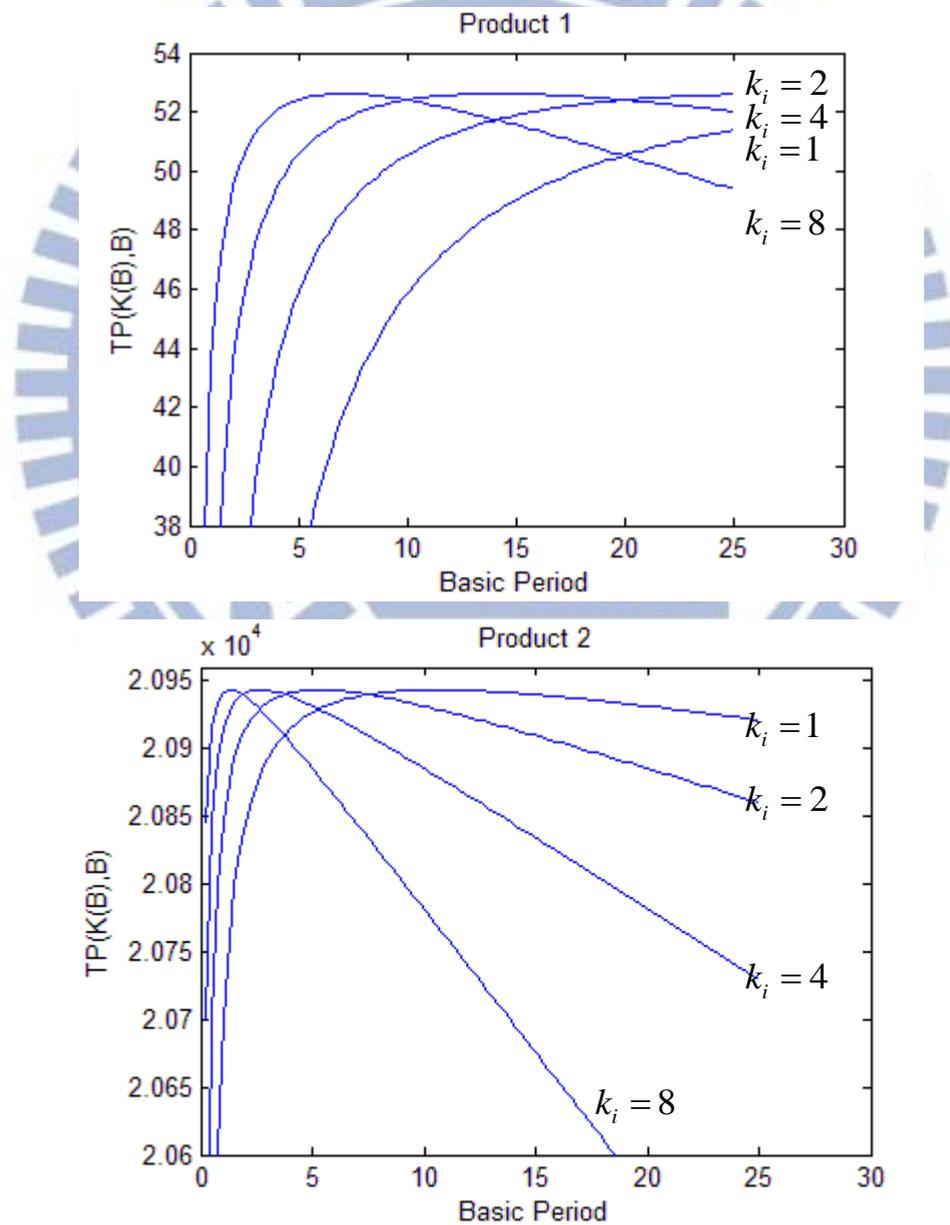


圖 4.5—產品 1 之最大化利潤函數圖型

而因為分段凹函數之加總亦為分段凹函數，故 $TP_{PoT}(B)$ 亦為分段凹函數。

現在定義 $TP_{PoT}^* \equiv \max_B \{TP_{PoT}(B)\}$ ，即 TP_{PoT}^* 為 $TP_{PoT}(B)$ 中的最大值。接著，

根據 PoT 策略對於基本週期乘數之假設，可以提出一理論：

推論 1：若 B_{PoT}^* 為 TP_{PoT}^* 之最佳基本週期，而 $K_{PoT}^*(B_{PoT}^*)$ 該為對應之最佳基本週期乘數，則可以透過將基本週期改為 $B_{PoT}^*/2$ ，得到 $2K_{PoT}^*(B_{PoT}^*)$ 。

證明：令 $K_{PoT}^*(B_{PoT}^*) = \{k_i^*\}$ ，則

$$TP_{PoT}^* = \sum_{i=1}^{2n} \left\{ 2\sigma_i d_i - \frac{a_i}{\theta_i k_i^* B_{PoT}^*} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i^* B_{PoT}^* - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i^* B_{PoT}^*} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i^* B_{PoT}^* - \frac{\sigma_i d_i}{\theta_i} \right\}$$

$$\Rightarrow TP_{PoT}^* = \sum_{i=1}^{2n} \left\{ 2\sigma_i d_i - \frac{a_i}{\theta_i (2k_i^*) (B_{PoT}^*/2)} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i (2k_i^*) (B_{PoT}^*/2) - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i (2k_i^*) (B_{PoT}^*/2)} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i (2k_i^*) (B_{PoT}^*/2) - \frac{\sigma_i d_i}{\theta_i} \right\}$$

故得證。而透過圖型，可以畫出下圖 4.6 的 $TP_{PoT}(B)$ 函數圖型，其中便可以看出函數的重複性，而另外所謂兩個凹函數相交之處，即為所謂的接合點(Junction Point)。

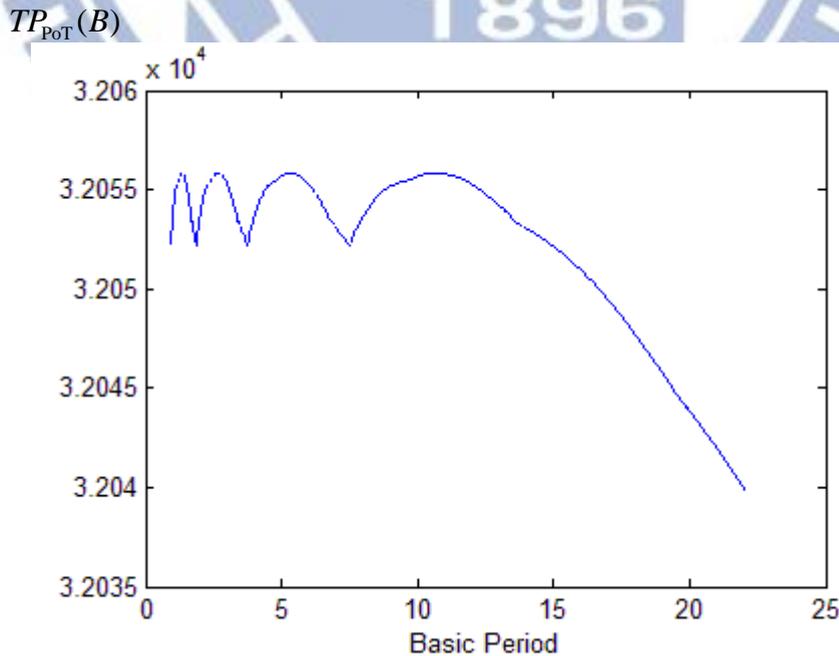


圖 4.6— $TP_{PoT}(B)$ 函數圖型

4.5.2 最佳利潤曲線之接合點及其性質

首先定義接合點：所謂接合點是發生在分段凹函數中，兩段凹函數的相接之處。而為何接合點如此重要，是因為在兩個接合點之中的區段是單純的凹函數，可以明確的指出，有區域最大值的存在，可能發生在一次微分等於零之處，或是發生在接合點該點上。而事實上每一組凹函數都是由一組乘數 $\{k_i\}$ 來決定，而相連的點代表著其餘的 k_i 值都相同，僅有一個 k_i 改變為 $2k_i$ 。

若是使用微步搜尋法的方式來求出所有的接合點，但顯然效率和品質都受到步伐大小的影響，所以本節修改 Yao(2001) 提出的接合點搜尋方法，用以協助求解。

首先接合點發生之處代表其餘的 k_i 值都相同，僅有一個 k_i 改變為 $2k_i$ 的情況，故可以寫出一等式：

$$2\sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} =$$

$$2\sigma_i d_i - \frac{a_i}{\theta_i (2k_i) B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i (2k_i) B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i (2k_i) B} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i (2k_i) B - \frac{\sigma_i d_i}{\theta_i}$$

同項相消後得到：

$$\frac{a_i + \beta_i}{\theta_i k_i B} = \left(h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i \right) B$$

$$\Rightarrow B^2 = \frac{\frac{a_i + \beta_i}{\theta_i k_i}}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}$$

$$\Rightarrow B = \sqrt{\frac{\frac{a_i + \beta_i}{\theta_i k_i}}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}}$$

$$\text{即當 } B = \sqrt{\frac{\frac{a_i + \beta_i}{\theta_i k_i}}{h_i d_i (1 - \rho_i) \theta_i k_i + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i}} \text{ 時，該處存在一接合點，令接}$$

合點為：

$$\delta_i(k_i) = \frac{1}{\theta_i k_i} \sqrt{\frac{a_i + \beta_i}{h_i d_i (1 - \rho_i) + \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2}} \quad (42)$$

故當代入不同的 k_i ，就可以將全部的接合點確定出來，在此介紹修改 Yao(2001) 後的接合點確定法(Junction Point Locating Procedure, JP Locating)。

接合點確定法

步驟一：設定最大冪次方數，本研究設定為 $8(2^8=256$ ，相當於一年工作天 240 天)，並設定 $m=0$ 。

步驟二：對每個產品計算 $\delta_i(k_i) = \frac{1}{\theta_i k_i} \sqrt{\frac{a_i + \beta_i}{h_i d_i (1 - \rho_i) + h_i \gamma_i (\tau_i - 1) \rho_i^2}}$ 。

步驟三： $m \leftarrow m+1$ ，重複步驟二直到 $m > v_i$ ，則結束。

步驟四：將其依照大小作出排序，並紀錄乘數發生變化的產品。

下表 4.1 為對範例題使用接合點確定法之結果，範例題之數據將在第五章說明。

表 4.1—接合點確定法之演算結果

No.	B 值	改變之產品	No.	B 值	改變之產品	No.	B 值	改變之產品	No.	B 值	改變之產品	No.	B 值	改變之產品
1	88.2469	8	26	7.4803	2	51	1.3789	8	76	0.2338	2	101	0.0389	1
2	77.7949	10	27	6.9679	4	52	1.2436	1	77	0.2177	4	102	0.0294	9
3	73.5018	6	28	6.9338	5	53	1.2155	10	78	0.2167	5	103	0.0292	2
4	55.4700	5	29	6.8150	7	54	1.1485	6	79	0.2130	7	104	0.0272	4
5	44.1235	8	30	6.2429	3	55	0.9401	9	80	0.1951	3	105	0.0266	7
6	39.7946	1	31	5.5154	8	56	0.9350	2	81	0.1724	8	106	0.0244	3
7	38.8975	10	32	4.9743	1	57	0.8710	4	82	0.1554	1	107	0.0146	2
8	36.7509	6	33	4.8622	10	58	0.8667	5	83	0.1519	10	108	0.0136	4
9	30.0828	9	34	4.5939	6	59	0.8519	7	84	0.1436	6	109	0.0133	7
10	27.7350	5	35	3.7603	9	60	0.7804	3	85	0.1175	9	110	0.0073	2
11	24.9714	3	36	3.7401	2	61	0.6894	8	86	0.1169	2			
12	22.0617	8	37	3.4840	4	62	0.6218	1	87	0.1089	4			
13	19.8973	1	38	3.4669	5	63	0.6078	10	88	0.1083	5			
14	19.4487	10	39	3.4075	7	64	0.5742	6	89	0.1065	7			
15	18.3755	6	40	3.1214	3	65	0.4700	9	90	0.0975	3			
16	15.0414	9	41	2.7577	8	66	0.4675	2	91	0.0862	8			
17	13.9359	4	42	2.4872	1	67	0.4355	4	92	0.0777	1			
18	13.8675	5	43	2.4311	10	68	0.4334	5	93	0.0760	10			
19	13.6299	7	44	2.2969	6	69	0.4259	7	94	0.0718	6			
20	12.4857	3	45	1.8802	9	70	0.3902	3	95	0.0588	9			
21	11.0309	8	46	1.8701	2	71	0.3447	8	96	0.0584	2			
22	9.9486	1	47	1.7420	4	72	0.3109	1	97	0.0544	4			
23	9.7244	10	48	1.7334	5	73	0.3039	10	98	0.0542	5			
24	9.1877	6	49	1.7037	7	74	0.2871	6	99	0.0532	7			
25	7.5207	9	50	1.5607	3	75	0.2350	9	100	0.0488	3			

4.5.3 PoT 策略下無產能限制之 BP-based 模式理論分析

在 4.5.1 節末提到， $TP_{\text{PoT}}(B)$ 的函數存在著重複性，本節將推導相關理論，來縮小模式求解的搜尋範圍，有效率的確定出 $TP_{\text{PoT}}(B)$ 函數的最佳解。

首先定義 w_{j+1}, w_j 為 $TP_{\text{PoT}}(B)$ 上兩相鄰接合點，且 $w_{j+1} < w_j$ ，而兩接合點

區段有最佳乘數組 $\{k_i\}$ ，定義 \check{B}_j 為：

$$\begin{aligned} \frac{dTP_{\text{PoT}}}{dB} &= \frac{d}{dB} \left\{ \sum_{i=1}^{2n} \left[2\sigma_i d_i - \frac{a_i}{\theta_i k_i B} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i B - \alpha_i \gamma_i \rho_i - \frac{\beta_i}{\theta_i k_i B} \right. \right. \\ &\quad \left. \left. - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i B - \frac{\sigma_i d_i}{\theta_i} \right] \right\} \\ &= \sum_{i=1}^{2n} \left[\frac{a_i}{\theta_i k_i B^2} - \frac{1}{2} h_i d_i (1 - \rho_i) \theta_i k_i + \frac{\beta_i}{\theta_i k_i B^2} - \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i \right] = 0 \\ \Rightarrow \check{B} &= \sqrt{\frac{2 \sum_{i=1}^{2n} \left(\frac{a_i + \beta_i}{\theta_i k_i} \right)}{\sum_{i=1}^{2n} \left[h_i d_i (1 - \rho_i) \theta_i k_i + \frac{1}{2} \bar{h}_i \gamma_i (\tau_i - 1) \rho_i^2 \theta_i k_i \right]}} \end{aligned} \quad (43)$$

如果 $\check{B}_j \in (w_{j+1}, w_j]$ ， TC_{PoT} 上的一個區域最小值將發生在 \check{B}_j 之處，否則區域最小值會在端點 w_{j+1} 上，即令 $\check{B}_j = w_{j+1}$ 。

再來針對發生重複之接合點位置提出推論：

推論 2 TP_{PoT} 在 \check{B}_j 有區域最大值，且 $\check{B}_{j+1} < \check{B}_j$ ，則 $\left[\frac{\check{B}_1}{2}, \check{B}_1 \right]$ 會不斷在

$\left[\frac{\check{B}_1}{2^{p+1}}, \frac{\check{B}_1}{2^p} \right]$ 裡不斷重複， $p=1, 2, \dots$ 。

證明 與推論 1 的證明理論相同。

藉由推論 2 之結果可以知道，在 TP_{PoT} 函數上，區域最大值會依照一種固定模式不斷出現，所以在最簡便的情況下，我們找到第一組最大值發生的 \check{B}_1 後，計算到 $\frac{\check{B}_1}{2}$ 所得到的所有數值，會依照一定的模式在剩下的區段重複出現，

所以只考慮 $\left[\frac{\check{B}_l}{2}, \check{B}_l \right]$ 的區段，就等於是考慮的全部 TP_{PoT} 函數可能會發生的區

域最大值，再從中找出最大之解，即為全域最佳解。

4.5.4 接合點搜尋演算法

在此先定義新的符號， $l_j(w_j)$ 是在 w_j 接合點要由 k 變成 $2k$ 的產品下標，即表 4.1 裡面的「改變的產品」。而 $\kappa(w_j)$ 是在 $(w_{j+1}, w_j]$ 區間的最佳乘數組，亦可寫成 $\kappa(w_j) = (\kappa(w_{j-1}) - \{k_{l_j(w_j)}\}) \cup (2k_{l_j(w_j)})$ 。接著為搜尋法之步驟：

步驟一：透過接合點定位法找出 $TP_{PoT}(B)$ 上所有的接合點。

步驟二：將接合點做排序，並標明出各點之要改變產品之下標 $l_j(w_j)$ 。

步驟三：設定起始 B 值為 $w_1 + \varepsilon$ ，其中 ε 為一極小之實數，令起始乘數組

$$K_{PoT}(w_1 + \varepsilon) = \{1, 1, \dots, 1\}，計算是否 $\check{B}(K_{PoT}(w_1 + \varepsilon)) = T_{cc} \in \psi_{\infty}$ ，$$

如是，則令 $l = 1$ ， $B_l = B_1 = T_{cc}$ ，並計算 $TP_{PoT}(K_{PoT}(w_1 + \varepsilon), T_{cc})$ ；

如否，則令 $l = 0$ ， $j = 1$ ，

$$\kappa(w_1) = (K_{PoT}(w_1 + \varepsilon) - \{k_{l_1(w_1)}\}) \cup (2k_{l_1(w_1)})。之後進入下一步驟。$$

步驟四：確認是否 $\check{B}(\kappa(w_j)) \in (w_{j+1}, w_j]$ ，如是，令 $l = l + 1$ ，

$\check{B}_l = \check{B}(\kappa(w_j))$ ，並計算 $TP_{PoT}(\kappa(w_j), B_l)$ ；如否，則進入下一步驟。

步驟五：令 $j = j + 1$ ，如果 $w_1 < \frac{\check{B}_l}{2}$ ，進入下一步驟；否則記錄

$$\kappa(w_j) = (\kappa(w_{j-1}) - \{k_{l_j(w_j)}\}) \cup (2k_{l_j(w_j)}) 並回到步驟四。$$

步驟六：從儲存的函數值中找出最大的值以及代入的 $\kappa(w_j)$ 和 \check{B}_l ，即

$$(K_{PoT}^*, B_{PoT}^*) = \arg \min \{TP_{PoT}(\kappa(w_j), \check{B}_l)\}，停止。$$

此法完成後即可得到無限制情況下的最佳的基本週期乘數 K_{PoT}^* 及最佳基本週期 B_{PoT}^* ，將其輸入 LPT 派工法及繼續進行後續的 TVLS 最佳解求解演算法進行求解。

第五章 範例說明與數值測試

本章於最初先修改 Elmaghraby(1978)的範例六題目，將各種產品重新配對成符合本研究假設的一般產品與再生產品之配對關係，並隨機產生新增之變數，按照第四章說明之求解演算法進行求解。接著由於本研究提出為新的問題，數值測式將針對給定之數值範圍隨機建構出應用於數值測試的題目，最後作各項參數的敏感度測試。

5.1 範例說明

本研究的範例說明是透過修改 Elmaghraby(1978)的 Example 6，將各種產品重新配對成符合本研究假設的一般產品與再生產品之配對關係，對於本研究的參數基本假設有以下3項：

1. 一般產品售價大於等於再生產品 $\sigma_i \geq \sigma_{i+n}$
2. 再生產品的生產率小於等於一般產品的生產率 $p_{i+n} \leq p_i$
3. 再生產品存貨成本大於等於再生產品原料成本 $h_{i+n} \geq \bar{h}_{i+n}$

並對增加的變數給予合理的數值，回收參數 ξ_{i+n} 統一設為 0.4，消耗參數 λ_{i+n} 統一設為 1.2，最後對產品的需求與生產略做調整，使稼動率等於 0.57，建構出如表 5.1 的範例說明參數表格。

表 5.1—範例說明之參數設定

i	d_i	p_i	ρ_i	a_i	s_i	h_i	\bar{h}_i	σ_i	r_i	γ_i	α_i	β_i	τ_i
1	210	7000	0.030	100	0.875	0.00031	0.000000	0.2675	0	0	0.0000	0.00	0.000
2	14000	108000	0.130	300	0.125	0.00044	0.000000	1.5000	0	0	0.0000	0.00	0.000
3	1700	30000	0.057	60	0.125	0.00006	0.000000	3.9000	0	0	0.0000	0.00	0.000
4	1200	13000	0.092	55	0.250	0.00026	0.000000	0.9000	0	0	0.0000	0.00	0.000
5	200	8000	0.025	60	0.750	0.00010	0.000000	2.7850	0	0	0.0000	0.00	0.000
6	300	5400	0.056	140	0.375	0.00040	0.000280	0.1070	84	6480	0.0321	5.35	77.143
7	4000	70000	0.057	70	0.375	0.00010	0.000070	1.0000	5600	84000	0.3000	50.00	15.000
8	420	18000	0.023	20	0.625	0.00003	0.000021	2.7000	680	21600	0.8100	135.00	31.765
9	390	8200	0.048	30	1.000	0.00009	0.000063	0.5000	480	9840	0.1500	25.00	20.500
10	370	7000	0.053	350	0.500	0.00123	0.000861	2.1000	80	8400	0.6300	105.00	105.000

5.1.1 判斷缺貨產品及設定縮放比例

利用 4.1 小節推導出之缺貨判斷公式(49)對再生產品(編號 6 至 10 號的產品)進行缺貨檢察：

$$\text{缺貨判斷式：滿足 } \frac{d_i}{p_i} > \frac{r_i}{\gamma_i} \text{ 則缺貨}$$

結果：由下方表 5.2 可知，產品編號 6 及 10 有缺貨情況，故按照 4.1 節所述將其假定生產到上限值，因此縮放比例設定 θ_i 為 $\frac{(p_i - d_i)r_i}{d_i(\gamma_i - r_i)}$ ，而未缺貨的產品可以服務滿完整的理想產品週期 T_i ，故 $\theta_i = 1$ 。

5.1.2 計算補貨週期乘數 k_i 與基本週期 B

在此透過本研究提出的三個方法進行補貨週期乘數 k_i 與基本週期 B 的求解。

方法一：共同週期法

按照 4.3 小節求出之共同週期法公式(54)，可以得到 $T_{CC} = 21.274$ ，且因為

$$\text{週期下界 } T_{CC}^{LB} = \frac{\sum_{i=1}^n s_i}{1 - \sum_{i=1}^n \rho_i} = 11.629$$

兩者取大者之後仍得到 $T_{CC} = 21.274$ 。並設

定排程 $q = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ ，TVLS 週期 $T = T_{CC} = 21.274$ ，進入 5.1.3 求解 TVLS。

方法二：基因演算法

按照 4.4 節之內容，設定族群數為 40 條，菁英族群為 8 條，交配率 0.65，突變率 0.1 的情況下，進行 1000 個世代的演化，最後將最後一個世代的最佳解的 k_i 和 B 輸出，得到結果如下頁表 5.3。

表 5.2—判斷缺貨及縮放比例之結果

i	ρ_i	r_i / γ_i	缺貨	θ_i
6	0.056	0.012963	T	0.22326
7	0.057	0.066667	F	1.00000
8	0.023	0.031481	F	1.00000
9	0.048	0.048780	F	1.00000
10	0.053	0.009524	T	0.17230

表 5.3—基因演算法執行 1000 個世代運算之結果

項目	結果
基本週期乘數 k_i	[4, 1, 4, 2, 8, 8, 2, 8, 4, 8]
基本週期 B	11.629 (天)
運算時間	9.828 (秒)

方法三：接合點搜尋法

由於接合點搜尋法忽略產能帶來的限制，找出無限制上下的最佳乘數及週期，所以找出的結果未必與基因演算法相同。

首先執行接合點確定法並進行排序，接著設定所有產品乘數皆為 1，從起始 B 值為 $(w_1 + \varepsilon) = (88.247 + \varepsilon) = 89$ 之處開始向下進行搜尋，隨著接合點的區段不斷修改乘數使 $\kappa(w_j) = (\kappa(w_{j-1}) - \{k_{i_j(w_j)}\}) \cup (2k_{i_j(w_j)})$ 。

計算結果如下頁表 5.4，其中右邊的欄位只記錄存在於各段接合點區間內的區域最大值和對應的基本週期，因為若接合點區間所對應的乘數計算出的 B 不存在於該區間內，則最大值必發生在端點上，但已知該端點值小於全域最大值，故可忽略。若第一個記錄的區域最大值表示為 \check{B}_1 ，則整搜尋範圍至 $B \leq \frac{\check{B}_1}{2}$ 為止。

表 5.4—接合點搜尋法執行過程

編號	接合點	$t_j(w_j)$	產品 k_i										\check{B}_l	$TP_{PoT}(\check{B}_l)$
			1	2	3	4	5	6	7	8	9	10		
0	89.000		1	1	1	1	1	1	1	1	1	1		
1	88.247	8	1	1	1	1	1	1	1	2	1	1		
2	77.795	10	1	1	1	1	1	1	1	2	1	2		
3	73.502	6	1	1	1	1	1	2	1	2	1	2		
4	55.470	5	1	1	1	1	2	2	1	2	1	2		
5	44.124	8	1	1	1	1	2	2	1	4	1	2		
6	39.795	1	2	1	1	1	2	2	1	4	1	2		
7	38.898	10	2	1	1	1	2	2	1	4	1	4		
8	36.751	6	2	1	1	1	2	4	1	4	1	4		
9	30.083	9	2	1	1	1	2	4	1	4	2	4		
10	27.735	5	2	1	1	1	4	4	1	4	2	4		
11	24.971	3	2	1	2	1	4	4	1	4	2	4		
12	22.062	8	2	1	2	1	4	4	1	8	2	4		
13	19.897	1	4	1	2	1	4	4	1	8	2	4		
14	19.449	10	4	1	2	1	4	4	1	8	2	8		
15	18.376	6	4	1	2	1	4	8	1	8	2	8		
16	15.041	9	4	1	2	1	4	8	1	8	4	8		
17	13.936	4	4	1	2	2	4	8	1	8	4	8		
18	13.868	5	4	1	2	2	8	8	1	8	4	8		
19	13.630	7	4	1	2	2	8	8	2	8	4	8		
20	12.486	3	4	1	4	2	8	8	2	8	4	8		
21	11.031	8	4	1	4	2	8	8	2	16	4	8	10.659	32055.842
22	9.949	1	8	1	4	2	8	8	2	16	4	8		
23	9.724	10	8	1	4	2	8	8	2	16	4	16		
24	9.188	6	8	1	4	2	8	16	2	16	4	16		
25	7.521	9	8	1	4	2	8	16	2	16	8	16		
26	7.480	2	8	2	4	2	8	16	2	16	8	16		
27	6.968	4	8	2	4	4	8	16	2	16	8	16		
28	6.934	5	8	2	4	4	16	16	2	16	8	16		
29	6.815	7	8	2	4	4	16	16	4	16	8	16		
30	6.243	3	8	2	8	4	16	16	4	16	8	16		
31	5.515	8	8	2	8	4	16	16	4	32	8	16	5.329	32055.842
32	4.974	1	16	2	8	4	16	16	4	32	8	16		

由表的結果可以得知出現區域最大值只有在乘數 $k_i = [4, 1, 4, 2, 8, 8, 2, 16, 4, 8]$ 時，區域最大值發生點為 $\hat{B}_1 = 10.659$ ，對應的區域最大值 $TP_{PoT}(\hat{B}_1) = 32055.842$ 。而計算的範圍至 $4.974 \leq \frac{10.659}{2} = 5.329$ 為止，搜尋結果可以畫出圖 5.1 的搜尋圖。最後將求得的最佳 k_i 和 B 輸出，整理於表 5.5。

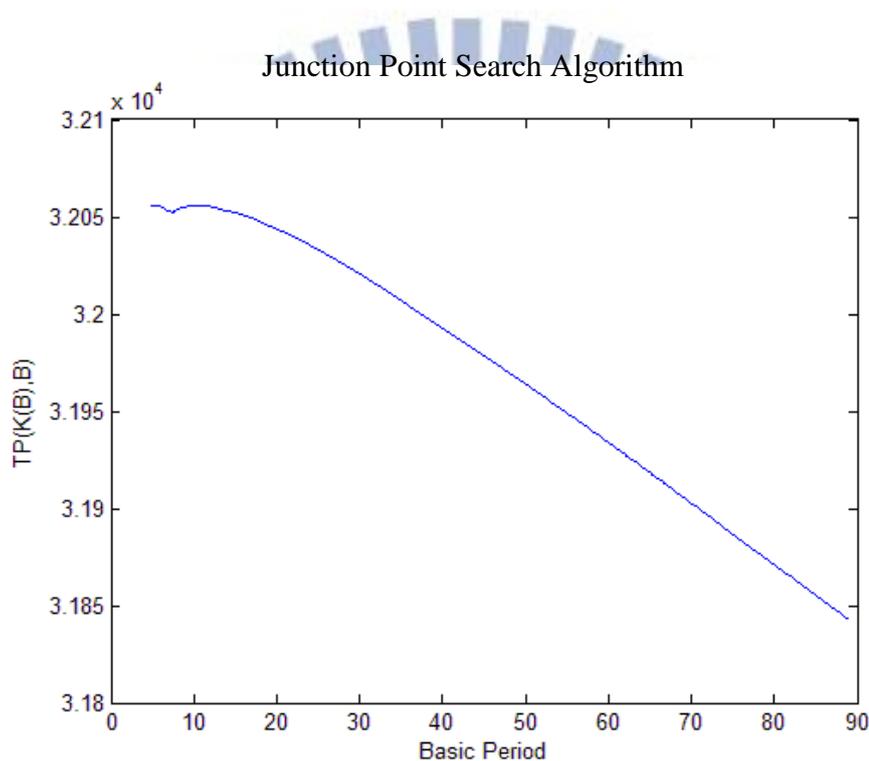


圖 4.7—接合點搜尋法執行過程

表 5.5—接合點搜尋法之結果

項目	結果
基本週期乘數 k_i	[4, 1, 4, 2, 8, 8, 2, 16, 4, 8]
基本週期 B	10.659 (天)
運算時間	0.078125 (秒)

5.1.3 建立排程(LPT 派工法)

經過三個方法求取 k_i 及 B 後，除了共同週期法已經給定排程 $q = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ 及設定 $T = T_{CC}$ ，其於兩者要將 k_i 及 B 轉換為 TVLS 模式中的排程與週期 T ，故將所有產品以 k_i 由小到大排序，其中相同 k_i 之產品則按實際生產時間 $RTPT_i$ 由大到小排序，成為派工清單，依序填入各基期，最後將所有基期之排程整併為一條排程序列，得到 TVLS 模式中排程 q ，而週期 $T = \max(k_i) \cdot B$ 。

將結果連同共同週期法整合呈現如下表 5.6，CC 表示共同週期法，GA 表示基因演算法，JP 表示接合點搜尋法。其中 CC 排程長度為 10 個產品，週期長度為 21.274 天；GA 排程長度為 26 個產品，週期長度 93.035 天；JP 排程長度為 51 個產品，週期長度為 170.541 天。

獲得以上資訊之後，即可以代入求解 TVLS。

5.1.4 求解 TVLS 模式

求解 TVLS 模式時，為了呈現(1)對 T 搜尋，及(2)改善排程的效果，將分別比較三種情況：

1. 直接使用排程及週期求解 TVLS 模式
2. 在固定排程下求解對 T 搜尋的 TVLS 模式
3. 使用鄰域搜尋法改善排程並求解對 T 搜尋的 TVLS 模式
並以同一情況下的共同週期法作為基準，比較改善之比例。

表 5.6—三種方法之排程及週期結果

方法	排程 q	週期 T
CC	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	21.274
GA	[2, 7, 9, 3, 1, 2, 4, 5, 8, 6, 2, 7, 10, 2, 4, 2, 7, 9, 3, 1, 2, 4, 2, 7, 2, 4]	93.035 (天)
JP	[2, 7, 9, 3, 6, 2, 4, 1, 5, 10, 2, 7, 8, 2, 4, 2, 7, 9, 3, 2, 4, 1, 2, 7, 2, 4, 2, 7, 9, 3, 6, 2, 4, 1, 5, 10, 2, 7, 2, 4, 2, 7, 9, 3, 2, 4, 1, 2, 7, 2, 4]	170.541(天)

情況一：直接使用排程及週期求解 TVLS 模式

在求解 TVLS 模式時，本研究使用 Matlab 的目標式最小化搜尋功能：
fmincon，在變數數量給定之下，該搜尋功能有能力處理限制式為線性或非線性、
目標式為線性或非線性的數學問題，而本研究再給定排程及週期下的 TVLS 問
題，屬於一種限制式為線性、目標式為非線性的數學問題。

在此介紹本研究使用到的 fmincon 搜尋功能，功能輸入如下：

```
x= fmincon(@(x) objective_fun(x,input),x0,A,B,Aeq,Beq,LB,UB,nonlcon,option)
```

x：功能求解之自變數，以本題目而言是生產時間與閒置時間，若排程序列
有 l 個，則自變數因有 $2l$ 個。

@(x) objective_fun(x,input)：目標式，由於本研究變數較多，建議另設一
function 檔案儲存目標式。檔名後面的括號填入目標式的輸入參數，除
了自變數 x 外，還需填入其他輸入參數，如生產率、需求率等，最後再
目標式前方用 @(x) 將求解自變數標示出來。

x0：自變數初始值，Matlab 將逐步修改此初始值直至將問題最佳化。

A 及 B：A 及 B 要滿足 $Ax \leq B$ ，即線性小於等於限制式，B 為右手值，若
無則設 []。

Aeq 及 Beq：Aeq 及 Beq 要滿足 $Aeqx = Beq$ ，即線性等於限制式，Beq 為右
手值，若無則設 []。

LB：自變數下界，若無則設 []。本研究因為生產時間與閒置時間皆必須為
大於 0 的數，故填入對應變數數量的全 0 矩陣。

UB：自變數上界，若無則設 []。

nonlcon：非線性限制式，本研究限制式皆為線性，故填入 []。

option：使用者可以對程式加入額外的指示，主要為設定程式求解時的額外
資訊 optiset 所使用，一般常見有以下種：

1. Algorithm：選擇演算法，預測為 'trust-region-reflective'，還可以選
擇 'active set'、'interior-point' 等。本研究採用 interior-point 演算法。
2. Display：顯示求解資訊，'off' 為不顯示、'iter' 按遞迴數顯示、
{'final'} 只顯示運算結果、'notify' 目標式未收斂時顯示運算結果。
3. MaxIter：輸入正整數，可設定最大運算遞迴數。

4. TolFun：輸入正數，可設定目標式誤差容忍值，本研究設為 10^{-3} 。

5. TolX：輸入正數，可設定自變數誤差容忍值。

輸入方法如：option=optiset('Algorithm','active set','MaxIter',40)。

按照以上之說明，將目標式及限制式放入對應的資訊格後，即可以用此功能找出對應的生產時間與閒置時間，生產時間可以再運用公式換成回收時間，三種不同方法求出之 k_i 與 B 在情況一之計算結果呈現如下頁表 5.6、5.7 及 5.8，表 5.9 是三者之間的必較，可以看出以接合點搜尋法的結果找出的 TVLS 值最為優良，數值上是 32066.011，比共同週期法找出的 TVLS 結果好 0.128%。

表 5.7—共同週期法結果代入 TVLS 模式求解(情況一)

TVLS(CC)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	1	2	3	4	5	6	7	8	9	10
t^k	0.638	2.758	1.206	1.964	0.532	0.276	1.216	0.496	1.012	0.203
\bar{t}^k	0.000	0.000	0.000	0.000	0.000	20.998	17.019	15.271	19.730	21.071
u^k	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
目標值：32024.897 (\$)										

表 5.8—基因演算法結果代入 TVLS 模式求解(情況一)

TVLS(GA)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	1	2	4	5	8	6
t^k	1.507	1.329	2.212	2.636	1.395	1.507	2.155	2.326	2.171	1.206
\bar{t}^k	0.000	18.606	43.142	0.000	0.000	0.000	0.000	0.000	66.784	91.829
u^k	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	10	2	4	2	7	9	3	1
t^k	1.507	1.329	0.886	1.507	2.141	1.508	1.329	2.212	2.636	1.396
\bar{t}^k	0.000	18.606	92.149	0.000	0.000	0.000	18.603	43.143	0.000	0.000
u^k	0.039	3.432	3.432	0.163	7.442	0.042	0.002	0.002	0.003	0.003
順序 k	21	22	23	24	25	26				
產品 i	2	4	2	7	2	4				
t^k	1.508	2.141	1.508	1.329	1.507	2.151				
\bar{t}^k	0.000	0.000	0.000	18.613	0.000	0.000				
u^k	0.098	7.511	0.029	8.267	0.022	7.573				
目標值：32065.687 (\$)										

表 5.9—接合點搜尋法結果代入 TVLS 模式求解(情況一)

TVLS(JP)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	6	2	4	1	5	10
t^k	1.382	1.218	2.027	2.412	1.105	1.382	1.967	1.283	2.132	0.812
\bar{t}^k	0.000	17.046	39.520	0.000	84.174	0.000	0.000	0.000	0.000	84.464
u^k	0.030	0.232	0.118	0.071	0.068	0.557	0.030	0.000	0.000	0.000
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	8	2	4	2	7	9	3	2
t^k	1.382	1.218	3.979	1.382	1.963	1.382	1.218	2.025	2.419	1.382
\bar{t}^k	0.000	17.052	122.421	0.000	0.000	0.000	17.051	39.495	0.000	0.000
u^k	0.013	1.471	1.471	0.544	6.397	0.007	0.221	0.064	1.701	0.490
順序 k	21	22	23	24	25	26	27	28	29	30
產品 i	4	1	2	7	2	4	2	7	9	3
t^k	1.970	1.275	1.382	1.219	1.381	1.971	1.381	1.220	2.024	2.421
\bar{t}^k	0.000	0.000	0.000	17.072	0.000	0.000	0.000	17.075	39.470	0.000
u^k	0.207	4.084	0.000	7.560	0.514	6.412	0.025	0.151	0.164	0.083
順序 k	31	32	33	34	35	36	37	38	39	40
產品 i	6	2	4	1	5	10	2	7	2	4
t^k	1.105	1.381	1.968	1.284	2.132	0.812	1.381	1.218	1.382	1.964
\bar{t}^k	84.156	0.000	0.000	0.000	0.000	84.452	0.000	17.053	0.000	0.000
u^k	0.081	0.559	0.013	0.002	0.002	0.002	0.059	7.498	0.565	6.374
順序 k	41	42	43	44	45	46	47	48	49	50
產品 i	2	7	9	3	2	4	1	2	7	2
t^k	1.382	1.218	2.035	2.413	1.381	1.970	1.275	1.381	1.217	1.382
\bar{t}^k	0.000	17.048	39.681	0.000	0.000	0.000	0.000	0.000	17.035	0.000
u^k	0.058	0.051	0.317	1.561	0.528	0.204	4.049	0.053	7.506	0.557
順序 k	51									
產品 i	4									
t^k	1.968									
\bar{t}^k	0.000									
u^k	6.377									
目標值：32066.011 (\$)										

表 5.10—情況一比較表

	TVLS(GA)	TVLS(JP)
TVLS(CC)	0.127%	0.128%
TVLS(GA)		0.001%

註: 0.127=(VLS(GA)之目標值-TVLS(CC)之目標值)/TVLS(CC)之目標值

情況二：在固定排程下求解對 T 搜尋的 TVLS 模式

延續上一情況，對於在 BP-based 模式中以微分找極值方法求出基本週期轉換而來的週期 T 在 TVLS 模式中是否會讓其值最大化的，然後並沒有理論說明基本週期法與 TVLS 方法所求出的極值會發生在同一個大生產週期，故本研究透過二分搜尋法的方法進行改善。

首先先觀察共同週期法 TVLS 在情況一與情況二的比較，誤差值設定至 10^{-4} ，結果如表 5.10。可以看出，經過搜尋求出的週期與最初的週期有所不同，而目標式結果也確實有較佳，不過數值的差異已經小於小數點後第四位，且因為誤差值過小，造成運算時間較長，所以若是在一般情況下，建議設定誤差值為 10^{-3} 即可。

接著對基因演算法及接合點搜尋法之結果代入 TVLS-search-T 進行與情況一的比較，基因演算法 TVLS-search-T 的部分，誤差值設定為 10^{-3} ，結果呈現如下頁表 5.11，可以看到目標值有提升，而週期之值也有所縮小，運算次數 12 次，但因為排城較共同週期法長、變數數量多，故求解時間相對拉長。

表 5.11—共同週期法結果代入 TVLS-search-T 模式求解(情況二)

TVLS(CC)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	1	2	3	4	5	6	7	8	9	10
t^k	0.638	2.758	1.206	1.964	0.532	0.276	1.216	0.496	1.012	0.203
\bar{t}^k	0.000	0.000	0.000	0.000	0.000	20.998	17.019	15.271	19.730	21.071
u^k	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
目標值： 32024.897232 (\$)					週期 T ： 21.274 (天)					
TVLS-search-T(CC)										
產品 i	1	2	3	4	5	6	7	8	9	10
t^k	0.638	2.758	1.206	1.964	0.532	0.276	1.216	0.496	1.012	0.203
\bar{t}^k	0.000	0.000	0.000	0.000	0.000	20.998	17.019	15.271	19.730	21.071
u^k	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
二方法搜尋次數： 10 (次)			運算時間： 0.406 (秒)			目標值： 32024.897253 (\$)			週期 T ： 21.268 (天)	

表 5.12—基因演算法結果代入 TVLS-search-T 模式求解(情況二)

TVLS(GA)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	1	2	4	5	8	6
t^k	1.507	1.329	2.212	2.636	1.395	1.507	2.155	2.326	2.171	1.206
\bar{t}^k	0.000	18.606	43.142	0.000	0.000	0.000	0.000	0.000	66.784	91.829
u^k	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597	0.597
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	10	2	4	2	7	9	3	1
t^k	1.507	1.329	0.886	1.507	2.141	1.508	1.329	2.212	2.636	1.396
\bar{t}^k	0.000	18.606	92.149	0.000	0.000	0.000	18.603	43.143	0.000	0.000
u^k	0.039	3.432	3.432	0.163	7.442	0.042	0.002	0.002	0.003	0.003
順序 k	21	22	23	24	25	26				
產品 i	2	4	2	7	2	4				
t^k	1.508	2.141	1.508	1.329	1.507	2.151				
\bar{t}^k	0.000	0.000	0.000	18.613	0.000	0.000				
u^k	0.098	7.511	0.029	8.267	0.022	7.573				
目標值： 32065.687 (\$)					週期 T ： 93.034 (天)					
TVLS-search-T (GA)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	1	2	4	5	8	6
t^k	1.4049	1.22627	2.02229	2.40914	1.2752	1.3825	1.95247	2.12295	1.9814	1.1008
\bar{t}^k	0.000	17.168	39.435	0.000	0.000	0.000	0.000	0.000	60.958	83.817
u^k	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	10	2	4	2	7	9	3	1
t^k	1.362	1.204	0.809	1.365	1.969	1.401	1.216	2.016	2.403	1.272
\bar{t}^k	0.000	16.857	84.109	0.000	0.000	0.000	17.018	39.322	0.000	0.000
u^k	0.000	3.065	3.065	0.000	6.818	0.000	0.000	0.000	0.000	0.000
順序 k	21	22	23	24	25	26				
產品 i	2	4	2	7	2	4				
t^k	1.361	1.965	1.365	1.207	1.367	1.952				
\bar{t}^k	0.000	0.000	0.000	16.892	0.000	0.000				
u^k	0.000	6.799	0.000	7.461	0.249	6.599				
二方法搜尋次數： 12 (次)			運算時間： 1.125(秒)			目標值： 32065.939 (\$)			週期 T ： 84.918 (天)	

但接合點搜尋法之結果代入 TVLS-search-T 時發生較意外的情況，誤差值設定為 10^{-3} ，則有可能找出小於原本接合點搜尋法 TVLS 之解，經過嘗試後將誤差值設定為 10^{-5} ，結果僅得到相同解，呈現如表 5.12。

表 5.13—接合點搜尋法結果代入 TVLS 模式求解(情況二) (上半)

TVLS(JP)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	6	2	4	1	5	10
t^k	1.382	1.218	2.027	2.412	1.105	1.382	1.967	1.283	2.132	0.812
\bar{t}^k	0.000	17.046	39.520	0.000	84.174	0.000	0.000	0.000	0.000	84.464
u^k	0.030	0.232	0.118	0.071	0.068	0.557	0.030	0.000	0.000	0.000
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	8	2	4	2	7	9	3	2
t^k	1.382	1.218	3.979	1.382	1.963	1.382	1.218	2.025	2.419	1.382
\bar{t}^k	0.000	17.052	122.421	0.000	0.000	0.000	17.051	39.495	0.000	0.000
u^k	0.013	1.471	1.471	0.544	6.397	0.007	0.221	0.064	1.701	0.490
順序 k	21	22	23	24	25	26	27	28	29	30
產品 i	4	1	2	7	2	4	2	7	9	3
t^k	1.970	1.275	1.382	1.219	1.381	1.971	1.381	1.220	2.024	2.421
\bar{t}^k	0.000	0.000	0.000	17.072	0.000	0.000	0.000	17.075	39.470	0.000
u^k	0.207	4.084	0.000	7.560	0.514	6.412	0.025	0.151	0.164	0.083
順序 k	31	32	33	34	35	36	37	38	39	40
產品 i	6	2	4	1	5	10	2	7	2	4
t^k	1.105	1.381	1.968	1.284	2.132	0.812	1.381	1.218	1.382	1.964
\bar{t}^k	84.156	0.000	0.000	0.000	0.000	84.452	0.000	17.053	0.000	0.000
u^k	0.081	0.559	0.013	0.002	0.002	0.002	0.059	7.498	0.565	6.374
順序 k	41	42	43	44	45	46	47	48	49	50
產品 i	2	7	9	3	2	4	1	2	7	2
t^k	1.382	1.218	2.035	2.413	1.381	1.970	1.275	1.381	1.217	1.382
\bar{t}^k	0.000	17.048	39.681	0.000	0.000	0.000	0.000	0.000	17.035	0.000
u^k	0.058	0.051	0.317	1.561	0.528	0.204	4.049	0.053	7.506	0.557
順序 k	51									
產品 i	4									
t^k	1.968									
\bar{t}^k	0.000									
u^k	6.377									
目標值： 32066.011 (\$)					週期 T ： 170.541 (天)					

表 5.13—接合點搜尋法結果代入 TVLS 模式求解(情況二) (下半)

TVLS-search-T (JP)										
順序 k	1	2	3	4	5	6	7	8	9	10
產品 i	2	7	9	3	6	2	4	1	5	10
t^k	1.382	1.218	2.027	2.412	1.105	1.382	1.967	1.283	2.132	0.812
$\overline{t^k}$	0.000	17.046	39.520	0.000	84.174	0.000	0.000	0.000	0.000	84.464
u^k	0.030	0.232	0.118	0.071	0.068	0.557	0.030	0.000	0.000	0.000
順序 k	11	12	13	14	15	16	17	18	19	20
產品 i	2	7	8	2	4	2	7	9	3	2
t^k	1.382	1.218	3.979	1.382	1.963	1.382	1.218	2.025	2.419	1.382
$\overline{t^k}$	0.000	17.052	122.421	0.000	0.000	0.000	17.051	39.495	0.000	0.000
u^k	0.013	1.471	1.471	0.544	6.397	0.007	0.221	0.064	1.701	0.490
順序 k	21	22	23	24	25	26	27	28	29	30
產品 i	4	1	2	7	2	4	2	7	9	3
t^k	1.970	1.275	1.382	1.219	1.381	1.971	1.381	1.220	2.024	2.421
$\overline{t^k}$	0.000	0.000	0.000	17.072	0.000	0.000	0.000	17.075	39.470	0.000
u^k	0.207	4.084	0.000	7.560	0.514	6.412	0.025	0.151	0.164	0.083
順序 k	31	32	33	34	35	36	37	38	39	40
產品 i	6	2	4	1	5	10	2	7	2	4
t^k	1.105	1.381	1.968	1.284	2.132	0.812	1.381	1.218	1.382	1.964
$\overline{t^k}$	84.156	0.000	0.000	0.000	0.000	84.452	0.000	17.053	0.000	0.000
u^k	0.081	0.559	0.013	0.002	0.002	0.002	0.059	7.498	0.565	6.374
順序 k	41	42	43	44	45	46	47	48	49	50
產品 i	2	7	9	3	2	4	1	2	7	2
t^k	1.382	1.218	2.035	2.413	1.381	1.970	1.275	1.381	1.217	1.382
$\overline{t^k}$	0.000	17.048	39.681	0.000	0.000	0.000	0.000	0.000	17.035	0.000
u^k	0.058	0.051	0.317	1.561	0.528	0.204	4.049	0.053	7.506	0.557
順序 k	51									
產品 i	4									
t^k	1.968									
$\overline{t^k}$	0.000									
u^k	6.377									
二方法搜尋次數： 54(次)		運算時間： 1.531(秒)			目標值： 32066.011 (\$)			週期 T ： 170.541 (天)		

由情況二之結果所示，在固定排程，僅經過 TVLS 求解所得到的目標值未必為全域最大值，經過 TVLS-search-T 通常可以找出較佳品質之解。故搭配情況三的排程改善，及有機會求解 TVLS 問題在基本週期乘數為 PoT 策略下的最佳解。

情況三：使用鄰域搜尋法改善排程並求解對 T 搜尋的 TVLS 模式

過往研究在僅有基本週期乘數及基本週期、未取得排程的情況下，多使用隨機構建排程，再搭配例如基因演算法的啟發式解法求解，但由於每產生一組排程都要求解 TVLS，其計算時間在代數過多或族群數設計太大的時候將帶來運算時間過長的結果，較無效率。

然而本研究使用鄰域搜尋法來改善排程，理由是因為截至此刻已經獲得由 LPT 派工法建立之初始排程，已經從一個較好的初始解開始搜尋，故選用鄰域搜尋法而不用較為複雜的啟發式解法，來簡省運算時間，停止條件設定為目標值改進程度小於誤差值即停止，誤差值設為 10^{-3} ，分別代入共同週期法、基因演算法、接合點搜尋法之排程。

比較結果如表 5.13 所示，NS-TVLS-search-T (CC) 表示以共同週期法參數求出之排程代入鄰域搜尋法改善排程、求解對 T 搜尋的 TVLS 模式，其他項目以此類推。可以看出在重新排列排程下，有機會找出更加之目標式值，但運算時間的部分與變數數量有關，此範例中以接合點搜尋法的變數數量為最多，共有 102 個，故求解時間為最久。

表 5.14—鄰域搜尋法改善排程與求解對 T 搜尋的 TVLS 模式數值比較

方法	排程 q	週期 T	目標值	運算時間
TVLS-search-T (CC)	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	21.268 (天)	32024.897 (\$)	0.406 (秒)
NS-TVLS-search-T (CC)	[9,1,2,4,5,6,7,8,3,10]	21.258 (天)	32024.897 (\$)	29.188 (秒)
TVLS-search-T (GA)	[2,7,9,3,1,2,4,5,8,6,2,7,10,2,4,2,7,9,3,1,2,4,2,7,2,4]	84.918 (天)	32065.939 (\$)	1.125 (秒)
NS-TVLS-search-T (GA)	[2,7,9,3,12,4,5,6,2,7,10,8,2,4,2,7,9,1,3,2,4,2,7,2,4]	86.211 (天)	32065.946 (\$)	500.328 (秒)
TVLS-search-T (JP)	[2,7,9,3,6,2,4,1,5,10,2,7,8,2,4,2,7,9,3,2,4,1,2,7,2,4,2,7,9,3,6,2,4,1,5,10,2,7,2,4,2,7,9,3,2,4,1,2,7,2,4]	170.541 (天)	32066.011 (\$)	1.531 (秒)
NS-TVLS-search-T (JP)	[7,2,4,1,2,7,9,2,4,5,2,7,3,8,2,4,1,2,7,9,2,4,6,10,2,7,3,2,4,1,2,7,9,2,4,5,2,7,3,2,4,1,2,7,9,2,4,6,10,2,3]	170.332 (天)	32066.011 (\$)	505.125 (秒)

5.2 隨機數據測試

本研究參考 Carreno(1990)所提出產品資料的範圍為基礎，並加上新增加的變數，其範圍如表 5.14 所示，每次隨機產生若干項產品，作為數值測式的題目。

Elmaghraby(1989)將產能利用率(即 $\sum_i \rho_i$)視為求解 ELSP 難易度的指標，通常情況下，利用率越高，越難找到較佳之解，求解時間上也會有所增加，因此本研究又將隨機產生資料的產能利用率限制在 0.50~0.90 之間，每 0.1 劃分一區間，即分為下列 4 個區間：[0.50-0.60]、[0.60-0.70]、[0.70-0.80]、[0.80-0.90]。每個區間使用 10、30 個產品數進行測試，每個區間皆測試 20 組資料，每組資料各有，分別進行求解，最後將每個區間的資料求平均值，來比較所找到解的品質及花費時間。

表 5.15—產品資料範圍

	Mean	Range
Demand Rate (units/day)	2500.000	4800.000
Production Rate (units/day)	14000.000	5000.000
Set-up Cost (\$)	200.000	400.000
Set-up Time (days)	0.280	0.440
Holding costs (\$/unit/year)	0.350	0.700
Price (\$/unit)	2.000	1.500
Variable Acquisition Cost (\$/unit)	0.800	0.600
Fixed Acquisition Cost (\$/unit)	100.000	75.000
Recycle Parameter	0.300	0.200
Consumption Parameter	1.300	0.200
Raw material Holding costs (\$/unit/year)	0.200	0.300
One year=240, 8-hours days.		

5.2.1 單次數值分析

若從 10 品項隨機數據測試中，分析單次的目標值及求解時間來，分別紀錄三種情況使用三種方法來計算出之值、排程長度 l 及求解時間，使用一隨機數據做說明，該次隨機參數數據如表 5.15，其求解結果如表 5.16，以下分點說明：

1. 三種方法中以接合點搜尋法求出之目標值為最好，比共同週期法高出 0.73%，而與基因演算法求出之排程相比則大約高出約 0.22%。
2. 越長的排程求解時間也越長，如 CC 預設排程長度為每個品項各生產一次，當品項數為 10 時的計算時間約是 30 秒，但當 GA 和 JP 求出的排程長度為 59 和 69 時，則提升至 430 秒及 1039 秒，原因是由於 Matlab 求解功能也是使用演算法求解，若變數個數增加將使求解複雜度提升。

表 5.16—10 品項隨機產品資料

No.	d	p	a	s	h	σ	α	β	ξ	λ	\bar{h}	θ
1	1039.028	18432.844	46.935	0.376	0.290	1.353	0	0	0	0	0	1
2	890.476	15290.414	37.109	0.134	0.442	2.143	0	0	0	0	0	1
3	532.832	14688.101	216.757	0.124	0.495	3.418	0	0	0	0	0	1
4	1306.938	15794.336	135.775	0.227	0.783	1.914	0	0	0	0	0	1
5	687.485	16108.310	423.377	0.615	0.163	3.181	0	0	0	0	0	1
6	995.249	11687.521	433.024	0.279	1.020	0.809	1.199	47.881	0.432	1.484	0.780	0.285
7	1002.235	14661.733	402.320	0.648	0.762	1.109	0.702	41.382	0.108	1.327	0.397	0.068
8	365.913	12645.558	444.686	0.181	0.556	0.997	0.567	41.444	0.301	1.366	0.329	0.315
9	780.713	12331.339	60.386	0.628	0.711	1.127	0.230	112.596	0.186	1.440	0.534	0.206
10	269.652	11415.784	502.323	0.190	1.035	2.097	1.011	85.263	0.451	1.215	0.357	0.946

表 5.17—隨機產品範例求解結果

方法	TVLS	TVLS-search T	NS-TVLS-search T	l	運算時間
CC	7159.957	7160.482	7160.482	10	29.750 秒
GA	7194.408	7198.063	7197.909	59	430.015 秒
與 CC 比較	0.47%	0.52%	0.52%		
JP	7212.810	7213.636	7213.636	69	1039.390 秒
與 CC 比較	0.73%	0.74%	0.74%		
與 GA 比較	0.26%	0.22%	0.22%		

5.2.2 多次數值分析

而從總體績效來講，對 20 組資料分別基因演算法與接合點搜尋法對共同週期法的改進百分比，取出平均、最大，及最小改進值，針對小型題目 10 品項和中大型題目 30 品項進從測試，呈現如下表 5.17 和 5.18。可以觀察：

1. 三種取得乘數 k_i 的方法中，確實是由 JP 方法可以取得三者最好的值。且如同一般 ELSP 問題一般，利用率越大對於問題可找到的平均改善率越低，原因是因為當問題利用率過高的時候，製程可以調動的彈性就越小，排程會漸漸趨近共同週期法。
2. 經過 TVLS-search-T 和 NS-TVLS-search-T 改善，值可以再獲得些許提升。
3. 然而從提升之比例來看，在平均值下，GA 或 JP 在各種方法都僅比 CC 提升些許，提升值基本上都不達 1%。

表 5.18—產品數為 10 項下對不同利用率求解結果(與共同週期法比較)

	TVLS(GA)			TVLS(JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.199%	1.235%	2.222%	0.199%	1.338%	2.580%
0.6~0.7	0.317%	0.941%	1.428%	0.317%	0.958%	1.583%
0.7~0.8	0.419%	0.510%	0.671%	0.485%	0.714%	1.172%
0.8~0.9	0.024%	0.463%	1.241%	0.243%	0.685%	1.460%
	TVLS-search T (GA)			TVLS-search T (JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.197%	1.188%	2.211%	0.200%	1.345%	2.602%
0.6~0.7	0.316%	0.829%	1.428%	0.317%	0.959%	1.583%
0.7~0.8	0.418%	0.538%	0.672%	0.485%	0.715%	1.173%
0.8~0.9	0.080%	0.514%	1.184%	0.242%	0.684%	1.465%
	NS-TVLS-search T (GA)			NS-TVLS-search T (JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.199%	1.235%	2.222%	0.200%	1.345%	2.602%
0.6~0.7	0.304%	0.902%	1.428%	0.317%	0.959%	1.583%
0.7~0.8	0.369%	0.539%	0.678%	0.485%	0.714%	1.172%
0.8~0.9	0.080%	0.550%	1.121%	0.243%	0.685%	1.465%

在 30 品項的中大型題目數據測中也有相同的情況結果，隨著利用率的上升，改善率逐漸下降，但傳統 ELSP 各種演算法與共同週期比較的改善率通常會在中型和大型的數據測試中，或得比較好的數值，然而本研究的數據分析並沒有因為品項增加而有顯著之改善，針對此結果進行思考，推測未考慮新品之原料成本會使得模式無法看出生產再生產品帶來的成本優勢(生產再生產品的取得成本可能較新品原料成本低)，會讓生產再生產品後目標值的改變較不顯著。

表 5.19—產品數為 30 項下對不同利用率求解結果(與共同週期法比較)

	TVLS(GA)			TVLS(JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.229%	1.323%	2.312%	1.023%	2.084%	4.693%
0.6~0.7	0.325%	1.019%	1.437%	1.089%	1.359%	1.531%
0.7~0.8	0.474%	0.832%	1.232%	0.972%	1.298%	1.652%
0.8~0.9	0.482%	0.762%	1.137%	0.988%	1.331%	1.634%
	TVLS-search T (GA)			TVLS-search T (JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.229%	1.328%	2.323%	1.026%	2.080%	4.693%
0.6~0.7	0.336%	1.022%	1.437%	1.093%	1.365%	1.531%
0.7~0.8	0.483%	0.873%	1.236%	0.976%	1.130%	1.643%
0.8~0.9	0.489%	0.773%	1.148%	0.736%	0.988%	1.371%
	NS-TVLS-search T (GA)			NS-TVLS-search T (JP)		
UF	Min	Avg	Max	Min	Avg	Max
0.5~0.6	0.236%	1.405%	2.327%	1.031%	2.166%	4.693%
0.6~0.7	0.344%	1.067%	1.479%	1.101%	1.440%	1.748%
0.7~0.8	0.564%	0.923%	1.279%	0.632%	1.114%	1.532%
0.8~0.9	0.513%	0.814%	1.347%	0.676%	0.994%	1.334%

5.2.3 運算時間分析

在單次數值分析中的第二點提到，本研究演算法的運算時間受到排程長度 l 的影響，本小節將以數據資料呈現此關係。以下圖 5.1 和 5.2 是 10 品項與 30 品項的運算時間散佈圖，可以看到兩者皆是在排程長度(橫軸)越大的情況下，運算時間(縱軸)越大。

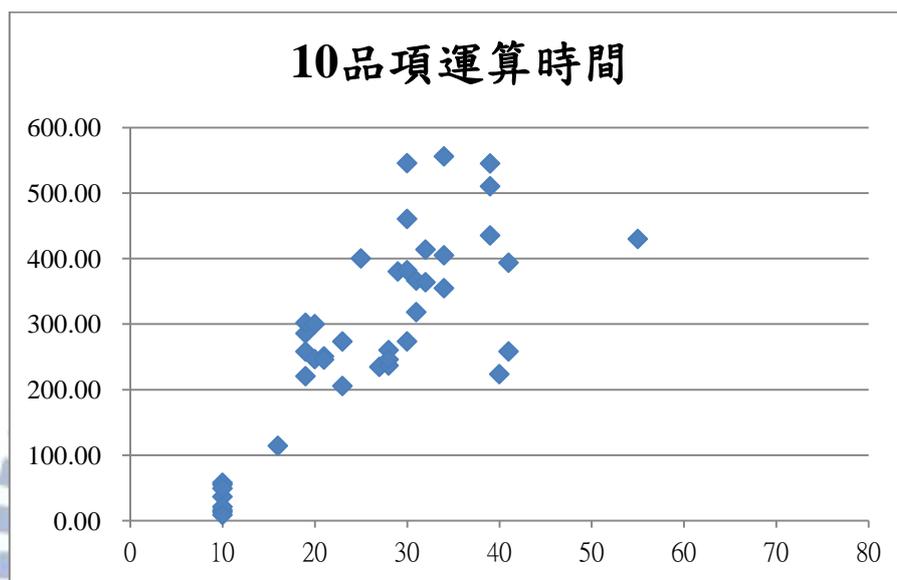


圖 5.1—10 品項運算時間散佈圖

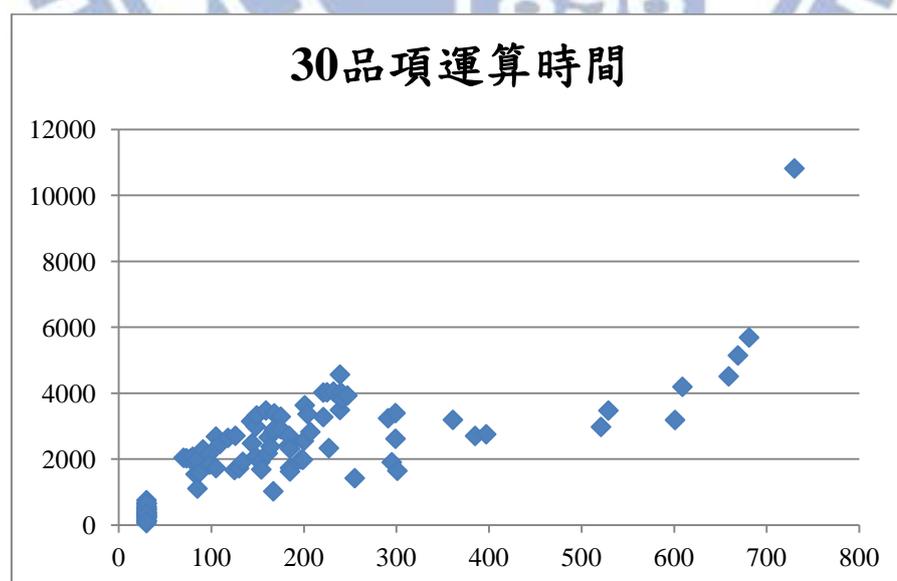


圖 5.2—30 品項運算時間散佈圖

5.2.4 小結

從數值分析可以觀察的一些情況，以下逐點分析：

1. 本研究提出之三種方法中，以接合點演算法求出之解的品質最好，不管在不同利用率下或是或是不同情況下都優於共同週期法與基因演算法求出之 TVLS 目標值。且如同一般 ELSP 問題一般，利用率越大對於問題可找到的平均改善率越低，原因是因為當問題利用率過高的時候，製程可以調動的彈性就越小，排程會漸漸趨近共同週期法。
2. 然而從提升之比例來看，在平均值下，GA 或 JP 在各種方法都僅比 CC 提升些許，提升值基本上都不達 1%。本研究認為此現象之原因，是由於本研究延續傳統 ELSP 問題之假設—不考慮一般產品原料成本，然而卻有考慮再生產品原料取得成本，因為此假設故生產一般產品有絕對的獲利性，造成即使生產了再生產品的獲利率提升也十分有限，因而無法看出目標式在生產一般產品與再生產品在利潤上的差異。
3. 求解時間與範例題品項數並於絕對關係，而是由排程長度 l 影響求解的變數數量，進而影響到求解時間，原因是由於 Matlab 目標式最小值求解功能 `fmincon` 求解，該運算功能也是使用演算法概念求解問題，若變數個數增加將使求解複雜度提升。

第六章 結論與建議

6.1 結論

本研究建立生產一般產品及再生產品的利潤最大化經濟批量排程問題之模式，並透過 TVLS 最佳解搜尋演算法進行求解，該題目為一個新的問題，從文獻中搜尋並無同時探討利潤最大化及回收商品的經濟批量排程問題。

針對求解此問題，本研究提出 TVLS 最佳解搜尋演算法。該方法分為前、中及後段三個部分，前段是以 BP-based 模式求出基本週期乘數 k_i 與基本週期 B ，本研究分別使用共同週期法(CC)、基因演算法(GA)、接合點搜尋法(JP)三種方法進行求解。

中段進行排程建立，本研究使用最長操作時間派工法(LPT)進行建立，將產品以乘數小及總生產時間大的品項優先排入各基期排程，再將各基期排程整合為 TVLS 用的排程，並用基本週期 B 乘以最大乘數作為 TVLS 週期 T 。

後段為 TVSL 求解，又分別考慮三種情況，情況一僅用 LPT 排程及週期 T 計算出滿足限制式的 TVLS 模式目標值；而情況二使用二分搜尋法，針對上下界進行給定排程下的最佳週期搜尋，改善了文獻中求解 TVLS 並未對週期 T 進行搜尋的求解方式；情況三再針對排程進行改善，本研究使用鄰域搜尋法進行排程的微調與重新求解，找出 TVLS 模式下的近似最佳解及其排程。

從數值分析的結果，可以判斷本研究提出之三種方法中以接合點演算法求出之解的品質最好，然而本研究延續傳統 ELSP 問題之假設—不考慮一般產品原料成本，故無法顯著的看出生產一般產品與再生產品在利潤上帶來的差異。模式的求解時間與方法並於絕對關係，而是由排程長度 l 影響求解的變數數量，進而影響到求解時間。

6.2 建議

1. 本研究在 TVLS 最佳解求解演算法中，針對改善排程使用了鄰域搜尋法，從結果來看確實是有較未修改排程前稍為在提升些許品質，但這未必是唯一可用的演算法，而鄰域的定義也會大大的影響其結果品質，故可以試用不同的演算法對排程進行改善。

2. 本研究使用二分搜尋法進行 TVLS 中週期 T 的搜尋，但僅提出產品整備時間加總作為合理的下界，及數理上合理之上界，並未提出據合理管理意涵之上界，由於推導該項上界在數理上十分複雜，故未來針對此題目進行延伸時可以嘗試提出合理的故定排程下週期搜尋上界。
3. 本研究在與共同週期法之數理上改善值較不為顯著，推斷原因是由於限制住再生產品的售價必須小於一般產品的售價，然而若進一步將一般產品的原料成本考慮進模式中，將有機會突顯出再生產品因為取得成本較一般產品原料成本為低，而又較明顯的改善值。



參考文獻

外語參考文獻：

- [1] Absi, N. and Kedad-Sidhoum, S., "The Multi-Item Capacitated Lot-Sizing Problem with Safety Stocks And Demand Shortage Costs," Computers & Operations Research, vol. 36, pp. 2926-2936, Nov 2009.
- [2] Boctor, F. F., "the 2-Product, Single-Machine, Static Demand, Infinite Horizon Lot Scheduling Problem," Management Science, vol. 28, pp. 798-807, 1982.
- [3] Bomberger, E., E., "a Dynamic Programming Approach to a Lot Size Scheduling Problem," Management Science, vol. 12, pp. 778-784, 1966.
- [4] Brander, P., Leven, E., and Segerstedt, A., "Lot Sizes in a Capacity Constrained Facility - a Simulation Study of Stationary Stochastic Demand," International Journal of Production Economics, vol. 93-4, pp. 375-386, Jan 2005.
- [5] Carreno, J. J., "Economic Lot Scheduling for Multiple Products on Parallel Identical Processors," Management Science, 36, 348-358, 1990.
- [6] Davis, S. G., "Scheduling Economic Lot Size Production-Runs," Management Science, vol. 36, pp. 985-998, Aug 1990.
- [7] Delporte, C. M. and Thomas, L. J., "Lot Sizing and Sequencing for N-Products on One Facility," Management Science, vol. 23, pp. 1070-1079, 1977.
- [8] Elmaghraby, S. E., "Economic Lot Scheduling Problem (ELSP) - Review and Extensions," Management Science, vol. 24, pp. 587-598, 1978.
- [9] Gallego, G., "Scheduling the Production of Several Items with Random Demands in a Single Facility," Management Science, vol. 36, pp. 1579-1592, Dec 1990.
- [10] Gascon, A. and Leachman, R. C., et al., "Multiitem, Single-Machine Scheduling Problem with Stochastic Demands - a Comparison of Heuristics," International Journal of Production Research, vol. 32, pp. 583-596, Mar 1994.
- [11] Giri, B. C., Moon, I., and Yun, W. Y., "Scheduling Economic Lot Sizes in Deteriorating Production Systems," Naval Research Logistics, vol. 50, pp. 650-661, Sep 2003.
- [12] Hanssmann, F., Opeation Research in Production and Inventory. New York: John Wiley & Sons, 1962.
- [13] Hsu, W. L., "on the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine," Management Science, vol. 29, pp. 93-105, 1983.

- [14] Khouja, M., Michalewicz, Z., and Wilmot, M., "the Use of Genetic Algorithms to Solve the Economic Lot Size Scheduling Problem," European Journal of Operational Research, vol. 110, pp. 509-524, Nov 1 1998.
- [15] Khoury, B. N., Abboud, N. E., and Tannous, M. M., "the Common Cycle Approach to the ELSP Problem with Insufficient Capacity," International Journal of Production Economics, vol. 73, pp. 189-199, Sep 2001.
- [16] Leachman, R. C. and Gascon, A., "A Heuristic Scheduling Policy for Multi-Item, Single-Machine Production Systems with Time-Varying, Stochastic Demands," Management Science, vol. 34, pp. 377-390, Mar 1988.
- [17] Li, Y. J., Chen, J., and Cai, X. Q., "Heuristic Genetic Algorithm for Capacitated Production Planning Problems with Batch Processing and Remanufacturing," International Journal of Production Economics, vol. 105, pp. 301-317, Feb 2007.
- [18] Maxwell, W., "The scheduling of economic lot sizes," Naval Research Logistics Quarterly, vol. 11, p. 35, 1964.
- [19] McClain, J. O., Perations Management: Production of Goods and Services, 3rd ed.: Prentice Hall, 1992.
- [20] Moon, I., Gallego, G., and Simchilevi, D., "Controllable Production-Rates in a Family Production Context," International Journal of Production Research, vol. 29, pp. 2459-2470, Dec 1991.
- [21] Pan, Z. D., Tang, J. F., and Liu, O., "Capacitated Dynamic Lot Sizing Problems in Closed-Loop Supply Chain," European Journal of Operational Research, vol. 198, pp. 810-821, Nov 2009.
- [22] Rogers, J., "A Computational Approach to the Economic Lot Scheduling Problem," Management Science, vol. 4, pp. 264-291, 1958.
- [23] Tang, O. and Teunter, R., "Economic Lot Scheduling Problem with Returns," Production and Operations Management, vol. 15, pp. 488-497, Win 2006.
- [24] Teunter, R., Kaparis, K., and Tang, O., "Multi-Product Economic Lot Scheduling Problem with Separate Production Lines for Manufacturing and Remanufacturing," European Journal of Operational Research, vol. 191, pp. 1241-1253, Dec 16 2008.
- [25] Teunter, R., Tang, O., and Kaparis, K., et al., "Heuristics for the Economic Lot Scheduling Problem with Returns," International Journal of Production Economics, vol. 118, pp. 323-330, Mar 2009.
- [26] Yao, M. J., "The Economic Lot Scheduling Problem under Power-of-Two

- Policy," Computer and Operations Research, 28, 1441-1460,2001.
- [27] Zandoni, S., Segerstedt, A., Tang, O., and Mazzoldi, L., "Multi-Product Economic Lot Scheduling Problem with Manufacturing and Remanufacturing Using a Basic Period Policy," Computers & Industrial Engineering, vol. 62, pp. 1025-1033, 2012.
- [28] Zhou, S. X., Tao, Z., and Chao, X., "Optimal Control of Inventory Systems with Multiple Types of Remanufacturable Products," Manufacturing & Service Operations Management, vol. 13, pp. 20-34, Win 2011.
- [29] Zhou, S. X. and Yu, Y., "Optimal Product Acquisition, Pricing, and Inventory Management for Systems with Remanufacturing," Operations Research, vol. 59, pp. 514-521, Mar-Apr 2011.

中文參考文獻：

- [1]王俊傑,"求解不完美系統下允許回收再製造之經濟批量排程問題," 國立成功大學, 碩士, 民 99.
- [2]李翔豐,"以共同週期法求解製造與再製造系統下允許變動生產速率之經濟批量排程問題," 國立成功大學, 碩士, 民 100.
- [3]傅俊源,"週期需求的批量製造與再製造之生產規劃問題," 國立清華大學, 碩士, 民 100.
- [4]黃志強,"再製造過程中使用努力值下之最佳回收量研究," 國立中央大學, 碩士, 民 100.
- [5]鄧喻心,"考慮再製造的兩階層供應鏈最佳存貨策略," 國立臺灣科技大學, 碩士, 民 98.

簡歷



姓名：蔡季軒

出生地：台北市

出生日期：民國 77 年 07 月 04 日

聯絡地址：台北市信義區忠孝東路五段 372 巷 27 弄 59 號 4 樓

電子信箱：julyforth.tsai@nctu.edu.tw

學歷：

民國 101 年 7 月 國立交通大學 運輸科技與管理學系 碩士班畢業

民國 99 年 6 月 國立交通大學 運輸科技與管理學系 學士班畢業

民國 95 年 6 月 台北市立中崙高級中學 畢業