

東 海 大 學

工業工程與經營資訊研究所

碩士論文

考慮維修產能限制下之群組機器維修排程
問題之研究

研 究 生：楊美翎

指導教授：姚銘忠 教授

黃嘉彥 教授

中 華 民 國 九 十 六 年 七 月

The Maintenance Scheduling Problem for Family of Machines with Maintenance Capacity Constraints

By
Mei-Ling Yang

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

考慮維修產能限制下之群組機器維修排程問題之研究

學生：楊美翎

指導教授：姚銘忠教授

黃嘉彥教授

東海大學工業工程與經營資訊研究所

摘要

本研究探討求解維修產能限制下之群組機器維修排程問題，首先針對本問題推導其數學模式，再進行完整的理論分析並運用理論分析的結果為基礎，提出一套有效的搜尋演算法求取候選解答(candidate)，同時運用一個啟發式解法判定候選解答是否可以產生一組可行的維修排程，最後利用簡單範例說明本研究提出的搜尋演算法之求解過程，及利用隨機實驗說明本研究之求解方法比 Goyal 和 Kusy(1985)與 Yao(2005)提出的解法，可以得到品質較佳的解答。

關鍵字：維修、排程、產能限制、搜尋演算法

The Maintenance Scheduling Problem for Family of Machines with Capacity Constraints Maintenance System

Student: Mei-Ling Yang

Advisor: Prof. Ming-Jong Yao

Prof. Jia-Yen Huang

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

In this study, we are interested in solving the maintenance scheduling problem for a family of machines with maintenance capacity constraints. In order to propose our solution approach, we first derived a mathematical model for the concerned problem. Then, we conduct full analysis on the mathematical model. By utilizing our theoretical results, we propose a search algorithm to obtain candidate solutions. Also, we employ an efficient heuristic that tries to generate a feasible maintenance schedule for each one of those candidate solutions. Finally, among those with feasible maintenance schedules, we pick the ‘best’ solution with the minimum average total costs. We demonstrate that the proposed solution approach out-performs a previously published one using our numerical experiments.

Keywords: Maintenance, scheduling, capacity constraints, search algorithm, heuristic

致謝

首先要感謝的是指導老師姚銘忠教授以及共同指導的黃嘉彥教授。除了在學問和課業上的教導外，於待人處事上的身教言教也令學生受益匪淺。十分感謝老師們在研究進行期間對論文的悉心鞭正與提出寶貴的意見。同時亦要感謝口試委員邱天嵩博士，在口試期間所不吝給予的指導與建議，使本論文得以更加完備。

在這兩年中，最大的收穫是遇到了每一位真心相待的伙伴。特別是E103-1的同學們，綉菱、嘉合、何陽、茜雯、原豪、豪哲和政益，不僅在論文撰寫期間互相打氣、毫不保留的互相幫助，兩年來的課業上和生活裡都是不可或缺的重要部份。還有學長昕煒、哲鋒、紹維和智鈞，總是毫不藏私的給予我們各種經驗分享和幫助。而永遠給我們支援和協助的學弟妹，奕茵和正偉，感謝你們的用心和辛苦。當然，其餘研究室同學們更是增添美好回憶的重要元素。還有系上熱心協助的助教們，宏華姊、玉玲、瑋珊、雅慧姊、素卿姊和陳助教，在此也要獻上真誠的感謝。

最後，以此文獻給我摯愛的雙親、溫爸爸、溫媽媽、男友和哥哥，因為有了你們在背後支持、體諒和包容，讓我能專心於課業及論文上之研究。

楊美翎 謹誌於

東海大學工業工程與經營資訊研究所

中華民國九十六年七月

目錄

摘要.....	I
ABSTRACT.....	II
致謝.....	III
目錄.....	IV
表目錄.....	VII
圖目錄.....	VIII
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	1
1.3 研究方法.....	2
1.4 論文架構.....	2
第二章 文獻探討.....	4
2.1 週期性維修排程問題.....	4
2.2 Goyal 和 Kusy 的啟發式方法.....	5
2.3 Yao 的啟發式方法.....	7
2.4 小結.....	8
第三章 考慮維修產能限制下之群組機器維修排程問題之數學模式.....	11
3.1 假設與符號說明.....	11
3.2 數學模式.....	12
3.3 小結.....	15
第四章 理論分析.....	16
4.1 近似最佳總成本函數之片段凸性.....	16
4.2 近似最佳總成本函數之接合點.....	18
4.2.1 接合點位置的推導.....	18
4.2.2 接合點的理論性質探討.....	19
4.3 近似最佳總成本曲線之結構.....	20
4.4 小結.....	21

第五章 搜尋演算法設計	22
5.1 搜尋演算法的上界.....	22
5.2 搜尋演算法的下界.....	23
5.2.1 搜尋演算法的下界 β_1	23
5.2.2 搜尋演算法的下界 β_2	25
5.3 運用接合點進行搜尋.....	28
第六章 可行解測試演算法.....	32
6.1 設定初始條件.....	35
6.2 初始排程(Proc. IS)與機器排程(Proc. PS)	35
6.3 調整週期的方式.....	38
6.3.1 移出(The Smooth-Out Routine).....	39
6.3.2 1 對 1 互換(The Pair-Exchange Routine)	41
6.3.3 2 對 1 互換(The Two-to-One Exchange Routine)	44
6.3.4 隨機選取 $\lfloor n/2 \rfloor$ 台機器重新排程.....	47
6.4 輸出	47
第七章 數據實驗	48
7.1 範例說明.....	48
7.1.1 本研究之搜尋演算法.....	48
7.1.2 Goyal 和 Kusy 與 Yao 之啟發式方法	49
7.2 隨機實驗.....	51
7.3 敏感度分析.....	52
第八章 結論與未來研究方向.....	56
8.1 結論.....	56
8.2 未來研究方向.....	56
參考文獻.....	57
附錄 A 數學證明.....	59
附錄 B-1 Proc. IS 之流程圖.....	60
附錄 B-2 Proc. PS 之流程圖	61

附錄 B-3	移出之流程圖.....	62
附錄 B-4	1 對 1 互換之流程圖.....	63
附錄 B-5	2 對 1 互換之流程圖.....	64
附錄 C	維修時間範圍之決定.....	65

表目錄

表 6.1 機器基本資料(Proc. IS 和 Proc. PS 範例).....	36
表 6.2 機器 2 加入排程(Proc. IS 和 Proc. PS 範例).....	37
表 6.3 機器 1 加入排程(Proc. IS 和 Proc. PS 範例).....	37
表 6.4 結果(Proc. IS 和 Proc. PS 範例).....	38
表 6.5 改善前(The Smooth-Out Routine 範例).....	40
表 6.6 改善後(The Smooth-Out Routine 範例).....	41
表 6.7 改善前(The Pair-Exchange Routine 範例).....	43
表 6.8 改善後(The Pair-Exchange Routine 範例).....	44
表 6.9 改善前(The Two-to-One Exchange Routine 範例).....	46
表 6.10 改善後(The Two-to-One Exchange Routine 範例).....	47
表 7.1 範例說明之數據.....	48
表 7.2 範例說明之維修排程.....	50
表 7.3 範例說明之維修排程.....	50
表 7.4 參數範圍.....	51
表 7.5 在 100 組隨機試驗中可行解個數與本研究在可行解成本之改善 ...	53
表 7.5 在 100 組隨機試驗中可行解個數與本研究在可行解成本之改善(續)	54
表 C.1 調整因子之選擇基準.....	66
表 C.2 在 100 組隨機問題中可行解個數.....	66

圖目錄

圖 1.1 論文架構流程.....	3
圖 2.1 Goyal 和 Kusy 之流程圖	9
圖 2.2 Yao 之流程圖.....	10
圖 3.1 維修排程之範例說明.....	14
圖 4.1 平均總成本曲線.....	18
圖 5.1 成本曲線($k_i^*(\bar{T})=1$)	26
圖 5.2 成本曲線($k_i^*(\bar{T})>1$)	27
圖 5.3 搜尋演算法流程圖.....	31
圖 6.1 可行解測試演算法流程圖.....	34
圖 B.1 Proc. IS 之流程圖	60
圖 B.2 Proc. PS 之流程圖	61
圖 B.3 移出之流程圖	62
圖 B.4 1 對 1 互換之流程圖	63
圖 B.5 2 對 1 互換之流程圖	64

第一章 緒論

1.1 研究背景與動機

本研究的主題為考慮維修產能限制下之群組機器的維修排程問題。對工廠內管理機器設備維修的經理人而言，其關心的是如何在多台機器的維修系統下，將某些機器合併維修，藉由分擔整備成本以降低平均總成本。因此，決策者必須決定每台機器的最佳維修週期，使其在滿足維修產能條件限制下，使得平均總成本達到最小。

在Luss和Kander(1974)、Luss(1976)與Christer和Doherty(1977)等文獻中提到決定群組機器經濟性維修頻率問題在科學管理、作業研究及工業工程皆已被廣泛的研究與探討。Luss和Kander(1974)與Christer和Doherty(1977)考慮的是相同群組機器的聯合維修政策。Sule和Harmon(1979)考慮的是不同機器的維修，有可能是進行聯合維修或是進行個別維修。Cho和Parlar(1991)與Dekker, et al.(1997)針對週期性維修排程問題指出在各種不同的決策情境下，多機器維修問題有多種不同的數學模式；上述所提到的文獻，其數學模式中的操作成本函數大多採用線性函數。Goyal和Kusy(1985)提出每台機器的操作成本函數為一非線性函數。Yao (2005) 延伸Goyal和Kusy(1985)的問題，其中包含針對啟發式方法的理論性質提出完整的理論分析與提出一套有效率的搜尋演算法求解Goyal和Kusy(1985)的問題。

探討有關群組機器維修問題的文獻很多，但是考慮有維修產能限制問題的論文卻不多。在考慮維修產能限制的機器排程問題之相關研究中，僅有 Amotz, et al.(2002)在其文章中有提到此類有關於維修產能限制的決策情境，允許最多有 M 台機器可以被安排在任何的時段以進行維修。有鑑於此類問題的重要性，本研究將針對考慮有維修產能限制條件下，最佳群組機器維修排程問題進行深入探討。

1.2 研究目的

本研究假設某工廠共有 n 台機器，為了避免機器在運作過程中發生損壞的情形，所以機器必須定期的接受維修。機器設備的管理者必須妥善規劃各部機器的維修排程，以使得平均總成本能夠降到最低，而且同時能兼顧到維修產能的限制。

由於在工廠內某些機器維修過程中，需要特殊專業維修人員，因此，管理機器設備的經理人必須聘請外界專業維修人員到工廠內進行維修，故會產生一項固定成本，此固定成本包含聘請專業人員的費用與當進行維修工作時，因為機器無法繼續運作，導致機器停擺將付出的閒置成本。此項固定成本和機器數目無關，並且只有在進行維修工作才會產生。

當針對機器 i 開始進行維修工作時，除了會產生一項維修整備成本 m_i ，還會產生操作成本 z_i ，也就是機器 i 操作一段 t 時間後，且在此段時間內沒有進行維修工作所產生的成本。此操作成本包含操作機器 i 的固定成本和變動成本。我們可合理地假設從上一次的維修工作結束後，機器 i 的操作成本是隨時間增加而增加。本研究假設操作成本函數為 $O_i(t) = f_i + v_i t^e$ ，是一非線性函數。

若在維修系統中，沒有考量維修產能的限制，僅追求最低的平均總成本，在實務上可能造成機器設備的管理者作出錯誤的決策，產生不可行的維修排程。為了填補在文獻上無考量維修產能限制的缺陷，本研究加入維修產能限制條件，使數學模式建構所求得之最佳解更能貼近實務操作的情境。

因此，本研究的目的是找出一個合理可行的維修排程，包括決定每台機器 i 的最佳維修乘數 k_i 與最佳維修基本週期 T ，以使得所有機器的平均總成本降到最低。

1.3 研究方法

本研究在求解的過程中，除了必須搜尋 $(\{k_i\}, T)$ 以達到最低的平均總成本之外，並且要求在以 $(\{k_i\}, T)$ 為基準的維修排程必須滿足維修產能限制是可行。因此，本研究分成二階段，在第一階段探討最佳成本曲線的理論性質，並依此設計演算法以搜尋區域解作為最佳解之候選(candidate)；第二階段再將第一階段所產生的候選解，運用一個啟發式解法(heuristic)的演算程序產生一個可行的維修排程，以求得考慮維修產能限制下之群組機器維修排程問題的最佳解。

1.4 論文架構

本研究論文的內容共分為八章，研究架構如圖1.1所示，分別簡述如下：第一章簡介本論文之研究背景、目的及研究方法；第二章整理及探討

與本研究相關的文獻；第三章介紹考慮維修產能限制下之群組機器維修排程問題之數學模式，求解最佳的維修乘數與維修基本週期，使得平均總成本最小並且在第四章本研究為最佳成本曲線進行理論性質分析，了解區域解在最佳成本曲線上分佈之位置與特性。第五章運用第四章所得之理論性質提出一套搜尋演算法，以求得最佳成本曲線上的區域解。第六章介紹一個啟發式解法，來判斷某一區域解是否可以產生可行的維修排程。第七章運用範例說明本問題的求解過程，並以隨機實驗展現本論文所提出的研究方法可以有效率地求得品質優良的解，並可以產生一個可行的維修排程。第八章歸納本研究的結論、建議事項及未來研究方向。

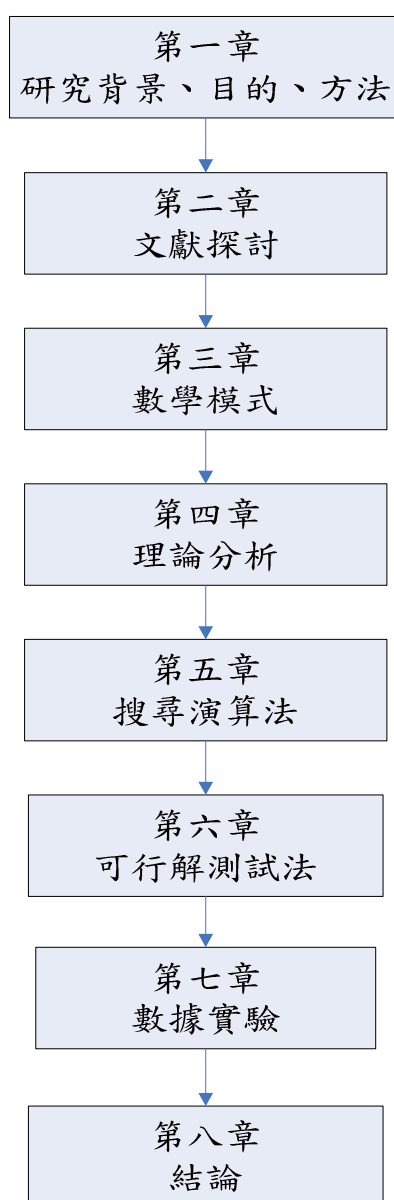


圖 1.1 論文架構流程

第二章 文獻探討

本章節主要針對「維修排程問題」進行文獻探討。第2.1節整理週期性維修排程問題(Periodic Maintenance Scheduling Problem)。因為本研究之數學模式是建構在Goyal和Kusy(1985)與Yao(2005)的模式之上，因此分別在第2.2節與第2.3節介紹Goyal和Kusy(1985)與Yao(2005)提出的數學模式與啟發式演算法。第2.4節小結。

2.1 週期性維修排程問題

首先在本小節整理和群組機器維修排程(Maintenance Scheduling Problem for Family of Machines, MSPFM)相關的週期性維修排程問題(Periodic Maintenance Scheduling Problem)的文獻。

Cho 和 Parlal(1991)整理有關多單位系統的最佳化維修和更換模式，並將此類問題區分成五種模式。Dekker, et al(1997)探討具有經濟相依的多組成維修模式，並分成穩定模式：假設維修系統為長期穩定狀況與動態模式：可以把短期的資訊當作是可用的資訊。Hassin 和 Megiddo(1991)分析具有時間區間限制的存貨排程模式，目標函數是使單位時間平均總成本最小，其中和大多數存貨理論不同的是文章中假設沒有任何條件限制。Holte, et al.(1992)運用二種方法決定最小循環排程長度並且快速產生一個線上計劃表。Glass(1992,1994)考慮經濟批量排程問題，也就是如何決定多產品在單一機台上的生產排程，以使得持有成本和整備成本達到最小。Goyal 和 Gunasekaran(1992)運用疊代的方法找出問題的最佳解，缺點是他們提出的搜尋方法通常在第一次疊代中假如沒有獲得最佳解則會停止，也就是會被困在局部最佳解而無法找到全域最佳解。上述提到的文獻與 Chan 和 Chin(1992,1993), Su 和 Tassiulas(1997)其數學模式中的操作成本函數為線性函數。Amotz, et al.(2002)針對有維修限制之排程問題提出一套廣泛作決策的情境，並允許在每個時段裡，最多可以有 M 台機器被安排維修，自上一次維修之後，任何時段的操作成本和時段數相關，假設第 h 時段，機器 i 的操作成本為 $(h + b)a_i$ ，其中 $h \geq 0, b \geq 0$ 且為整數， a_i 為實數，並且和維修服務相關的成本為 $c_i \geq 0$ ，目標函數為找出最佳維修排程，使得長期平均成本最小，同時提出維修排程問題為 NP-hardness 問題，並且提供有效率的演算法求解近似最佳解。Anily, Glass 和 Hassin(1998)在 Amotz, et al.(2002)之

前，提出的一個特例，其中 $M = 1, c_i = 0, i = 1 \dots n$ 。Anily, Glass 和 Hassin(1999) 提出啟發式方法求解，並得到最糟的成果比例為 1.033。Krämer and Bai(1996) 說明當機器接受維修時，則生產系統停工，所以需求將不被滿足，因此研究目標是決定如何生產以降低缺貨並藉由最佳控制模式系統化生產維修管理問題，且使用 Pontryagin 最小化原則求解。

2.2 Goyal 和 Kusy 的啟發式方法

Goyal 和 Kusy(1985) 假設機器的操作成本函數是非線性函數 ($O_i(t) = f_i + v_i t^e$)，所以無法使用先前其他研究者提及的求解方法求解。因此，他們提出一套疊代演算法，試圖找出 MSPFM 模式的基本週期和維修乘數。

以下將介紹 Goyal 和 Kusy(1985) 提出的啟發式方法。茲先將 MSPFM 模式所運用之符號定義及說明如下：

n = 機器數目。

M = 在開始維修之前整個群組機器的整備成本。

T_i = 第 i 台機器的維修週期。

T = 基本週期。

$Z_i(T, k_i)$ = 第 i 台機器在維修週期 $k_i T$ 內的平均總成本。

$Z(T, k_1, k_2, \dots, k_n)$ = 整個群組機器在維修週期內的平均總成本。

m_i = 第 i 台機器的整備成本。

f_i = 操作機器 i 的固定成本。

v_i = 操作機器 i 的變動成本。

e = 操作機器的變動成本中對時間的冪次數。

$O_i(t)$ = 在操作機器 i 一段 t 時間後的操作成本。

MSPFM 模式所運用之假設如下：

1. 從上一次的維修工作結束後，每台機器的操作成本是隨時間增加而增加。
2. 操作機器 i 一段 t 時間後，且此段時間內沒有進行維修的操作成本為

$$O_i(t) = f_i + v_i t^e。$$

3. 維修之前的整備時間和維修機器的工作時間皆假設為零。
4. 每台機器是進行週期性維修，而非等到機器損壞後才進行維修，因此第 i 台機器的維修週期為 $T_i = k_i T, k_i \in \mathbb{R}^+, i=1, \dots, n$ 。
5. 在維修排程中不考慮突發狀況(非計畫性)下的機器損壞。
6. 在維修工作開始之前會產生一項和機器數目無關的整備成本。

MSPFM 模式中考慮的成本項，包含以下三項：

1. 平均維修主要整備成本(Average cost of maintenance setups) = $\frac{M}{T}$ 。
2. 平均維修次要整備成本(Average cost of carrying out maintenance) = $\sum_{i=1}^n \frac{m_i}{k_i T}$ 。

3. 平均操作成本(Average operating cost for the machines)

$$= \sum_{i=1}^n f_i + \frac{T^e}{e+1} \sum_{i=1}^n v_i k_i^e。$$

因此，平均總成本為

$$Z(T, k_1, k_2, \dots, k_n) = \frac{M}{T} + \sum_{i=1}^n f_i + \sum_{i=1}^n Z_i(T, k_i) \quad (2.1)$$

$$\text{其中 } \sum_{i=1}^n Z_i(T, k_i) = \sum_{i=1}^n \frac{m_i}{k_i T} + \sum_{i=1}^n \frac{T^e}{e+1} v_i k_i^e。$$

MSPFM 模式中，目標函數是決定機器 i 的維修週期 $k_i T$ ，也就是要找出維修基本週期 T 及維修乘數 k_i 值，以使得單位時間平均總成本最小。因此，令平均總成本(式 2.1)的一階導數等於零，找其最佳值。計算結果可得到下列二式：

$$T(k_i) = \left[\left(\frac{e+1}{e} \right) \left(\frac{M + \sum_{i=1}^n \frac{m_i}{k_i}}{\sum_{i=1}^n v_i k_i^e} \right) \right]^{\frac{1}{e+1}} \quad (2.2)$$

$$k_i(T) = \left[\frac{(e+1)m_i}{e v_i} \right]^{\frac{1}{e+1}} \frac{1}{T} \quad (2.3)$$

給定 T 值，則平均總成本(式 2.1)的前二項 $(\frac{M}{T} + \sum_{i=1}^n f_i)$ 為一常數，所以

要使平均總成本最小，則 $\sum_{i=1}^n Z_i(T, k_i)$ 要最小，因此要符合下列二式：

$$Z_i(T, k_i(T)) \leq Z_i(T, k_i(T) + 1) \quad (2.4)$$

$$Z_i(T, k_i(T)) \leq Z_i(T, k_i(T) - 1) \quad (2.5)$$

式 (2.4) 與 (2.5) 經過整理後可得到式 (2.6) 與 (2.7)，如下所示：

$$\frac{X_i}{T^{e+1}} \leq k_i(T)[k_i(T) + 1]\{[k_i(T) + 1]^e - [k_i(T)]^e\} \quad (2.6)$$

$$\frac{X_i}{T^{e+1}} \geq k_i(T)[k_i(T) - 1]\{[k_i(T)]^e - [k_i(T) - 1]^e\} \quad (2.7)$$

其中 $X_i = \frac{m_i(e+1)}{v_i}$ 。

Goyal 和 Kusy(1985)的啟發式方法之流程圖參見圖 2.1 所示，求解步驟如下所示：

Step1：根據 X_i 的遞增順序去安排機器，其中 $X_i = \frac{m_i(e+1)}{v_i}$ 。

Step2：在第一次疊代中，假設 $k_i = k_i^{(0)} = 1$ ，並代入式(2.2)，求得 $T = T^{(1)}$ 的估計值。

Step3：利用式(2.6)與(2.7)找出 $k_i^{(1)}$ 值。

Step4：將 $k_i^{(1)}$ 代入式(2.2)，求得 $T = T^{(2)}$ 。如此重複 Step2 和 Step3，直到

$$k_i^{(r)} = k_i^{(r-1)} \text{ 才停止，同時找到最佳解 } T^* = T^{(r)} \text{ 和 } (k_1^*, \dots, k_n^*) = (k_1^{(r)}, \dots, k_n^{(r)})。$$

Goyal 和 Kusy(1985)提出的啟發式方法，無法保證所得到的解為最佳解，因為在數據實驗中，疊代方法會限制一些局部最佳解的搜尋。

2.3 Yao的啟發式方法

Yao(2005)針對 Goyal 和 Kusy(1985)的問題進行完整的理論分析，並根據理論分析的結果提出一套新的求解方法。由最佳成本函數的曲線圖可知，其曲線型式如同許多凹型曲線連續組合而成，因此最佳成本函數為一個對 T 具有片段凸性的函數。另外，在片段凸性函數中，兩個連續曲線之交點稱作接合點，同時接合點提供了在某個時間位置上，所應該要選擇的維修乘數之資訊。由於這些性質皆與週期 T 有關，因此搜尋最佳解時，決

定在訂定的搜尋上下界內進行搜尋，且保證在此範圍之外不可能有最佳解存在。Yao(2005)採用共同週期法(Common Cycle approach, CC)當作是搜尋過程中的上界 T_{\max} ，並且藉由最佳基本週期和最佳目標函數兩者之間的關係推出演算法的下界 T_{\min} 。

由數據實驗觀察到在區間 $[T_{\min}, T_A]$ 與 $[T_B, T_{\max}]$ ，目標函數是單調遞減，其中 $T_{\min} \leq T_A < T_B \leq T_{\max}$ ，假如能改善搜尋範圍使其縮短為 $[T_A, T_B]$ ，相對的就可以減少不必要的搜尋時間，進而達到加快搜尋的速度。

Yao(2005)提出之搜尋演算法架構，如下：假設 T_c 是目前搜尋過程中的基本週期值，接下來找離 T_c 最近的接合點位置且定義為 $\delta_\pi(k_\pi)$ ，其中 $\pi = \arg \max_i \{\delta_i(k_i) < T_c\}$ 為下一個要改變維修乘數的機器下標，透過 $\mathbf{k}^*(\delta_\pi(k_\pi)) \equiv (\mathbf{k}^*(T_c) \setminus \{k_\pi\}) \cup \{k_\pi + 1\}$ ，得到新的最佳維修乘數，同時計算此時的局部最小值 $\tilde{T}(\mathbf{k}^*(\delta_\pi(k_\pi)))$ 和目標函數值 $\Psi(\mathbf{k}^*(\delta_\pi(k_\pi)), \tilde{T}(\mathbf{k}^*(\delta_\pi(k_\pi))))$ ，並與手邊現有的目標函數值進行比較，若新的目標函數值較小，則手邊現有的基本週期、維修乘數和目標函數則被取代；反之，手邊現有的基本週期、維修乘數和目標函數值為最佳值。接著重覆相同的步驟，直到 $T_c \leq T_{\min}$ 則停止搜尋動作，並找到最佳基本週期、最佳維修乘數和最佳目標函數。演算細部流程圖如圖 2.2 所示。

2.4 小結

綜合以上文獻，可知有關考慮維修產能限制下之群組機器維修排程問題的研究很少。其中在 Goyal 和 Kusy(1985)與 Yao(2005)的決策情境下，假設維修整備時間和維修時間皆為零，對於實務操作上，該模式下所得的最佳解，可能無法產生可行的維修排程，因而誤導管理機器設備之經理人的維修決策。有鑑於此，本研究將在第三章數學模式建構中考慮維修產能限制，使求解品質更能符合實務操作。

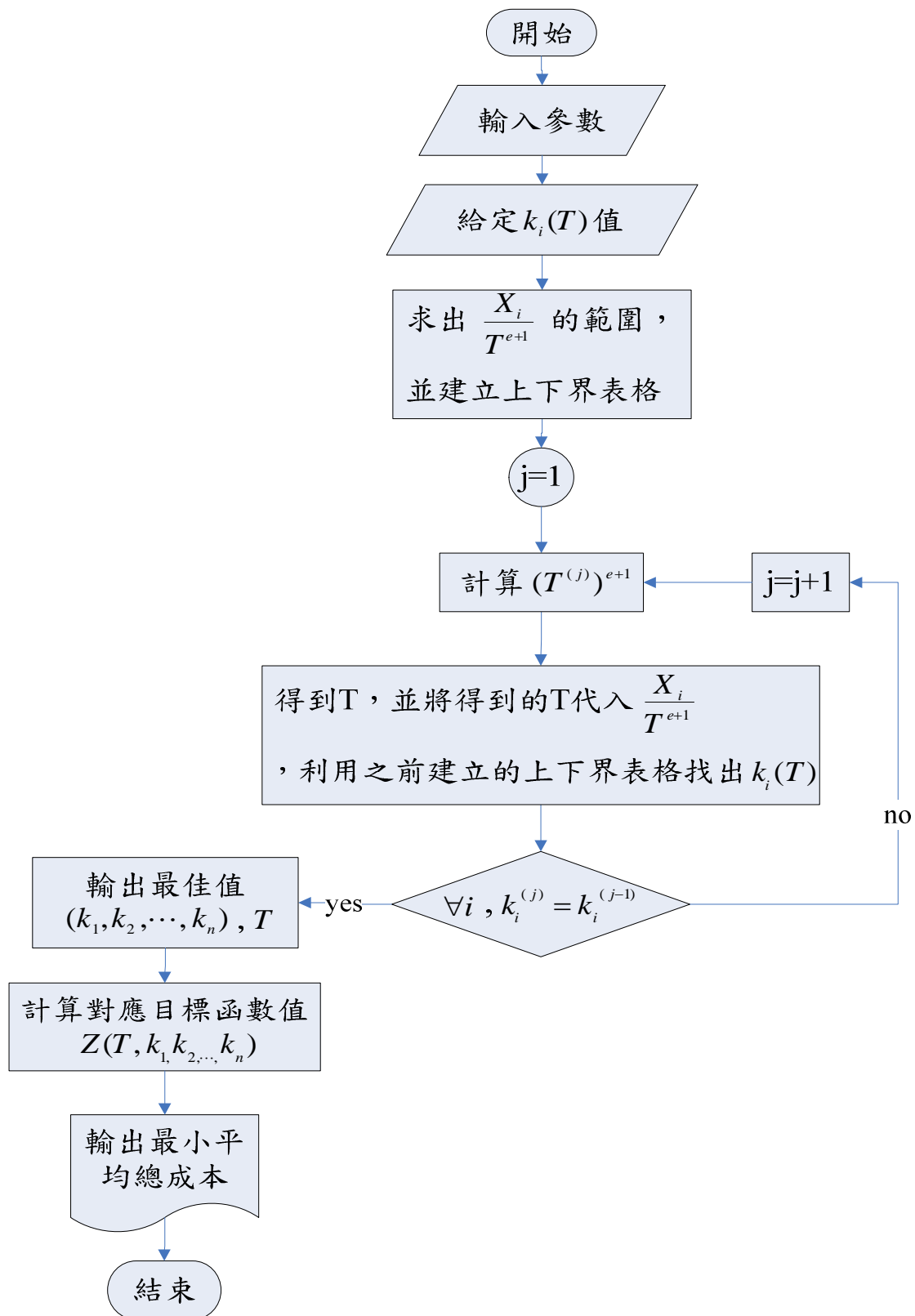


圖 2.1 Goyal 和 Kusy 之流程圖

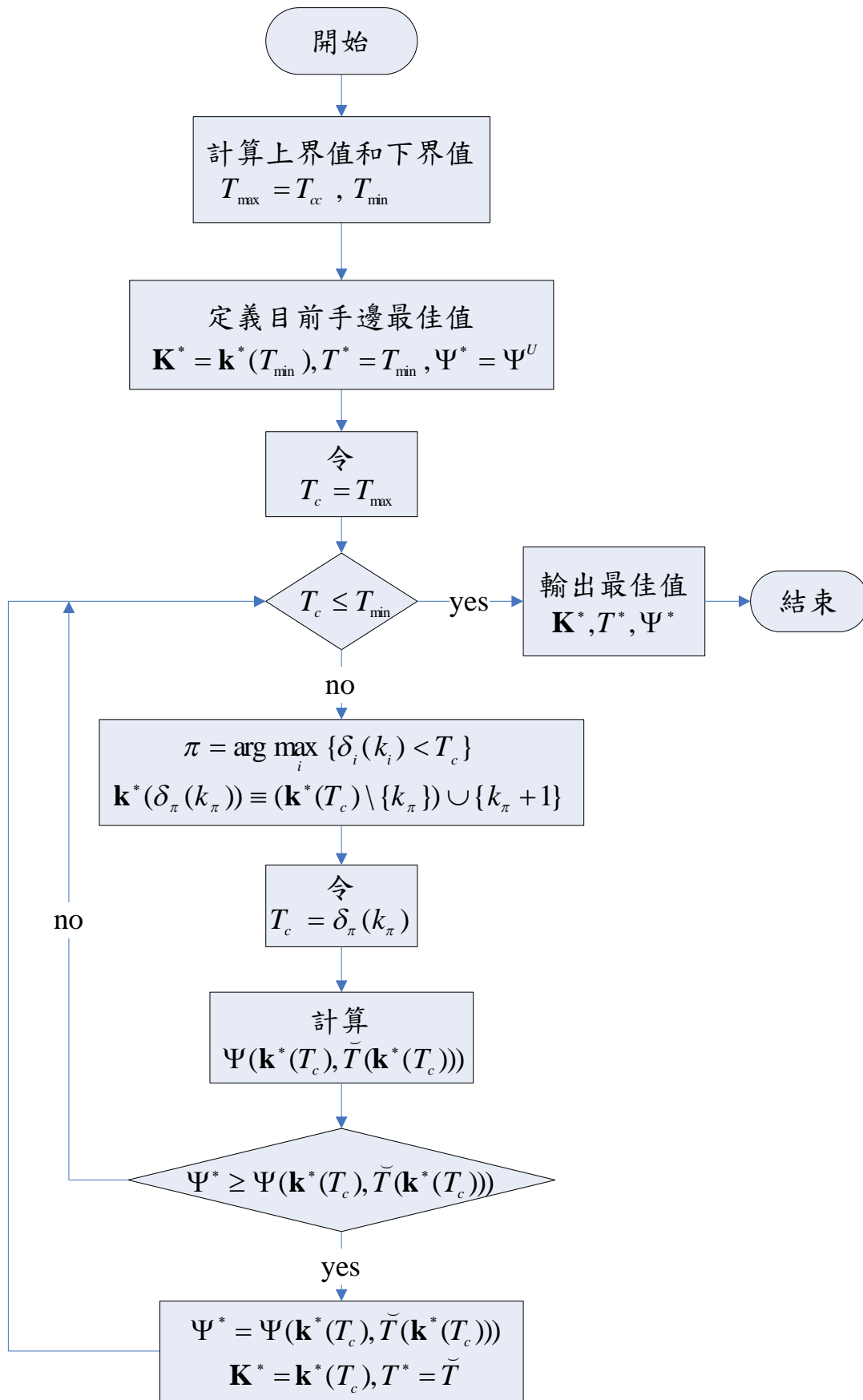


圖 2.2 Yao 之流程圖

第三章 考慮維修產能限制下之群組機器維修排程問題之 數學模式

本章將介紹考慮維修產能限制下之群組機器維修排程問題(MSPFM)之數學模式。本問題運用到的假設與符號說明將在第 3.1 節介紹，第 3.2 節說明目標函數、決策變數與數學模式。

3.1 假設與符號說明

考慮維修產能限制下之 MSPFM 模式所運用之假設如下：

1. 從上一次的維修工作結束後，每台機器的操作成本是隨間隔時間增加而增加；操作成本與間隔時間呈非線性遞增的關係。
2. 操作機器 i 一段 t 時間後，且此段時間內沒有進行維修工作的操作成本函數為 $O_i(t) = f_i + v_i t^e$ 。
3. 維修系統中考慮維修產能上的限制，例如：只有一台維修設備或是僅有一位專業維修人員。
4. 同一時間點，只能維修一台機器，也就是說必須等待上一台機器的維修工作完成後才能繼續下一台機器的維修工作。
5. 每台機器是進行週期性維修，而非等到機器損壞後才進行維修，因此第 i 台機器的維修週期為 $T_i = k_i T, k_i \in \mathbb{N}^+, i = 1, \dots, n$ 。
6. 在維修排程中不考慮突發狀況（非計畫性）下的機器損壞。
7. 在維修工作開始之前會產生一個和機器數目無關的主要整備成本。
8. 機器 i 整個維修工作花費 $X_i = \tau_i + \pi_i$ 單位時間，其中 τ_i 為機器 i 維修之前的整備時間， π_i 為機器 i 的維修時間。

考慮有維修產能限制下之 MSPFM 模式中，所運用的符號定義如下：

n = 機器數目。

M = 主要整備成本，表示開始維修之前整個群組機器的整備成本。

i = 機器指標， $i = 1, 2, \dots, n$ 。

T_i = 第 i 台機器的維修週期。

T = 基本週期，例如：一天、一週或一個月。

$Z(T, k_1, k_2, \dots, k_n)$ = 整個群組機器在維修週期內的平均總成本。

m_i =第 i 台機器的次要整備成本。

$O_i(t)$ =在操作機器 i 一段 t 時間後的操作成本。

f_i =操作機器 i 的固定成本。

v_i =操作機器 i 的變動成本。

e =操作機器的變動成本中對時間的冪次數。

3.2 數學模式

假設 n 台機器的主要整備成本、次要整備成本與操作成本皆為已知參數，目標函數要求解一組 $(\{k_i\}, T)$ 參數，能同時降低平均總成本 $Z(T, k_1, k_2, \dots, k_n)$ 且滿足維修產能限制。MSPFM 模式的目標函數可以表示為式 (3.1)，維修產能限制如式 (3.2~3.7) 所示，機器 i 的維修週期可表示成基本週期的整數倍，如式 (3.3) 所示，本模式為非線性整數規劃模式。

$$\text{Minimize } Z(k_1, k_2, \dots, k_n, T) = \frac{M}{T} + \sum_{i=1}^n \frac{(m_i + z_i)}{k_i T} \quad (3.1)$$

$$\text{Subject to } \sum_{i=1}^n (\tau_i + \pi_i) w_{i\varphi(i,t)} \leq T, \quad t = 1, 2, \dots, K \quad (3.2)$$

$$T_i = k_i T, \quad k_i \in \mathbb{N}^+, \quad i = 1, \dots, n \quad (3.3)$$

$$\sum_{t=1}^{k_i} w_{it} = 1, \quad i = 1, 2, \dots, n \quad (3.4)$$

$$\left. \begin{array}{l} w_{it} = 1, \text{ 機器 } i \text{ 在第 } t \text{ 基期有進行維修} \\ w_{it} = 0, \text{ 其他} \end{array} \right\} \quad (3.5)$$

$$K = \text{lcm}(k_1, k_2, \dots, k_n) \quad (3.6)$$

$$\varphi(i, t) = \begin{cases} t \bmod k_i, & \text{if } t \neq \gamma k_i, \gamma \in \mathbb{N} \\ k_i, & \text{if } t = \gamma k_i, \gamma \in \mathbb{N} \end{cases} \quad (3.7)$$

首先討論目標函數的推導： $Z(k_1, k_2, \dots, k_n, T)$ 表示平均總成本， $\frac{M}{T}$ 表示單位時間主要整備成本， m_i 與 z_i 分別表示機器 i 的次要整備成本與操作成本。其中機器 i 的操作時間為維修週期減去整個維修工作花費的時間 $(k_i T - X_i)$ 。操作成本 z_i 的推導過程如下所示：

$$\begin{aligned}
z_i &= \int_0^{k_i T - X_i} O_i(t) dt \\
&= \int_0^{k_i T - X_i} (f_i + v_i t^e) dt \\
&= f_i(k_i T - X_i) + \frac{v_i}{e+1} (k_i T - X_i)^{e+1} \\
&= f_i(k_i T - X_i) + \frac{v_i}{e+1} (k_i T)^{e+1} \left(1 - \frac{X_i}{k_i T}\right)^{e+1}
\end{aligned} \tag{3.8}$$

其中 $X_i = \tau_i + \pi_i$ 表示整個維修工作花費的時間為整備時間與維修時間之總和。機器 i 單位時間次要整備成本與操作成本之和以 $\bar{\Phi}_i(k_i, T) = \frac{m_i + z_i}{k_i T}$ 表示，將式(3.1)進行整理如下：

$$\begin{aligned}
Z(k_1, k_2, \dots, k_n, T) &= \frac{M}{T} + \sum_{i=1}^n \frac{(m_i + z_i)}{k_i T} \\
&= \frac{M}{T} + \sum_{i=1}^n \frac{m_i}{k_i T} + \sum_{i=1}^n \frac{f_i(k_i T - X_i) + \frac{v_i}{e+1} (k_i T)^{e+1} \left(1 - \frac{X_i}{k_i T}\right)^{e+1}}{k_i T} \\
&= \frac{M}{T} + \sum_{i=1}^n \frac{m_i}{k_i T} + \sum_{i=1}^n f_i - \sum_{i=1}^n \frac{f_i X_i}{k_i T} + \sum_{i=1}^n \frac{v_i}{e+1} (k_i T)^e \left(1 - \frac{X_i}{k_i T}\right)^{e+1} \\
&= \frac{M}{T} + \sum_{i=1}^n \frac{m_i - f_i X_i}{k_i T} + \sum_{i=1}^n f_i + \sum_{i=1}^n \frac{v_i}{e+1} (k_i T)^e \left(1 - \frac{X_i}{k_i T}\right)^{e+1} \\
&= \frac{M}{T} + \sum_{i=1}^n \bar{\Phi}_i(k_i, T)
\end{aligned}$$

$$\text{其中 } \bar{\Phi}_i(k_i, T) = \frac{m_i - f_i X_i}{k_i T} + \frac{v_i}{e+1} (k_i T)^e \left(1 - \frac{X_i}{k_i T}\right)^{e+1} + f_i \tag{3.9}$$

有關維修產能限制式之說明如下：式(3.2)表示在每一基期有進行維修工作的機器之整個維修工作花費時間的加總必須符合維修產能限制。式(3.4)表示機器 i 每 k_i 個基期只會發生一次維修工作，若有發生維修工作則 w_{it} 值為 1，沒有發生維修工作則 w_{it} 值為 0，如式(3.5)所示。全部維修排程之總基期數 (K) 為 n 台機器之維修乘數的最小公倍數，如式(3.6)所示。式(3.7)表示當第一次維修工作排定後，其餘基期如何依照維修乘數進行維修工作的安排。

在考慮有維修產能限制下之 MSPFM 問題，其模式的決策變數欲求得能使平均總成本最小的維修乘數與基本週期 ($\{k_i\}, T$)，並且滿足維修產能限制。

在此以一個簡單的例子說明如何安排4台機器的維修排程。假設基本週期為 T ，維修乘數為 $\{k_1, k_2, k_3, k_4\} = \{2, 2, 4, 4\}$ ，整備時間與維修時間之和為 $\tau_i + \pi_i$ ， $i = 1, \dots, 4$ 。維修排程結果參見圖3.1所示。

MSPFM 模式的一個特殊情況，也就是當操作成本函數之操作機器的變動成本中對時間的冪次數 $e=1$ 時，MSPFM 模式即變成是聯合補貨問題 (Joint Replenishment Problem, JRP) 模式。JRP 探討在單一補貨系統下多項產品的批量補貨法則，其主要的目的是在滿足顧客需求的前提下，減少換模/換線所需要的整備時間及成本，使平均總成本最小。Arkin 等 (1989) 證明 JRP 模式為非多項式求解的問題，大部分學者都以啟發式演算法求解。Shu (1971)、Nocturne (1973)、Goyal (1973) 將所有產品分成兩組求解。Shu (1971) 和 Nocturne (1973) 使用繪圖式的啟發式演算法。van Eijs (1993) 提出一套更有效率的演算法，修正 Goyal (1974) 所提出的方法。Viswanathan (1996) 提出一般整數策略下 JRP 模式的最佳解結構為基本週期的片段凹性函數，但在文章中並沒有多加證明，且無法保證其解為全面最佳解。Lee 和 Yao (2003) 提出在二冪策略下，可求得全面最佳解的演算法。

綜合以上文獻，我們可以知道 JRP 模式是沒有考慮產能限制下的補貨系統。但是在真實的環境中，不可能無限提供產能，以滿足需求。因此，在本研究中，我們加入了有限維修產能限制這項因素，使數學模式建構所求得之最佳解更能符合真實環境。

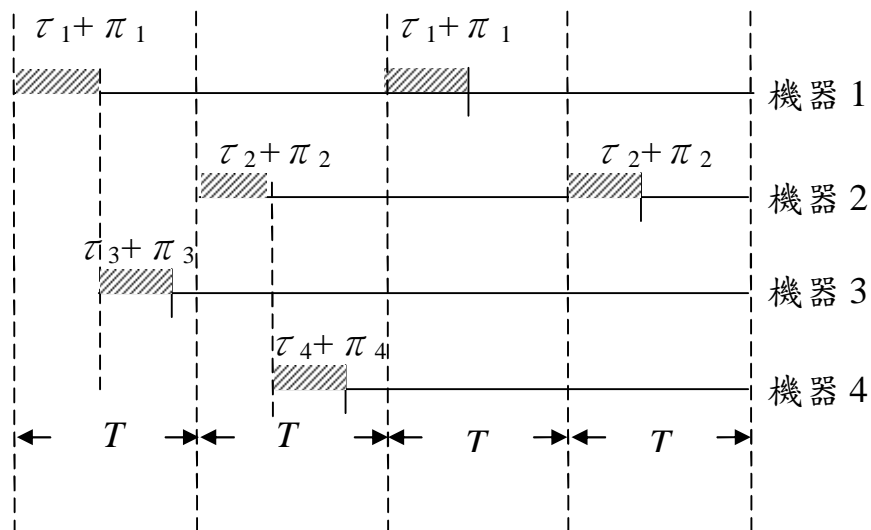


圖 3.1 維修排程之範例說明

3.3 小結

本研究為了彌補文獻沒有考慮維修產能上的限制，所以在本章節之數學模式的建構中，本研究加入有限維修產能這項因素，因此本研究之數學模式與文獻不同之處在於因考量維修產能限制而產生式(3.2~3.7)這六條限制式，除此之外，在操作成本中操作時間是由 $[0, k_i T]$ 修訂成為 $[0, k_i T - X_i]$ ，所以導致目標函數多了 $\frac{-f_i X_i}{k_i T}$ 與 $(1 - \frac{X_i}{k_i T})^{e+1}$ 這二項(參見式(2.1)與式(3.9))。

第四章 理論分析

在第二章文獻探討，Goyal 與 Kusy(1985)所提出的搜尋演算法，並沒有完整的理論根據，所以無法確保求解的品質。因此，在本章節我們主要根據 Yao(2005)所提出的理論分析為基礎，並延伸其觀念作為本研究推導理論分析之依據，因為本研究加入了產能限制條件，所以在數學模式的推導過程中會多產生 $\frac{-f_i X_i}{k_i T}$ 項。首先推導數學模式的理論性質並提出一套完整的理論分析，以利設計搜尋演算法求解最佳解。由於操作成本函數太過於複雜(式(3.8))，為了方便進行接下來的研究，我們運用近似成本操作函數方法求得近似值，推導本章節的理論分析。第 4.1 節中介紹近似最佳總成本函數之片段凸性，說明平均總成本曲線是由多個凹型曲線所組成。第 4.2 節介紹近似最佳總成本函數之接合點，說明可透過相鄰的成本凹型曲線之交點去改變維修乘數。第 4.3 節介紹近似最佳總成本曲線之結構。第 4.4 節小結。

4.1 近似最佳總成本函數之片段凸性

當機器維修時間佔整個維修週期比例很小時，可以假設 $(1 - \frac{X_i}{k_i T})^{e+1} \approx 1$ ，所以操作成本函數可以表達成式(4.1)。

$$z_i = f_i(k_i T - X_i) + \frac{v_i}{e+1}(k_i T)^{e+1} (1 - \frac{X_i}{k_i T})^{e+1} \approx f_i(k_i T - X_i) + \frac{v_i}{e+1}(k_i T)^{e+1} \quad (4.1)$$

並定義新函數，如下所示：

$$\Phi_i(k_i, T) \equiv \frac{m_i - f_i X_i}{k_i T} + \frac{v_i}{e+1}(k_i T)^e + f_i \equiv \bar{\Phi}_i(k_i, T) \quad (4.2)$$

我們以相同的時間區間 T 來看，針對不同的 k 可以得到不同的平均總成本曲線。從圖4.1我們可以清楚地知道，對應不同的 k 之平均總成本曲線，會產生不同的交點，我們把所有平均總成本曲線 $Z(k_1, \dots, k_n, T)$ 的最低部份命名為 $\Psi(k_1, \dots, k_n, T)$ ，即

$$\Psi(k_1, \dots, k_n, T) = \inf_{T>0} \left\{ \frac{M}{T} + \sum_{i=1}^n \Phi_i(k_i, T) \mid k_i \in \square^+, i=1, \dots, n \right\} \quad (4.3)$$

其曲線型式就如同許多凹型曲線連續組合而成，我們將此性質以下列

Proposition 1 敘述之。但是為了證明Proposition 1，我們必須先探討Lemma 1，因此先令

$$g_i(T) \equiv \inf_{k_i \in N^+} \{\Phi_i(k_i, T') \mid T = T' \in \square^+\} \quad (4.4)$$

所以式(4.3)將被改寫成

$$\Gamma(T) = \inf_{T>0} \left\{ \frac{M}{T} + \sum_{i=1}^n g_i(T) \right\} \quad (4.5)$$

Lemma 1： $g_i(T)$ 對 T 是一個片段凸性的函數。

[Proof]：

在平均總成本函數裡， $Z(k_1, \dots, k_n, T) \approx \frac{M}{T} + \sum_{i=1}^n \Phi_i(k_i, T)$ ，其中

$$\Phi_i(k_i, T) = \frac{m_i - f_i X_i}{k_i T} + \frac{v_i k_i^e T^e}{e+1} + f_i, \quad k_i \in \square^+, \quad i=1, \dots, n。$$

我們先針對函數 $\Phi_i(k_i, T)$ 分別進行一階導數和二階導數，得到：

$$\frac{d^2 \Phi(k_i, T)}{dk_i^2} = \frac{2(m_i - f_i X_i)}{k_i^3 T} + \frac{e(e-1)v_i T^e}{e+1} k_i^{e-2} > 0 \quad (4.6)$$

表示當 $T > 0$ 時，對一特定之 k_i 值，函數 $\Phi_i(k_i, T)$ 為開口向上的凹型曲線 (convex curve)，有極小值在 $T = \frac{x_i^*}{k_i}$ 位置產生，其中

$$x_i^* = \left[\frac{(e+1)(m_i - f_i X_i)}{e v_i} \right]^{\frac{1}{e+1}} \quad (4.7)$$

所以 $\Phi_i(k_i, T)$ 對 T 而言是一個連續可微分的函數。因為 $g_i(T)$ 是由許多不同的 k_i 之 $\Phi_i(k_i, T)$ 函數所組成，也就是 $g_i(T) \equiv \inf_{k_i \in N^+} \{\Phi_i(k_i, T') \mid T = T' \in \square^+\}$ ，所以 $g_i(T)$ 同樣為一個對 T 具有片段凸性的函數。■

在此我們運用 Lemma 1 來證明 Proposition 1。

Proposition 1： $\Gamma(T) = \inf_{T>0} \left\{ \frac{M}{T} + \sum_{i=1}^n g_i(T) \right\}$ 對 T 為一片段凸性的函數

[Proof]：

在平均總成本函數裡，我們不考慮 $\Phi_i(k_i, T)$ 的部份，同時令 $\theta(T) = \frac{M}{T}$ ，

再針對 $\theta(T)$ 分別進行一階導數與二階導數，得到在 $\forall T > 0$ 時， $\frac{d^2\theta}{dT^2} > 0$ 。因此，我們可以知道 $\theta(T)$ 為一個開口向上的凹型曲線函數。

透過Lemma 1 我們可以知道 $g_i(T)$ 具有片段凸性的特性，而凹型曲線函數加上片段凸性函數還是為具有片段凸性的函數。因為 $\Gamma(T)$ 是 $\theta(T)$ 與 $g_i(T)$ 之和，所以 $\Gamma(T)$ 同樣也是對 T 為一片段凸性的函數。■

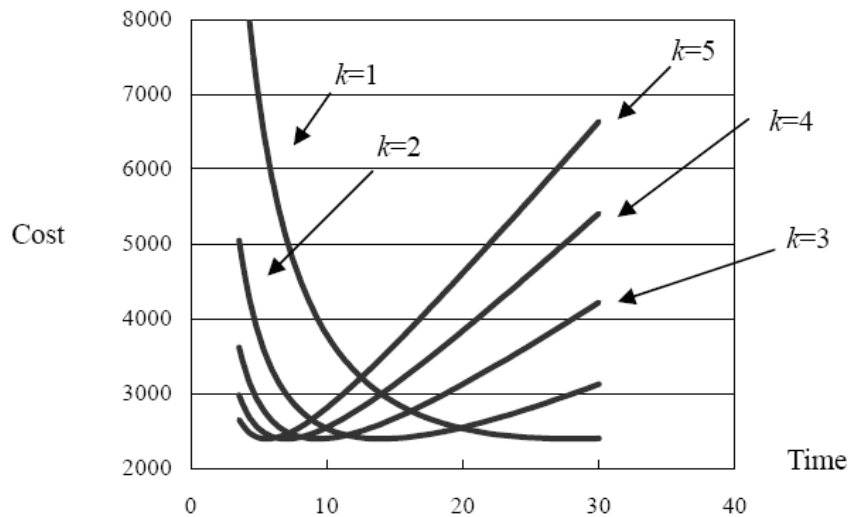


圖 4.1 平均總成本曲線

4.2 近似最佳總成本函數之接合點

在前一節中我們已經證明 $\Gamma(T)$ 是具有片段凸性的函數。而在本節中我們將介紹在 $\Gamma(T)$ 函數曲線上的接合點(Junction Point)。首先定義接合點是在片段凸性函數中，於特定時間 T 上之兩個連續曲線的交點。另外，我們將會證明 $\Gamma(T)$ 函數曲線上的接合點會等於 $g_i(T)$ 函數曲線上的接合點(見 Proposition 2)。根據第4.1節可知 $g_i(T)$ 是一具有片段凸性的函數，所以我們將先透過 $g_i(T)$ 函數上接合點位置的推導(即求出 $\Psi(k_1, \dots, k_n, T)$ 接合點的位置)，再來證明 $\Gamma(T)$ 函數曲線將會繼承 $g_i(T)$ 函數曲線上所有接合點的特性。

4.2.1 接合點位置的推導

$g_i(T)$ 函數上接合點的位置之推導可以透過鄰近的 $\Phi_i(k_i, T)$ 與 $\Phi_i(k_i + 1, T)$ 二個函數求得。首先令兩鄰近函數 $\Phi_i(k_i, T)$ 之差為 $\Delta_i(k_i, T)$ ：

$$\Delta_i(k_i, T) = \Phi_i(k_i + 1, T) - \Phi_i(k_i, T)$$

$$= -\frac{(m_i - f_i X_i)}{T(k_i + 1)k_i} + \frac{v_i((k_i + 1)^e - k_i^e)}{e + 1} T^e \quad (4.8)$$

我們注意到 $\Delta_i(k_i, T)$ 為維修週期 k_i 與 $k_i + 1$ 的成本差，且 $\Delta_i(k_i, T)$ 是隨著時間 T 增加而遞增的函數。假設有一搜尋演算法從上界往時間 T 遞減的方向開始進行搜尋，同時計算出 $\Delta_i(k_i, T)$ 的值會由正值轉變為零，再由零轉變為負值。我們令 w 為 $\Delta_i(k_i, T)$ 之值為零的時間點， k_i 為 $T > w$ 之前的最佳維修乘數。這表示在時間 T 到達 w 之前， k_i 都是最佳的維修乘數，而當時間 T 小於 w 時，則是以 $k_i + 1$ 為其最佳的維修乘數。

其中值得注意的是： w 是兩個鄰近的凹型曲線 $\Phi_i(k_i, T)$ 與 $\Phi_i(k_i + 1, T)$ 之交點，也就是接合點。而更重要的是接合點提供了在那個時間位置上，所應該要選擇的維修乘數之資訊。透過 $\Delta_i(k_i, T) = 0$ ，可以得到接合點位置之公式，如下所示：

$$\delta_i(k_i) = \left[\frac{(m_i - f_i X_i)(e + 1)}{(k_i + 1)k_i v_i ((k_i + 1)^e - k_i^e)} \right]^{\frac{1}{e+1}} \quad (4.9)$$

綜合上述， $\delta_i(k_i)$ 代表第 k_i 個接合點，其中 $k_i \in N^+$ ， $i = 1, \dots, n$ 。當時間 $T > \delta_i(k_i)$ 時，我們應選擇的維修乘數為 $k_i^* = k_i$ ，否則選擇 $k_i^* = k_i + 1$ 來促使 $g_i(T)$ 函數的值為最小。而接合點位置在時間 T 上是彼此有順序的，從時間由長至短來看，其順序依序為 $\delta_i(1), \delta_i(2), \delta_i(3), \dots$ ，等。其中 $\delta_i(k_i)$ 代表的是 k_i 和 $k_i + 1$ 的交點。例如：對應於機器 i 近似成本曲線之 $\delta_i(1)$ 代表 $k_i = 1$ 與 $k_i = 2$ 的交點。

4.2.2 接合點的理論性質探討

本節之理論性質，對於近似最佳總成本曲線結構之研究，有非常重要的影響，其特性將在第4.3節中繼續討論，在此我們先加以證明。

Proposition 2： $\Gamma(T)$ 曲線將會繼承 $g_i(T)$ 曲線所有接合點的特性。

[Proof]：

由於 $\Gamma(T) = \theta(T) + \sum_{i=1}^n g_i(T)$ 是可以分割的，所以 $\Gamma(T)$ 會繼承 $g_i(T)$ 所有接合點的特性。換句話說，Proposition 2 說明在 $g_i(T)$ 的片段凸性曲線上出現之接合點 $\delta_i(k_i)$ ，也同樣會出現在 $\Gamma(T)$ 的片段凸性曲線上而成為接合點。假設 w 是機器 i 之接合點，但不是其他 $n - 1$ 台機器的接合點，並假設存在一誤

差值 ε ，其中 $\varepsilon > 0$ ，並符合下列：

1. 假如針對每個 $g_i(T)$ 在區間 $[w - \varepsilon, w + \varepsilon]$ 為凸函數，則曲線 $\sum_{j \neq i} g_j(T)$ 在區間 $[w - \varepsilon, w + \varepsilon]$ 是一凸函數，其中 $j \neq i$ 。
2. $g_i(T)$ 在區間 $[w - \varepsilon, w]$ 和 $[w, w + \varepsilon]$ 是一凸函數。
3. $\theta(T)$ 在區間 $[w - \varepsilon, w]$ 和 $[w, w + \varepsilon]$ 是一凸函數。

所以 $\Gamma(T) = \theta(T) + g_i(T) + \sum_{j \neq i} g_j(T)$ 在區間 $[w - \varepsilon, w]$ 和 $[w, w + \varepsilon]$ 仍然是一凸函數，因此 w 為 $\Gamma(T)$ 曲線上的接合點。■

以下將再介紹接合點的另一特性，以作為搜尋演算法的基礎。

Lemma 2: 假設 k_L^* 和 k_R^* 分別是在 $g_i(T)$ 函數上接合點左右兩邊的最佳維修乘數，則 $k_L^* = k_R^* + 1$ 。

[Proof] :

透過接合點位置之公式可以得到：

$$\delta_i(1) > \delta_i(2) > \dots > \delta_i(k_i) > \delta_i(k_i + 1) > \dots > \delta_i(v) \quad (4.10)$$

其中 v 為 k_i 值的上界。

我們以 $k_i^*(T)$ 代表在給定時間 T 值的 $g_i(T)$ 函數中，所得到的最佳維修乘數。由式(4.9)與 $\Phi_i(k_i, T)$ 是凹型曲線的特性，可以知道當 $T \in [\delta_i(1), \infty)$ 時， $k_i^*(T) = 1$ ；當 $T \in [\delta_i(k_i + 1), \delta_i(k_i))$ 時， $k_i^*(T) = k_i + 1$ ，所以我們可以清楚地得到 $k_L^* = k_R^* + 1$ 。■

延伸 Lemma 2 的觀念，我們可以得到 Corollary 1，如下所示：

Corollary 1: 在一給定的時間點 T ， $g_i(T)$ 函數的最佳維修乘數，如下所示：

$$k_i^*(T) = \begin{cases} 1, & T > \delta(1) \\ m, & \delta(m) \leq T < \delta(m-1) \end{cases} \quad (4.11)$$

4.3 近似最佳總成本曲線之結構

在第4.2節中所提到接合點的特性給了我們以下的提示。為使符號簡單化，我們定義維修乘數向量 $\mathbf{k} \equiv (k_1, \dots, k_n)$ 。

Theorem 1: 假設 $\mathbf{k}^{(L)}$ 和 $\mathbf{k}^{(R)}$ 分別是在 $\Gamma(T)$ 函數上接合點左右兩邊的最佳維修乘數向量，則 $\mathbf{k}^{(L)}$ 可藉由 $k_i^{(L)} = k_i^{(R)} + 1$ 從 $\mathbf{k}^{(R)}$ 中獲得。

[Proof] :

透過之前Proposition 2與Lemma 2即可以得證。■

我們先定義 T_m 是 $\Psi(k_1, \dots, k_n, T)$ 函數曲線中，對應於維修乘數 k_i 的局部最小值之時間位置。根據第4.1節和第4.2節的敘述，我們知道 $\Psi(k_1, \dots, k_n, T)$ 函數最佳解的位置會落在接合點以及 $\Psi(k_1, \dots, k_n, T)$ 曲線的局部最低端點上。因此，我們首先要找出接合點的時間位置，再判斷 $\Psi(k_1, \dots, k_n, T)$ 的局部最低端點之時間位置 T_m 是否位於接合點 $\delta_i(k_i - 1)$ 與 $\delta_i(k_i)$ 之間。若有，則比較 $Z(k_1, \dots, k_n, T_m)$ 與 $Z(k_1, \dots, k_n, \delta_i(k_i))$ 之值，取成本值較小者為最佳解。

$\Psi(k_1, \dots, k_n, T)$ 函數的局部最小值 T_m 求法如下所示：

$$Z(k_1, k_2, \dots, k_n, T) \approx \frac{M}{T} + \sum_{i=1}^n \frac{m_i - f_i X_i}{k_i T} + \sum_{i=1}^n f_i + \sum_{i=1}^n \frac{v_i}{e+1} (k_i T)^e$$

$$\text{令 } \frac{\partial Z(k_1, \dots, k_n, T)}{\partial T} = 0$$

$$-\frac{M}{T^2} - \sum_{i=1}^n \frac{m_i - f_i X_i}{k_i T^2} + \sum_{i=1}^n \frac{v_i k_i^e e}{e+1} T^{e-1} = 0$$

$$T_m = \left[\left(\frac{e+1}{e} \right) \left(\frac{M + \sum_{i=1}^n \frac{m_i - f_i X_i}{k_i}}{\sum_{i=1}^n v_i k_i^e} \right) \right]^{\frac{1}{e+1}} \quad (4.12)$$

4.4 小結

本章節針對數學模式的理論性質提出一套完整的理論分析，說明在求解的過程中可以利用近似總成本曲線的接合點去改變維修乘數，進而找到問題的最佳解。因此，接下來將在第五章介紹以此理論分析為基礎所設計出來的搜尋演算法。

第五章 搜尋演算法設計

在第四章，我們已經證明目標函數 $\Psi(k_1, \dots, k_n, T)$ 是一條片段凸性函數曲線，且得到一些片段凸性函數的性質，例如：每一段凸性函數有局部最小值、透過接合點可得到相對應的維修乘數。由於這些性質皆與時間 T 有關，因此搜尋 $\Psi(k_1, \dots, k_n, T)$ 函數的最佳解時，決定沿著時間 T 軸進行搜尋。但是在搜尋的過程中，必須定義搜尋範圍的上下界 $[T_U, T_L]$ ，且保證在此範圍之外不可能有最佳解存在。所以，在本章的第5.1節與第5.2節，我們先訂定搜尋範圍的上下界。第5.3節則介紹本研究所提出來的搜尋演算法。

5.1 搜尋演算法的上界

為了在時間軸 T 上進行搜尋，我們需要訂定搜尋時的範圍。換言之，需要訂定上界 T_U (upper bound) 與下界 T_L (lower bound)。且訂定上下界後，可以確保在下界之下與上界之上，目標函數不會有滿足維修產能限制的局部最小值產生。因此，目標函數之全域最佳值一定發生在上下界之間。後續求取上下界的過程中，將分別予以證明此一主張。

由於本研究是以基本週期 T 與維修乘數 k_i 作為決策變數，以達到滿足維修產能限制的最小維修成本之問題。因此，推導上界時，我們是運用 Yao(2005) 的觀念，以不考慮維修產能限制的第一個接合點位置 $\delta_i(1)$ 當作是搜尋過程中的上界，在此情況下，針對每一個 k_i 值皆等於 1， $i=1, \dots, n$ ，如此一來，機器 i 會享有相同的維修週期。

Lemma 3：對於 $\Psi(k_1, \dots, k_n, T)$ 函數，在 $T > \delta_i(1)$ 時無局部最小值存在。

[Proof]：

給定一組維修乘數向量 \mathbf{k} ，且令 $\Psi(k_1, \dots, k_n, T)$ 函數一階微分為零，得到局部最小值的位置 $\tilde{T}(\mathbf{k})$ 。在 $k_i \geq 1$ ， $i=1, \dots, n$ 時， $\tilde{T}(\mathbf{k}) \leq \delta_i(1)$ ，所以在 $T > \delta_i(1)$ 時無局部最小值存在。其中

$$\tilde{T}(\mathbf{k}) = \left[\left(\frac{e+1}{e} \right) \left(\frac{M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i}}{\sum_{i=1}^n v_i k_i^e} \right) \right]^{\frac{1}{e+1}} \blacksquare \quad (5.1)$$

由 Lemma 3，雖然可以得到在 $\delta_i(1)$ 之上， $\Psi(k_1, \dots, k_n, T)$ 不會有全域極小值發生，但是卻不能保證以 $\delta_i(1)$ 為維修基本週期時，可以滿足維修產能

限制條件。因此，我們將搜尋上界修訂為 $T_U = \max\{\delta_i(1), \sum_{i=1}^n X_i\}$ ，除了保證滿足維修產能限制條件之外，還可以保證在 T_U 之上不會有全域極小值發生。有了搜尋上界，我們將於下一節，訂定演算法搜尋時的下界，作為搜尋演算法的終點，以避免浪費過多的搜尋時間。

5.2 搜尋演算法的下界

在第5.1節中，我們已經確定演算法搜尋範圍的上界，並且也以證明，在上界之上，目標函數不會有任何比滿足維修產能限制的局部最小值更小的函數值出現。同理，本節訂定搜尋時的下界時，除了訂定下界的位置之外，還需確定在下界之下，不會有比任何滿足維修產能限制的局部最小值更小的函數值出現。因此，沿著時間軸 T 進行搜尋時，可以在上下界所界定的範圍內搜尋，以減少演算法搜尋時所需花費的時間。

接下來在第5.2.1小節與第5.2.2小節分別介紹二種搜尋演算法的下界，即 β_1 與 β_2 。開始進行搜尋時，下界為 β_1 ，而當從搜尋上界沿時間軸 T 往左搜尋時，若新得到的成本比目前手上的成本還小時，此時搜尋範圍的下界會進行調整。因此，這個時候搜尋演算法的下界將會取決於 β_1 與 β_2 中較大者，也就是 $T_L = \max\{\beta_1, \beta_2\}$ 。

5.2.1 搜尋演算法的下界 β_1

本小節是根據 Yao(2005)所提出的搜尋下界為基礎，進而推導出本研究之下界 β_1 值。在本問題中，我們希望得到在最佳基本週期 (T^*) 和最佳維修乘數 (\mathbf{k}^*) 下的最佳目標函數 (Ψ^*)，然而當得到 T^* 時就可以找出相對應的 \mathbf{k}^* ，因此可藉由最佳基本週期和最佳目標函數兩者之間的關係，進而推導出剛開始進行搜尋演算法的下界，即 β_1 。

Lemma 4：令 β_1 為在最佳基本週期值 T^* 時的搜尋下界值，而 $\beta_1 = \frac{((e+1)/e)M}{\Psi^U}$ ，其中 Ψ^U 為本問題最佳目標函數值 Ψ^* 之上界。

[Proof]：

給定一組維修乘數向量 \mathbf{k} ，其對應的局部最小值位置為 $\tilde{T}(\mathbf{k})$ ，如下所示：

$$\tilde{T}(\mathbf{k}) = \left[\left(\frac{e+1}{e} \right) \left(\frac{M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i}}{\sum_{i=1}^n v_i k_i^e} \right) \right]^{\frac{1}{e+1}} \quad (5.2)$$

將式(5.2)代入本問題目標函數(式(4.3))，整理後得到最佳目標函數值 Ψ^* ，如下所示：

$$\Psi^* = \left(M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i} \right)^{\frac{e}{e+1}} \left[\left(\sum_{i=1}^n v_i k_i^e \right)^{\frac{1}{e+1}} \left(\frac{e+1}{e} \right)^{\frac{e}{e+1}} \right] + \sum_{i=1}^n f_i \quad (5.3)$$

接著藉由下面的數學推導過程，我們得到 $\Psi^* T^* > ((e+1)/e)M$ ，如下所示：

$T^* \cdot \Psi^*$

$$\begin{aligned} &= \left[\left(\frac{e+1}{e} \right) \left(\frac{M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i}}{\sum_{i=1}^n v_i k_i^e} \right) \right]^{\frac{1}{e+1}} \cdot \left\{ \left(M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i} \right)^{\frac{e}{e+1}} \left[\left(\sum_{i=1}^n v_i k_i^e \right)^{\frac{1}{e+1}} \left(\frac{e+1}{e} \right)^{\frac{e}{e+1}} \right] + \sum_{i=1}^n f_i \right\} \\ &= \left(\frac{e+1}{e} \right) \left(M + \sum_{i=1}^n \frac{(m_i - f_i X_i)}{k_i} \right) + T^* \sum_{i=1}^n f_i > \left(\frac{e+1}{e} \right) M \end{aligned} \quad (5.4)$$

將式(5.4)移項後得到

$$T^* > \frac{((e+1)/e)M}{\Psi^*} \quad (5.5)$$

而 Ψ^U 為本問題最佳目標函數值之上界，所以 $\Psi^U \geq \Psi^*$ ，因此式(5.5)可以改寫為

$$T^* > \frac{((e+1)/e)M}{\Psi^U} \quad (5.6)$$

所以當知道 Ψ^U 值後，我們就可以知道搜尋範圍的下界 β_1 。假設

$$T_0 = \min_i \left\{ \frac{[(e+1)(m_i - f_i X_i)/(e v_i)]^{\frac{1}{e+1}}}{k_i} \right\} \quad (5.7)$$

並且由 T_0 得到相對應的 $\mathbf{k}^*(T_0)$ 值，並計算 $\Psi^U = \Psi(\mathbf{k}^*(T_0), \tilde{T}(\mathbf{k}^*(T_0)))$ 。因此，

可得到初始搜尋下界值 $\beta_1 = \frac{((e+1)/e)M}{\Psi^U}$ 。■

5.2.2 搜尋演算法的下界 β_2

本小節是根據Huang(2004)所提出的搜尋下界為基礎，進而推導出本研究之下界 β_2 值。在透過Theorem 2得到搜尋下界 β_2 前，先說明在搜尋範圍內，搜尋最佳解時的更新動作。令 \tilde{T} 與 \mathbf{k}^* 為目前手邊最好的局部最小值與相對應的最佳維修乘數向量，其中 $\mathbf{k}^* = \{k_1^*(\tilde{T}), k_2^*(\tilde{T}), \dots, k_n^*(\tilde{T})\}$ 。當得到一個新的局部最小值 \tilde{T} 時，首先確定 \tilde{T} 所相對應的目標函數值 $Z(\mathbf{k}(\tilde{T}), \tilde{T})$ ，是否小於目前手邊最好的最佳值 $Z(\mathbf{k}(T^*), T^*)$ ，若 $Z(\mathbf{k}(\tilde{T}), \tilde{T}) < Z(\mathbf{k}(T^*), T^*)$ ，則將手邊的最佳解更新為 $T^* = \tilde{T}$ 且 $\mathbf{k}^* = \mathbf{k}(\tilde{T})$ ，另外，也需將目前手上的下界值進行更新，以縮短沿時間軸 T 進行搜尋所需的時間。因此，可以由Theorem 2得到搜尋下界值，即 β_2 。

Theorem 2: 若 $Z(T)$ 之局部極小值發生在 \tilde{T} ，則搜尋演算法下界值為 β_2 ，如下所示：

$$\beta_2 = \frac{M}{M/\tilde{T} + \sum_{i=1}^n \phi_i(k_i^*(\tilde{T}), \tilde{T})} \quad (5.8)$$

其中

$$\phi_i(k_i^*(\tilde{T}), \tilde{T}) = \begin{cases} \frac{(m_i - f_i X_i)}{\tilde{T}} + \frac{v_i}{e+1} \tilde{T}^e - \left\{ \left(\frac{(e+1)(m_i - f_i X_i)}{e} \right)^e v_i \right\}^{\frac{1}{e+1}}, & k_i^*(\tilde{T}) = 1 \\ \frac{(m_i - f_i X_i)}{k_i \gamma} + \frac{v_i k_i^e}{e+1} \gamma^e - \left\{ \left(\frac{(e+1)(m_i - f_i X_i)}{e} \right)^e v_i \right\}^{\frac{1}{e+1}}, & k_i^*(\tilde{T}) > 1 \end{cases}$$

$$\gamma = \left[\frac{(m_i - f_i X_i)(e+1)}{(k_i + 1)k_i v_i ((k_i + 1)^e - k_i^e)} \right]^{\frac{1}{e+1}}$$

[Proof] :

針對機器 i ，可以畫出其成本曲線圖 $g_i(T) \equiv \inf_{k_i \in N^+} \{\Phi_i(k_i, T) \mid T = T' \in R^+\}$ ，並找出接合點的位置 $\delta_i(k_i)$ 。

$$\text{其中 } \Phi_i(k_i, T) = \frac{(m_i - f_i X_i)}{k_i T} + \frac{v_i k_i^e T^e}{e+1} + f_i。$$

(case_1) $k_i^*(\tilde{T}) = 1$ 時，

也就是在第一個接合點 $\delta_i(1)$ 右邊的區間，如圖 5.1 所示。在此區間內的最低點為

$$\left\{ \left(\frac{(e+1)(m_i - f_i X_i)}{e} \right)^e v_i \right\}^{\frac{1}{e+1}} + f_i \quad (5.9)$$

假設有一時間點 \tilde{T} 落在此區間中，其 $\Phi_i(k_i, T)$ 值為

$$\frac{m_i - f_i X_i}{\tilde{T}} + \frac{v_i \tilde{T}^e}{e+1} + f_i \quad (5.10)$$

所以在此區間裡，最大的成本減少幅度為式(5.10)減式(5.9)，如下所示：

$$\frac{(m_i - f_i X_i)}{\tilde{T}} + \frac{v_i}{e+1} \tilde{T}^e - \left\{ \left(\frac{(e+1)(m_i - f_i X_i)}{e} \right)^e v_i \right\}^{\frac{1}{e+1}} \quad (5.11)$$

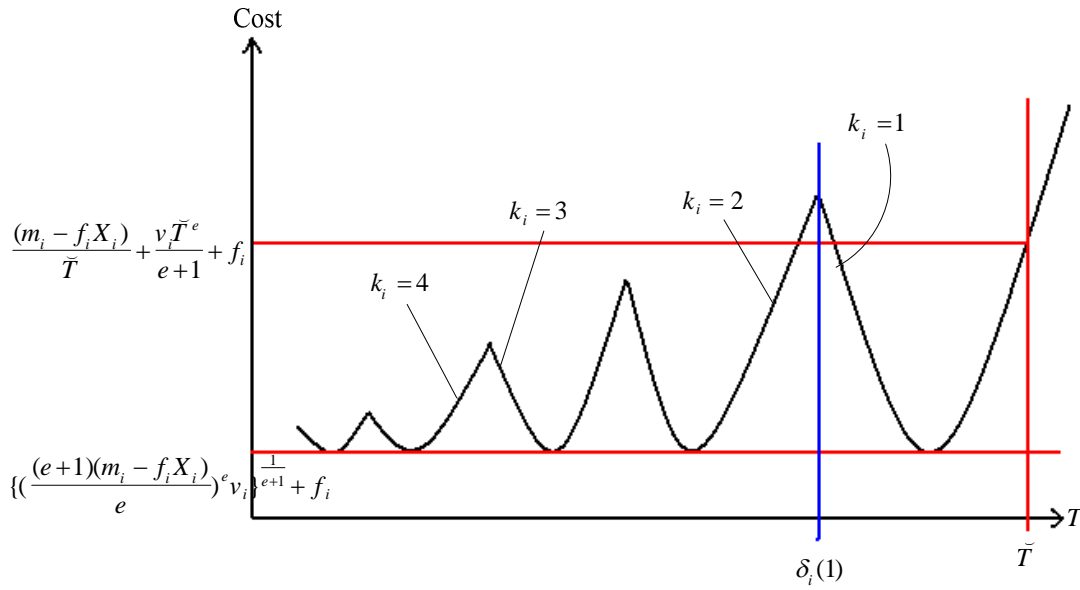


圖 5.1 成本曲線($k_i^*(\tilde{T})=1$)

(case_2) $k_i^*(\tilde{T}) > 1$ 時，

也就是指在第一個接合點 $\delta_i(1)$ 左方的範圍，如圖 5.2 所示。假設有一時間點 \tilde{T} 落在此範圍的某一區間內，其 $\Phi_i(k_i, T)$ 值為 $\Phi_i(k_i^*(\tilde{T}), \tilde{T})$ ，而且 $\Phi_i(k_i^*(\tilde{T}), \tilde{T})$ 值不會比在此區間左右兩邊接合點的 $\Phi_i(k_i, T)$ 值之最大者還大，也就是

$$\Phi_i(k_i^*(\tilde{T}), \tilde{T}) \leq \max\{\Phi_i(k_i^*(\tilde{T})-1, \delta_i(k_i^*(\tilde{T})-1)), \Phi_i(k_i^*(\tilde{T}), \delta_i(k_i^*(\tilde{T})))\} \quad (5.12)$$

觀察圖 5.2 並利用數學歸納法驗證，可以得到在此區間左右兩邊接合點之 $\Phi_i(k_i, T)$ 值，右邊接合點之 $\Phi_i(k_i, T)$ 值會比左邊接合點之 $\Phi_i(k_i, T)$ 值還大，也就是 $\Phi_i(k_i^*(\tilde{T})-1, \delta_i(k_i^*(\tilde{T})-1)) > \Phi_i(k_i^*(\tilde{T}), \delta_i(k_i^*(\tilde{T})))$ 。證明過程參見附錄 A 所示。因此，由數學歸納法得到，不管 \tilde{T} 落在第一個接合點 $\delta_i(1)$ 左

方範圍的任何一個區間內，其 $\Phi_i(k_i, T)$ 值皆會小於等於 $\Phi_i(k_i(\bar{T})-1, \delta_i(k_i(\bar{T})-1))$ 值。

另外，在此範圍的任何一個區間的最低點依然為 $\{(\frac{(e+1)(m_i - f_i X_i)}{e})^e v_i\}^{\frac{1}{e+1}} + f_i$ 。所以，我們可以得到在此範圍的任何一個區間內，最大的成本減少幅度為

$$\frac{(m_i - f_i X_i)}{k_i \gamma} + \frac{v_i k_i^e}{e+1} \gamma^e - \{(\frac{(e+1)(m_i - f_i X_i)}{e})^e v_i\}^{\frac{1}{e+1}} \quad (5.13)$$

另一方面，我們是沿時間軸 T 由搜尋上界往下進行搜尋，所以單位時間主要整備成本 $(\frac{M}{T})$ 是隨時間遞減而遞增。因此，當 \bar{T} 往 $T < \bar{T}$ 進行時，所有機器最大的成本減少幅度之總和不會比單位時間主要整備成本增加幅度還大，也就是 $\frac{M}{T} - \frac{M}{\bar{T}} \geq \sum_{i=1}^n \phi_i(k_i^*(\bar{T}), \bar{T})$ 。 β_2 就是在最大減少幅度與增加幅度拉扯中，取得平衡的時間點。即在 $\frac{M}{T} - \frac{M}{\bar{T}} \geq \sum_{i=1}^n \phi_i(k_i^*(\bar{T}), \bar{T})$ 情形下，

$$\frac{M}{\frac{M}{\bar{T}} + \sum_{i=1}^n \phi_i(k_i^*(\bar{T}), \bar{T})} \geq T \text{ 成立，其可得 } \beta_2 = \frac{M}{\frac{M}{\bar{T}} + \sum_{i=1}^n \phi_i(k_i^*(\bar{T}), \bar{T})} \text{。} \blacksquare$$

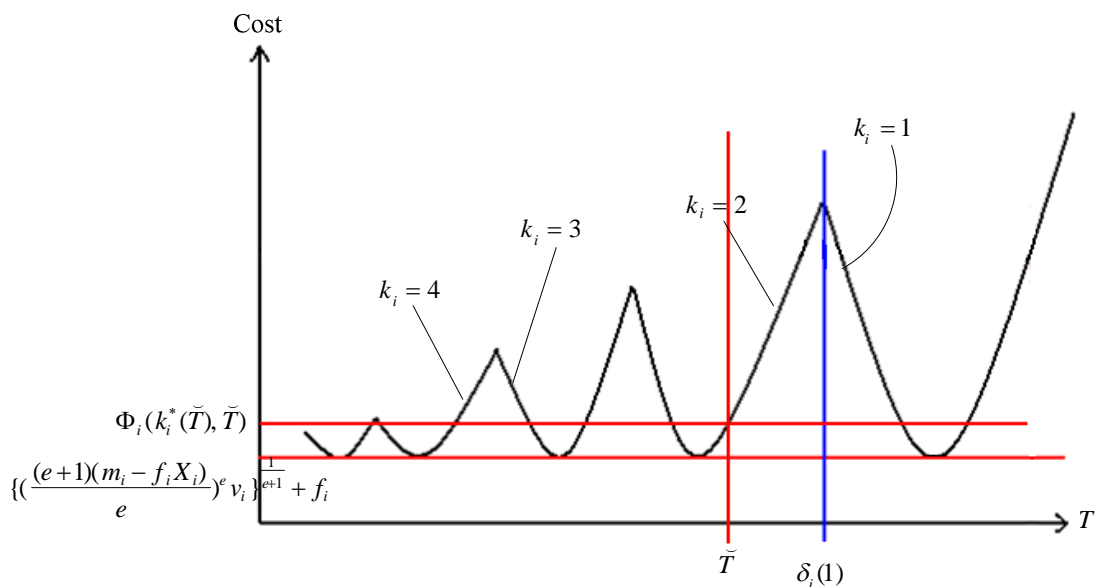


圖 5.2 成本曲線 $(k_i^*(\bar{T}) > 1)$

5.3 運用接合點進行搜尋

由第四章理論分析的 Proposition 1 和 Proposition 2 可知，我們可藉由接合點進行搜尋，並且可透過接合點的性質去改變維修乘數，以獲得 $\Gamma(T)$ 函數的最佳值。

本問題之搜尋演算法的設計概念是以不考慮維修產能限制的片段凸性曲線之接合點為搜尋基礎，並利用可行解測試法(The Feasibility Testing Procedure, Proc. FT)驗證目前搜尋的解是否滿足維修產能限制。可行解測試方法將在第六章進行說明。

整個搜尋演算法的架構，概述如下：首先得到搜尋上界值 T_U ，並找到離 T_U 最近的下一個接合點位置 w_{p+1} 和相對應之維修乘數 $K(w_{p+1})$ 後，針對此組 $(K(w_{p+1}), w_{p+1})$ 進行可行解測試。當此組 $(K(w_{p+1}), w_{p+1})$ 為可行解時，表示此組 $(K(w_{p+1}), w_{p+1})$ 滿足維修產能限制條件；反之，當此組 $(K(w_{p+1}), w_{p+1})$ 為不可行解時，表示此組 $(K(w_{p+1}), w_{p+1})$ 不滿足維修產能限制條件。因此，當 $(K(w_{p+1}), w_{p+1})$ 為不可行解時，就要測試在下一個接合點 w_{p+1} 與上一個接合點之相對維修乘數 $K(w_p)$ 這組 $(K(w_p), w_{p+1})$ 是否為可行解，當 $(K(w_p), w_{p+1})$ 依然為不可行解時，我們允許調整一台成本增加幅度最小的機器之維修乘數，使其滿足維修產能限制。每當得到一組可行維修排程解時，我們要與手邊最好的成本值進行比較，若新得到的成本值較低則手邊最好值被取代，並且重新計算搜尋下界值；反之則不用進行取代。反覆上述的過程，直到目前搜尋的時間點小於搜尋範圍的下界時才停止搜尋，並得到本問題的最佳解。

本研究所提出之搜尋演算法，在搜尋過程中可能遭遇三種情況，茲說明如下：假設目前搜尋的接合點為 w_p ，相對應之維修乘數為 $K(w_p)$ ； w_{p+1} 為離 w_p 最近的下一個接合點，相對應之維修乘數為 $K(w_{p+1})$ 。

Case 1：測試 $(K(w_{p+1}), w_{p+1})$ 是否為可行解：

- (a) 可行解，判斷成本大小並繼續找下一個接合點。
- (b) 不可行解，進入 Case 2

Case 2：測試 $(K(w_p), w_{p+1})$ 是否為可行解：

- (a) 可行解，判斷成本大小並繼續找下一個接合點。

(b) 不可行解，進入 Case 3

Case 3：在 $K(w_p)$ 中，找一台機器的維修乘數減 1 之後，其成本增加幅度最小。回到 Case 2。

完整搜尋演算法演算步驟程序說明如下，演算細部流程圖如圖 5.3 所示。

Step 1：計算搜尋上下界及目前所得之最佳解(best-on-hand)。

1.1 計算上界 $T_U = \max\{\delta_i(1), \sum_{i=1}^n X_i\}$ 與下界 $T_L = \beta_1$ 。

1.2 令 $K^* = (1, \dots, 1)$, $T^* = T_U$ 與 $TC^* = (K^*, T^*)$ 。

1.3 令 $p = \arg \min_i \{\delta_i(k_i) \geq T_U\}$ 。

Step 2：找離 w_p 最近的接合點之機器下標 $\pi = \arg \max_i \{\delta_i(k_i) < w_p\}$ ，並計算時間點 w_{p+1} 之維修乘數 $K(w_{p+1}) = (K(w_p) \setminus \{k_\pi\} \cup \{k_\pi + 1\})$ 。

Step 3：判斷 $w_{p+1} < T_L$ ，是則進入 Step 7。否則進入 Step 4。

Step 4：判斷 $(K(w_{p+1}), w_{p+1})$ 是否為可行解，是則進入 4.1。否則進入 Step 5。

4.1 計算 $T = \max(\tilde{T}(K(w_{p+1})), L)$ 與 $TC(K(w_{p+1}), T)$ 。

4.2 判斷 $TC^* \geq TC(K(w_{p+1}), T)$ ，是則進入 4.2.1。否則進入 4.2.2。

4.2.1 變更手上最好的解為 $K^* = K(w_{p+1})$, $T^* = T$ 與 $TC^* = TC(K^*, T^*)$

及下界變更為 $T_L = \max\{T_L, \beta_2\}$ 。

4.2.2 不變更手上最好的解。

4.3 令 $w_p = w_{p+1}$ 與 $K(w_p) = K(w_{p+1})$ 。回到 Step 2。

Step 5：判斷 $(K(w_p), w_{p+1})$ 是否為可行解，是則進入 5.1。否則進入 Step 6。

5.1 計算 $T = \max(\tilde{T}(K(w_p)), L)$ 與 $TC(K(w_p), T)$ 。

5.2 判斷 $TC^* \geq TC(K(w_p), T)$ ，是則進入 5.2.1。否則進入 5.2.2。

5.2.1 變更手上最好的解為 $K^* = K(w_p)$, $T^* = T$ 與 $TC^* = TC(K^*, T^*)$ 及

下界變更為 $T_L = \max\{T_L, \beta_2\}$ 。

5.2.2 不變更手上最好的解。

5.3 令 $w_p = w_{p+1}$ 與 $K(w_p) = K(w_p)$ 。回到 Step 2。

Step 6: 判斷是否有維修乘數大於 1 之機器，是則進入 6.1。否則進入 6.2。

6.1 找出成本增加幅度最小之機器下標

$\lambda = \arg \min_i \{TC_i(k_i - 1, w_{p+1}) - TC_i(k_i, w_{p+1})\}$ ，並計算時間點 w_{p+1} 之維修乘數 $K(w_p) = (K(w_p) \setminus \{k_\lambda\} \cup \{k_\lambda - 1\})$ 。回到 Step 5。

6.2 計算 $T = \max(\tilde{T}(K(w_p)), L)$ 與 $TC(K(w_p), T)$ 。

6.2.1 判斷 $TC^* \geq TC(K(w_p), T)$ ，是則令 $K^* = K(w_p), T^* = T$ 與

$TC^* = TC(K(w_p), T)$ 。否則進入 Step 7。

Step 7: 輸出最佳解 (K^*, T^*) 與對應的最小成本值 TC^* 並停止演算程序。

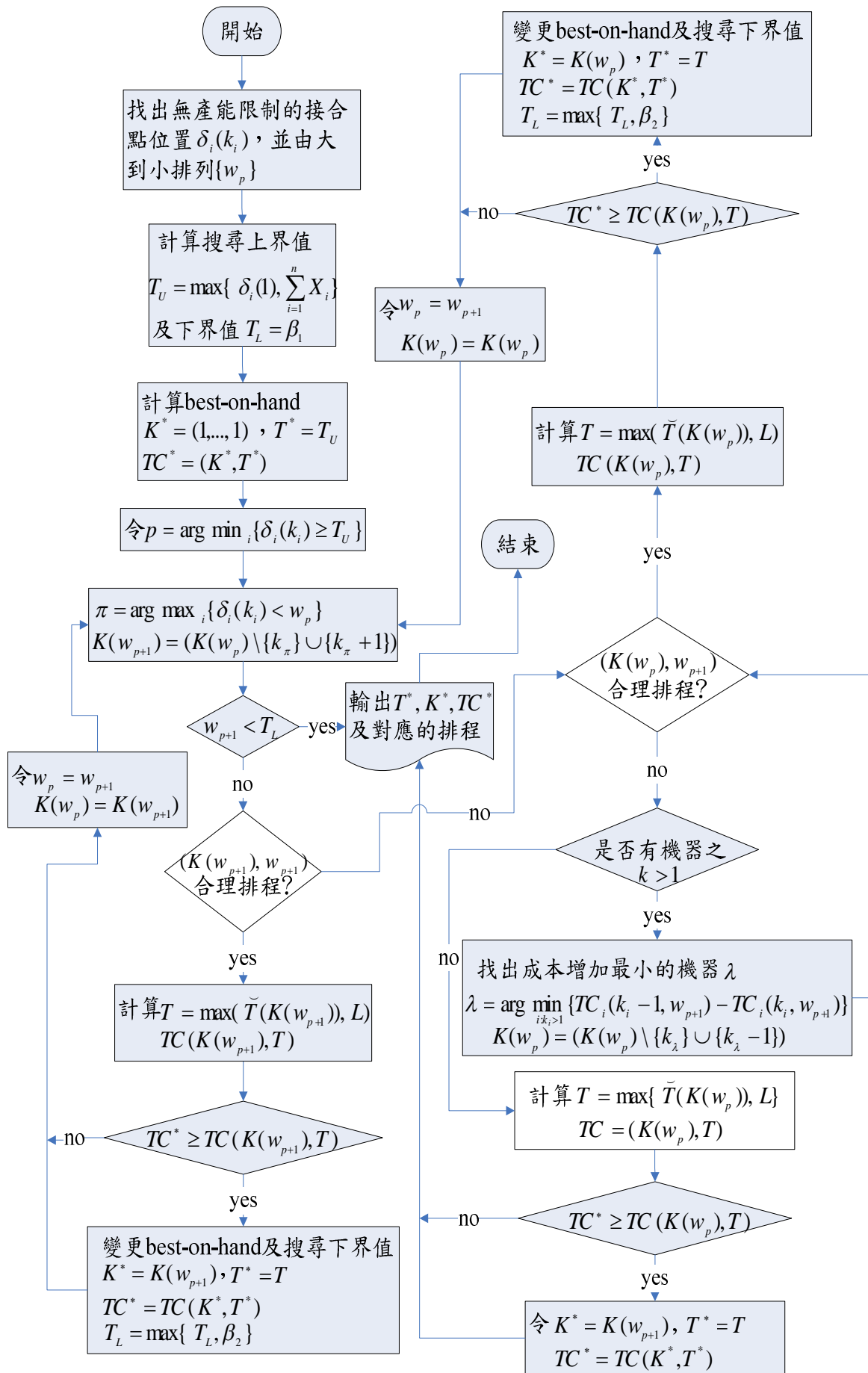


圖 5.3 搜尋演算法流程圖

第六章 可行解測試演算法

在第三章數學模式的建立，因為考慮了維修產能限制條件，因此式(3.2)提到在每一個基期上有進行維修工作的機器之整個維修工作時間的總和必須要小於或等於基本週期。經由第五章提出的搜尋演算法我們可得到一組 $(\{k_i\}, T)$ ，但該組解必須套回本問題之數學模式的限制式才能說明該組解是否為本問題之最佳解。因此，在本章節我們利用Yao(2002)所提出的可行解測試方法(The Feasibility Testing Procedure, Proc. FT)，來判斷某組 $(\{k_i\}, T)$ 是否為可行解。Proc. FT 是以延伸基本週期法(Extended Basic Period, EBP)為基礎，而提出的一套啟發式方法，說明如何安排及調整每個產品的生產週期，進而判斷該組解是否為可行解。Proc. FT對於不可行的解，都要經過多次調整之後，才能認定為不可行解。若在調整過程中，有符合每個基期的總維修時間皆小於或等於基本週期的限制條件，則表示為可行解，即可停止調整的動作。

Proc. FT 演算步驟程序說明如下，演算細部流程圖如圖6.1所示。

Step 1：設定初始條件。

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

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

2.1 令 $W^m = W$ 和 $L(W^m) = L(W)$ 。

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

2.3 判斷 $L(W^m) \leq T$ ，是則令 $L(W^F) = L(W^m)$ ， $W^F = W^m$ 與 $\varphi = 1$ ，並進入Step4。否則進入Step 3。

Step 3：應用調整週期的方式改善 W^m 的 $L(W^m)$ 。

3.1 使用移出程序來改善。

3.1.1 判斷 $L(W^m) \leq T$ ，是則令 $\varphi = 1$ ， $L(W^F) = L(W^m)$ 和 $W^F = W^m$ ，並進入Step 4。否則進入Step 3.1.2。

3.1.2 判斷 $L(W^m) > T$ 和 $L(W^m)$ 是否已被改善，是則回到Step 3。否則進入Step 3.2。

3.2 使用1對1互換程序來改善。

3.2.1 判斷 $L(W^m) \leq T$ ，是則令 $\varphi = 1, L(W^F) = L(W^m)$ 和 $W^F = W^m$ ，並進入Step 4。否則進入Step 3.2.2。

3.2.2 判斷 $L(W^m) > T$ 和 $L(W^m)$ 是否已被改善，是則回到Step 3。否則進入Step 3.3。

3.3 使用2對1互換程序來改善。

3.3.1 判斷 $L(W^m) \leq T$ ，是則令 $\varphi = 1, L(W^F) = L(W^m)$ 和 $W^F = W^m$ ，並進入Step 4。否則進入Step 3.3.2。

3.3.2 判斷 $L(W^m) > T$ 和 $L(W^m)$ 是否已被改善，是則回到Step 3。否則進入Step 3.4。

3.4 判斷 $L^*(W^*)$ 是否已經改善。

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

3.4.2 判斷 $L(W^m) \geq L^*(W^*)$ ，是則令 $\chi = \chi + 1$ ，並進入Step 4。

Step 4：判斷Proc. FT 程序是否已經達到終止條件。

4.1 判斷 $\varphi = 1$ ，是則停止Proc. FT演算，並輸出 W^F 和 $L(W^F)$ 。

4.2 判斷 $\chi > Y$ 且 $\varphi = 0$ ，是則停止Proc. FT演算，並輸出”無可行解維修排程”。

4.3 判斷 $\chi \leq Y$ 且 $\varphi = 0$ ，是則隨機選取 $\lfloor n/2 \rfloor$ 台機器重新排程並標記為 \bar{F} ，令 $\gamma = 1$ ，並進入Step 2再次改善。

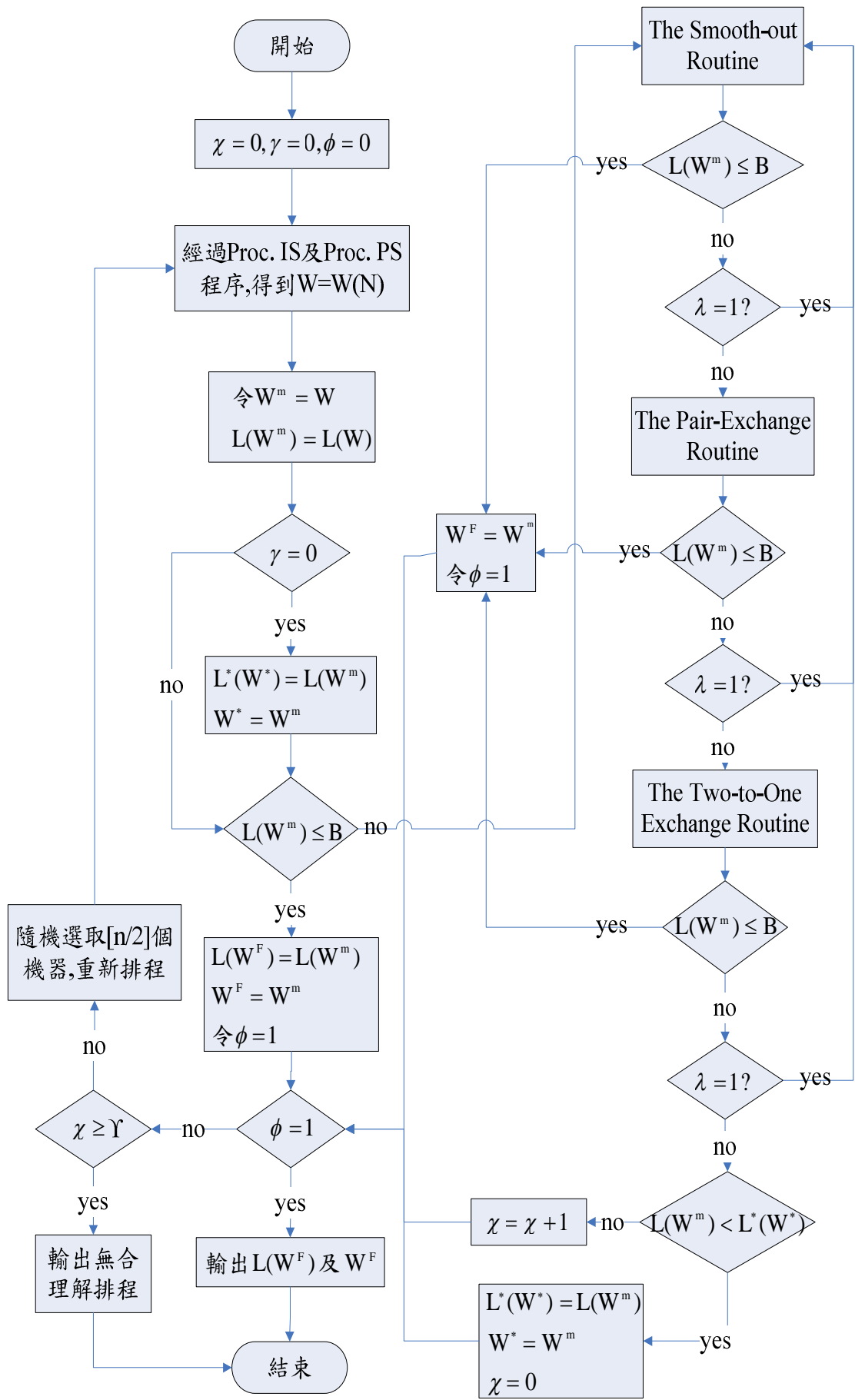


圖 6.1 可行解測試演算法流程圖

6.1 設定初始條件

1. 設定門檻值(Υ)：作為運算結束的標準值， Υ 值愈大，找到可行解的機率就愈大，但運算所花費的時間相對地會愈多。
2. 設定計次器 $\chi = 0$ ：表連續使用Proc. FT 程序且無法改善 L^* 的次數，若 $\chi > \Upsilon$ ，則停止運算。
3. 設定可行解 $\varphi = 0$ ：表示排程為不可行，但若經Proc. FT方法調整其排程，並且使得每個基期所有機器的總維修時間皆小於或等於 T ，則表示其排程是可行的，此時 $\varphi = 1$ ，並停止運算。

6.2 初始排程(Proc. IS)與機器排程(Proc. PS)

每台機器是依其 k 值及第一個開始維修基期，來決定所有基期中，那個基期要進行維修，那個基期不維修（即該機器在該基期的維修時間為0）。在進行維修排程時，一個機器接一個機器加入維修排程，每次之維修排程皆以該次所有基期負荷最小的值為主，且其總基期數為參與維修排程的機器 k_i 之最小公倍數。若所有機器排完後，每個基期所有機器的總維修時間皆小於或等於 T ，則表示其維修排程是可行的，則停止運算，否則進入下列調整週期的方式。Proc. IS演算步驟程序說明如下，演算細部流程圖參閱附錄B-1所示。

Step 1：檢驗 γ 的值。

- 1.1 判斷 $\gamma = 0$ ，是則令 $\mathcal{G} = \emptyset$ 和 $W(\mathcal{G}) = \emptyset$ ，並進入Step 2。
- 1.2 判斷 $\gamma > 0$ ，是則令 $\mathcal{G} = F$ 和 $W(\mathcal{G}) \equiv W(F) = W_0(F)$ ，即將原先 W_0 排程中之 F 機器集合排程維持不變，並進入Step 2。

Step 2：優先指派 k_i 乘數為1的機器進入排程。

- 2.1 對所有未指派到排程的機器 \bar{F} ，若其 k_i 乘數為1，則優先指派 X_i 到 $W(\mathcal{G})$ 排程中，並更新 $\mathcal{G} = \mathcal{G} \cup \{j \mid j \in \bar{F} \text{ and } k_j = 1\}$ 。假使所有 \bar{F} 中的 k_j 均大於1，則略過Step 2。

Step 3：將 $i \in (N - \mathcal{G})$ 的 X_i 由大到小進行降冪排列，並設 Ξ 代表排序 X_i 後的機器集合之識別子。

Step 4：檢驗在 Ξ 集合中的值：

4.1 判斷 $\Xi \neq \phi$ ，是則進入Step 4.1.1。

4.1.1 從 Ξ 集合中得到機器 \hat{i} ，且 $\hat{i} = \operatorname{argmax}_{j \in \Xi} \{\sigma_j\}$ 。

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

4.1.3 將 \mathcal{G} 和 Ξ 集合更新為 $\mathcal{G} = \mathcal{G} \cup \{\hat{i}\}$ 和 $\Xi = \Xi - \{\hat{i}\}$ ，並回到 Step 4。

4.2 判斷 $\Xi = \phi$ ，是令 $W = W(N)$ ，並停止 Proc. IS 演算程序。

Proc. PS 演算步驟程序說明如下，演算細部流程圖參閱附錄B-2所示。

Step 1：檢驗 γ 的值。

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

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

Step 2：計算所有 k_i 的最小公倍數， $K = \operatorname{lcm}\{k_1, k_2, \dots, k_N\}$ 。

Step 3：計算 R_t 值。

3.1 R_t 為在初始部分維修排程 $W(\mathcal{G})$ 中的第 t 個維修基期之目前總維修時間， $t = 1, \dots, K$ 。

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

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

Step 4：選擇 w_{il}^* 。

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

我們以簡單的範例來說明 Proc. IS 和 Proc. PS 演算程序所代表的意義。假設各機器的資料如表 6-1 所示。

表 6.1 機器基本資料(Proc. IS 和 Proc. PS 範例)

機器	1	2
k 值	2	3
維修時間 X_i	5	10

初始排程和機器排程之過程如下：

1. 將機器依 X_i 的大小由大至小排序，結果機器2先排，再排機器1。
2. 機器2加入排程，因為 k_1 及 k_2 的最小公倍數為6，所以排至基期6，如表6-2所示。

表 6.2 機器 2 加入排程(Proc. IS 和 Proc. PS 範例)

	機器2	總維修時間
k 值	3	
基期1	10	10
基期2	0	0
基期3	0	0
基期4	10	10
基期5	0	0
基期6	0	0

3. 機器1加入排程。
4. 將機器1的維修時間填入所有基期內，如表6-3所示。
5. 合計目前每個基期的總維修時間，如表6-3所示。

表 6.3 機器 1 加入排程(Proc. IS 和 Proc. PS 範例)

	機器1	機器2	總維修時間
k 值	2	3	
基期1	5	10	15
基期2	5	0	5
基期3	5	0	5
基期4	5	10	15
基期5	5	0	5
基期6	5	0	5

6. 由表6-3知，基期2為總維修時間最小的第1個，所以機器1以基期2開始排程，並且以2個基期維修一次的方式往下排，結果如表6-4所示。

表 6.4 結果(Proc. IS 和 Proc. PS 範例)

	機器1	機器2	總維修時間
k 值	2	3	
基期1	0	10	10
基期2	5	0	5
基期3	0	0	0
基期4	5	10	15
基期5	0	0	0
基期6	5	0	5

6.3 調整週期的方式

在執行調整週期演算程序中，本研究使用了一些新的參數，其定義如下所示：

τ ：擁有最大總維修時間的維修基期。

W^m ：在每次個別執行Proc. FT程序及區域搜尋或重複改善 L^* 的過程，所得到的最小化總維修時間的維修排程。

$L(W^m)$ ： W^m 維修排程中，最大的總維修時間。

W^* ：第一次使用Proc. IS所得到的初始維修排程設為最佳維修排程。

$L(W^*)$ ： W^* 維修排程中，最大的總維修時間。

W ：一種可能的排程結果。

$L(W)$ ：使用 W 排程後，在所有維修基期中，最大的總維修時間。

λ ：假如 $L(W^m)$ 已被改善，則 $\lambda = 1$ ；否則 $\lambda = 0$ 。

$\Gamma(\lambda, W^m)$ ： $\{i | w_{i, \varphi(k_i, \lambda)} = 1 \text{ and } k_i > 1\}$ ，即位在最大總維修時間的維修基期上進行維修且 k_i 乘數大於1的機器集合。

$\Lambda(\lambda, W^m)$ ： $\{i | w_{i, \varphi(k_i, \lambda)} = 0 \text{ and } k_i > 1\}$ 即非位在最大總維修時間的維修基期上

進行維修且 k_i 乘數大於1的機器集合。

λ ：連續使用Proc. FT程序且無法改善 L^* 的次數。

Proc. FT共有移出、1對1互換及2對1互換三種調整週期的方式，詳細演算步驟程序分別說明如下。

6.3.1 移出(The Smooth-Out Routine)

從總維修時間最大的基期中，選1個 k_i 大於1且維修時間最大的機器，移至其他總維修時間較小的基期中。若移出後，每個基期的總維修時間皆小於或等於基本週期，則為可行解，即可停止運算，否則要進入下列1對1互換的方式。移出演算步驟程序說明如下，演算細部流程圖參閱附錄B-3所示。

Step 1：將所有屬於 $\Gamma(\lambda, W^m)$ 的機器 i 之 X_i 由大到小進行降冪排列且標示為未受試驗的機器。

Step 2：檢驗是否有任何在 $\Gamma(\lambda, W^m)$ 中的機器是未受試驗的。

2.1 判斷 $\Gamma(\lambda, W^m)$ 是否有未受試驗的機器，是則找出第一個未受試驗的機器 i ，且將機器 i 使用Proc. PS重新排程以獲得一個較小的維修時間 $L(W)$ 和其對應之排程 W 。更新 $\Gamma(\lambda, W^m) = \Gamma(\lambda, W^m) - \{i\}$ 。進入Step 3

2.2 判斷 $\Gamma(\lambda, W^m) = \phi$ ，是表示 $\Gamma(\lambda, W^m)$ 所有機器均已受過試驗，停止移出演算程序。

Step 3：檢驗 $L(W^m)$ 是否已經改善：

3.1 判斷 $L(W) < L(W^m)$ ，是則令 $L(W^m) = L(W)$, $W^m = W$ 和 $\lambda = 1$ 並停止移出演算程序。

判斷 $L(W) \geq L(W^m)$ ，是則標記機器 i 為已受試驗的，並回到Step 2。

我們以簡單的範例來說明移出演算程序所代表的意義。假設 $T=17$ ，但調整前最大的總維修時間為21（即基期5）大於 T ，所以為不可行解，其移出改善前的維修排程如表6-5所示。將位於最大總維修時間(即基期5)上進行維修且 $k_i > 1$ 的機器集合，按其機器維修時間由大到小進行降冪排序後可得到 $\Gamma(\lambda, W^m) = \{3, 2, 1\}$ 。因此，優先將機器3開始維修基期由基期1移至總維修時間最小的基期4，再重新排程，調整後每個基期的總維修時間皆小於 T （最大總維修時間變為16），如表6-6所示，所以為一合理可行的排程解。

表 6.5 改善前(The Smooth-Out Routine 範例)

	機器1	機器2	機器3	總維修時間
k 值	2	3	4	
基期1	5	0	10	15
基期2	0	6	0	6
基期3	5	0	0	5
基期4	0	0	0	0
基期5	5	6	10	21
基期6	0	0	0	0
基期7	5	0	0	5
基期8	0	6	0	6
基期9	5	0	10	15
基期10	0	0	0	0
基期11	5	6	0	11
基期12	0	0	0	0

表 6.6 改善後(The Smooth-Out Routine 範例)

	機器1	機器2	機器3	總維修時間
k 值	2	3	4	
基期1	5	0	0	5
基期2	0	6	0	6
基期3	5	0	0	5
基期4	0	0	10	10
基期5	5	6	0	11
基期6	0	0	0	0
基期7	5	0	0	5
基期8	0	6	10	16
基期9	5	0	0	5
基期10	0	0	0	0
基期11	5	6	0	11
基期12	0	0	10	10

6.3.2 1對1互換(The Pair-Exchange Routine)

從總維修時間最大的基期中，選1個 k_i 大於1且維修時間最大的機器，和其他基期中維修時間較小且 k_i 大於1之機器互換。若1對1互換後，每個基期的總維修時間皆小於或等於基本週期 T ，則為可行解，即可停止運算，否則要進入下列2對1互換的方式。1對1互換演算步驟程序說明如下，演算細部流程圖參閱附錄B-4所示。

Step 1：將 $i \in \Gamma(\lambda, W^m)$ 中的 X_i 由大到小進行降冪排列，且 $j \in \Lambda(\lambda, W^m)$ 中的 X_j 由小到大進行升冪排列。並標記 $i \in \Gamma(\lambda, W^m)$ 和 $j \in \Lambda(\lambda, W^m)$ 中的機器為未受試驗的。

Step 2：檢驗是否有任何在 $\Gamma(\lambda, W^m)$ 中的機器是為未受試驗的。

2.1 判斷 $\Gamma(\lambda, W^m)$ 是否有未受試驗的機器，是則找出第一個未受試驗的機器 i ，並進入Step 3。

2.2 判斷 $\Gamma(\lambda, W^m) = \phi$ ，是則表示 $\Gamma(\lambda, W^m)$ 所有機器均已受過試驗，停止1對1互換演算程序。

Step 3：檢驗是否有任何在 $\Lambda(\lambda, W^m)$ 中的機器是為未受試驗的。

3.1 判斷 $\Lambda(\lambda, W^m)$ 是否有未受試驗的機器，是則找出第一個未受試驗的機器 j ，進入Step 4。

3.2 判斷 $\Lambda(\lambda, W^m)$ 是否均受過試驗，是則將 $\Lambda(\lambda, W^m)$ 集合中的機器重新標記為未受試驗的且將機器 i 標記為已受試驗的，回到Step 2。

Step 4：比較 X_i 和 X_j 。

4.1 判斷 $X_i > X_j$ ，是則進入Step 4.1.1。

4.1.1 將機器 i 從 W^m 排程中移出，並指派機器 j 的排程使其在標記 λ 的維修基期上進行維修。

4.1.2 將機器 i 使用 Proc. PS 重新排程以獲得一個較小的維修時間 $L(W)$ 和其對應之排程 W 。

4.1.3 檢驗 $L(W^m)$ 是否已經改善：

A. 判斷 $L(W) < L(W^m)$ ，是則令 $L(W^m) = L(W)$, $W^m = W$ 和 $\lambda = 1$ ，並停止1對1互換演算程序。

B. 判斷 $L(W) \geq L(W^m)$ ，是則標記機器 j 為已受試驗的，並回到Step 3。

4.2 判斷 $X_i < X_j$ ，是則回到Step 3。

我們以簡單的範例來說明1對1互換演算程序所代表的意義。假設 $T=13$ ，但調整前最大的總維修時間為15（即基期2、6、10）大於 T ，所以該排程為不可行解，其1對1互換改善前的維修排程如表6-7所示。將位於最大總維修時間(即基期2、6、10)上進行維修且 $k_i > 1$ 的機器集合，按其機器維修時間由大到小進行降冪排序後可得到 $\Gamma(\lambda, W^m) = \{2,1\}$ ，而未在基期2上維修且 $k_i > 1$ 的機器集合，按其機器維修時間由小到大進行升冪排序後可得到 $\Lambda(\lambda, W^m) = \{3\}$ 。因此，將機器2和機器3開始維修基期互換，再重新排程，調整後每個基期的總維修時間皆小於 T （最大總維修時間變為10），如表6-8所示，所以為一合理可行的排程解。

表 6.7 改善前(The Pair-Exchange Routine 範例)

	機器1	機器2	機器3	總維修時間
k 值	2	4	2	
基期1	0	0	2	2
基期2	5	10	0	15
基期3	0	0	2	2
基期4	5	0	0	5
基期5	0	0	2	2
基期6	5	10	0	15
基期7	0	0	2	2
基期8	5	0	0	5
基期9	0	0	2	2
基期10	5	10	0	15
基期11	0	0	2	2
基期12	5	0	0	0

表 6.8 改善後(The Pair-Exchange Routine 範例)

	機器1	機器2	機器3	總維修時間
k 值	2	4	2	
基期1	0	10	0	10
基期2	5	0	2	7
基期3	0	0	0	0
基期4	5	0	2	7
基期5	0	10	0	10
基期6	5	0	2	7
基期7	0	0	0	0
基期8	5	0	2	7
基期9	0	10	0	10
基期10	5	0	2	7
基期11	0	0	0	0
基期12	5	0	2	7

6.3.3 2對1互換(The Two-to-One Exchange Routine)

從總維修時間最大的基期中，選1個 k_i 大於1且維修時間最大的機器，和其他基期中維修時間較小且 k_i 大於1之2種機器互換。若2對1互換後，每個基期的總維修時間皆小於或等於基本週期，則為可行解，即可停止運算，否則要進入下列隨機選取 $[n/2]$ 台機器重新排程的方式。2對1互換演算步驟程序說明如下，演算細部流程圖參閱附錄B-5所示。

Step 1：將 $i \in \Gamma(\lambda, W^m)$ 中的 X_i 由大到小進行降冪排列，且 $j, l \in \Lambda(\lambda, W^m)$ 中的 $X_j + X_l$ 由小到大進行升冪排列。並標記 $i \in \Gamma(\lambda, W^m)$ 和 $j, l \in \Lambda(\lambda, W^m)$ 中的機器為未受試驗的。

Step 2：檢驗是否有任何在 $\Gamma(\lambda, W^m)$ 中的機器為未受試驗的。

2.1 判斷 $\Gamma(\lambda, W^m)$ 是否有未受試驗的機器，是則找出第一個未受試驗的機器 i ，並進入 Step 3。

2.2 判斷 $\Gamma(\lambda, W^m) = \phi$ ，是則表示 $\Gamma(\lambda, W^m)$ 所有機器均已受過試驗，停止 2 對 1 互換演算程序。

Step 3：檢驗是否有成對機器 j 和 l 在 $\Lambda(\lambda, W^m)$ 中是為未受試驗的。

3.1 判斷 $\Lambda(\lambda, W^m)$ 是否有未受試驗的成對機器 j 和 l ，是則找出第一個未受試驗的成對機器 j 和 l ，進入 Step 4。

3.2 判斷 $\Lambda(\lambda, W^m)$ 是否均受過試驗，是則將 $\Lambda(\lambda, W^m)$ 集合中的機器重新標記為未受試驗的且將機器 i 標記為已受試驗的，回到 Step 2。

Step 4：比較 X_i 和 $X_j + X_l$ 。

4.1 判斷 $X_i > X_j + X_l$ ，是則進入 Step 4.1.1。

4.1.1 將機器 i 從 W^m 排程中移出，並指派成對機器 j 和 l 的排程使其在標記 λ 的維修基期上進行維修。

4.1.2 將機器 i 使用 Proc. PS 重新排程以獲得一個較小的維修時間 $L(W)$ 和其對應之排程 W 。

4.1.3 檢驗 $L(W^m)$ 是否已經改善：

A. 判斷 $L(W) < L(W^m)$ ，是則令 $L(W^m) = L(W)$, $W^m = W$ 和 $\lambda = 1$ ，並停止 2 對 1 互換演算程序。

B. 判斷 $L(W) \geq L(W^m)$ ，是則標記成對機器 j 和 l 為已受試驗的，並回到 Step 3。

4.2 判斷 $X_i < X_j + X_l$ ，是則回到 Step 3。

我們以簡單的範例來說明2對1互換演算程序所代表的意義。假設 $T=13$ ，但調整前最大的總維修時間為15（即基期1、5、9）大於 T ，所以為不可行解，其2對1互換改善前的維修排程如表6-9所示。將位於最大總維修時間(即基期1、5、9)上進行維修且 $k_i > 1$ 的機器集合，按其機器維修時間由大到小進行降冪排序後可得到 $\Gamma(\lambda, W^m) = \{2,1\}$ ，而未在基期1上進行維修且 $k_i > 1$ 的機器所形成的成對集合中，按成對機器維修總時間由小到大進行升冪排序後可得到 $\Lambda(\lambda, W^m) = \{3,4\}$ 。因此，將機器3及機器4同時和機器2之開始維修基期互換，再重新排程，調整後每個基期的總維修時間皆小於 T （最大總維修時間變為12），如表6-10所示，所以為一合理可行的排程解。

表 6.9 改善前(The Two-to-One Exchange Routine 範例)

	機器1	機器2	機器3	機器4	總維修時間
k 值	2	4	2	2	
基期1	5	10	0	0	15
基期2	0	0	2	5	7
基期3	5	0	0	0	5
基期4	0	0	2	5	7
基期5	5	10	0	0	15
基期6	0	0	2	5	7
基期7	5	0	0	0	5
基期8	0	0	2	5	7
基期9	5	10	0	0	15
基期10	0	0	2	5	7
基期11	5	0	0	0	5
基期12	0	0	2	5	7

表 6.10 改善後(The Two-to-One Exchange Routine 範例)

	機器1	機器2	機器3	機器4	總維修時間
k 值	2	4	2	2	
基期1	5	0	2	5	12
基期2	0	10	0	0	10
基期3	5	0	2	5	12
基期4	0	0	0	0	0
基期5	5	0	2	5	12
基期6	0	10	0	0	10
基期7	5	0	2	5	12
基期8	0	0	0	0	0
基期9	5	0	2	5	12
基期10	0	10	0	0	10
基期11	5	0	2	5	12
基期12	0	0	0	0	0

6.3.4 隨機選取 $[n/2]$ 台機器重新排程

如果經過移出、1對1互換及2對1互換後還找不到可行解，則從 n 台機器中，隨機選取 $[n/2]$ 台機器重新排程，其餘機器的排程固定，再重覆上述6.2~6.3的步驟，直到 $X > Y$ 為止。即找 $[n/2]$ 台機器來作局部的最佳化排程，以使得維修排程的最大總維修時間 $L(W)$ 能減少至可行解。

6.4 輸出

1. 找到可行解：在整個Proc. FT可行解的判斷過程中，只要找到可行解，即可停止程式的運算，並輸出可行解的資料，如：可行解的排程、總成本、 $(\{k_i\}, T)$ 及運算時間等資料。
2. 找不到可行解：若可行解的判斷次數超過 Y ，則停止程式的運算，並輸出找不到可行解的訊息。

第七章 數據實驗

本章節的主要目的，在於實驗驗證本研究所提出之搜尋演算法，其求解品質較文獻提出之方法更能得到品質較佳的維修基本週期和維修乘數。第7.1節以簡單範例說明本研究提出的搜尋演算法之求解過程，並和文獻提出的方法進行比較。第7.2節利用隨機實驗說明本研究提出之求解方法比文獻提出的求解方法更能得到品質較佳的解答。第7.3節為敏感度分析，說明操作成本函數的非線性對目標函數之影響程度。

7.1 範例說明

本節主要運用第五章所提出的搜尋演算法合併使用第六章可行解測試法，以一個簡單範例說明求解之過程與求解品質的好壞。

假設維修系統中共有 $n=5$ 台機器，主要整備成本 $M=50$ ，操作機器的變動成本中對時間的幕次數 $e=1$ 。在可行解測試法中，初始條件設定門檻值 $Y=3$ ，計次器 $\chi=0$ ，可行解 $\varphi=0$ 。其他運用到的數據如表7.1所示。

表 7.1 範例說明之數據

機器	1	2	3	4	5
m_i	88	192	193	205	204
f_i	23	8	21	69	13
v_i	35	18	5	60	4
X_i	0.5	0.8	0.7	1.02	0.4

7.1.1 本研究之搜尋演算法

將上述之數據代入本研究之方法，其詳細搜尋演算法之步驟說明如下所示：

步驟 1：計算搜尋上界值 $T_U = 7.1414$ 與搜尋下界值 $T_L = 0.1919$ 。目前所得之最佳解為 $T^* = 7.1414$ ， $K^* = \{1,1,1,1,1\}$ 與 $TC^* = \$592.7898$ 。因為 $T_U = 7.1414$ 為第 1 個接合點，所以 $p=1$ 。

步驟 2：找到離 $w_p = 7.1414$ 最近的接合點之機器下標 $\pi = 3$ 。計算時間點 $w_{p+1} = 6.2128$ 之維修乘數 $K(w_{p+1}) = (1,1,2,1,1)$ 。

步驟 3：不等式 $w_{p+1} < T_L$ 不成立。

步驟 4：經測試， $(K(w_{p+1}), w_{p+1})$ 可產生一組可行的維修排程。計算

$T = \max(\bar{T}(K(w_{p+1})), L) = 3.42$ 與 $TC(K(w_{p+1}), T) = \$480.1482$ 。因為不等式 $TC^* \geq TC(K(w_{p+1}), T)$ 成立，所以變更目前最佳值為

$T^* = 3.42, K^* = (1,1,2,1,1)$ 與 $TC^* = \$480.1482$ 及變更搜尋下界值為 $T_L = 1.4246$ 。令 $w_p = 6.2128$ 與 $K(w_p) = (1,1,2,1,1)$ 。

步驟 5：重複上述步驟 2、步驟 3 與步驟 4。直到 $(K(w_{p+1}), w_{p+1}) = (\{1,2,3,1,3\}, 2.9154)$ 產生一組不可行的維修排程。

步驟 6：接下來判斷 $(K(w_p), w_{p+1})$ 是否為可行維修排程，結果依然產生一組不可行的維修排程。

步驟 7：因為有維修乘數大於 1 之機器，所以找出成本增加幅度最小之機器下標 $\lambda = 3$ ，並計算時間點 $w_{p+1} = 2.9154$ 之維修乘數

$K(w_p) = (1,2,2,1,2)$ 。

步驟 8：經判斷， $(K(w_p), w_{p+1})$ 為可行維修排程。計算 $T = 2.6984$ 與

$TC(K(w_p), T) = \$452.4064$ 。因為不等式 $TC^* \geq TC(K(w_p), T)$ 不成立，所以不變更目前最佳值。令 $w_p = 2.9154$ 與 $K(w_p) = (1,2,2,1,2)$ 。

步驟 9：重複上述步驟 2 到步驟 8，直到 $K(w_p) = (1,1,1,1,1)$ ，即沒有維修乘數大於 1 之機器。

步驟 10：計算 $T = 3.6749$ 與 $TC(K(w_p), T) = \$495.8152$ 。因為不等式

$TC^* \geq TC(K(w_{p+1}), T)$ 不成立，所以不變更目前之最佳值。

步驟 11：輸出最佳值 $K^* = \{1,2,3,1,2\}, T^* = 3.02$ 與 $TC^* = \$451.0515$ 。

所以，經過上述搜尋的步驟之後，可得到本範例的最佳基本週期為 $T^* = 3.02$ ，最佳維修乘數 $K^* = \{1,2,3,1,2\}$ ，平均總成本為 $TC^* = \$451.05$ ，維修排程如表 7.2 所示，另外 $(1 - \frac{X_i}{k_i T})$ 分別為 $(0.834, 0.868, 0.923, 0.662, 0.934)$ 大

約趨近於 1，因此本研究確實可運用式(4.1)進行理論分析。

7.1.2 Goyal和Kusy與Yao之啟發式方法

採用 Goyal 和 Kusy(1985)與 Yao(2005)之啟發式方法計算相同數據，二

種方法皆得到最佳基本週期為 $T^* = 2.6160$ 與最佳維修乘數 $K^* = \{1, 2, 3, 1, 4\}$ 。將此組最佳解 (K^*, T^*) 代入本研究之目標函數(式(3.1))，可得到平均總成本為 $TC_G^* = \$439.59$ 。經可行解測試，此組解無法產生可行維修排程，如表 7.3 所示，另外 $(1 - \frac{X_i}{k_i T})$ 分別為 $(0.809, 0.847, 0.911, 0.61, 0.963)$ 大約趨近於 1。

表 7.2 範例說明之維修排程

基期	機器 1	機器 2	機器 3	機器 4	機器 5	總維修時間
1	0.5	0.8	0	1.02	0	2.32
2	0.5	0	0.7	1.02	0.4	2.62
3	0.5	0.8	0	1.02	0	2.32
4	0.5	0	0	1.02	0.4	1.92
5	0.5	0.8	0.7	1.02	0	3.02
6	0.5	0	0	1.02	0.4	1.92

表 7.3 範例說明之維修排程

基期	機器 1	機器 2	機器 3	機器 4	機器 5	總維修時間
1	0.5	0.8	0	1.02	0	2.32
2	0.5	0	0.7	1.02	0	2.22
3	0.5	0.8	0	1.02	0	2.32
4	0.5	0	0	1.02	0.4	1.92
5	0.5	0.8	0.7	1.02	0	3.02
6	0.5	0	0	1.02	0	1.52
7	0.5	0.8	0	1.02	0	2.32
8	0.5	0	0.7	1.02	0.4	2.62
9	0.5	0.8	0	1.02	0	2.32
10	0.5	0	0	1.02	0	1.52
11	0.5	0.8	0.7	1.02	0	3.02
12	0.5	0	0	1.02	0.4	1.92

由數據結果得知：雖然 Goyal 和 Kusy(1985)與 Yao(2005)提出的方法，平均總成本比本研究提出的方法低 2.54%，但是在 Goyal 和 Kusy(1985)與 Yao(2005)所產生的維修排程中，基期 5、基期 8 和基期 11 的總維修時間卻大於基本週期，不滿足維修產能限制，因此 Goyal 和 Kusy(1985)與 Yao(2005)提出的方法所產生的維修排程其實並不可行。所以在本範例中，本研究提出的方法之求解品質仍比 Goyal 和 Kusy(1985)與 Yao(2005)提出的方法佳。

7.2 隨機實驗

本節主要目的是利用隨機產生之數據來說明本研究提出之方法，求解品質比文獻提出之方法佳。首先，我們選取六種不同之機器數目 ($n = 3, 5, 7, 10, 25, 50$) 與主要整備成本 ($M = 50, 100, 200, 500, 750, 1000$)，一共產生 36 種 n 與 M 組合。針對每一種 n 與 M 組合，運用均勻分配函數產生 m_i, f_i, v_i, e ，如表 7.4 所示與維修時間 X_i ，參見附錄 C 所示，分別隨機產生 100 組數據，因此本研究之隨機實驗共產生 3600 組數據。

表 7.4 參數範圍

n	3,5,7,10,25,50
M	50,100,200,500,750,1000
m_i	[1,500]
f_i	[15,50]
v_i	[1,20]
e	[1,4]

將每一種 n 與 M 組合所產生的 100 組數據，分別代入文獻提出之方法與本研究提出之方法，並將文獻中所有產生可行維修排程的 $(\{k_i\}, T)$ 代入本研究之目標函數(式(3.1))，比較其平均總成本與本研究提出之方法的差異，實驗結果如表 7.5 所示。茲將 Goyal 和 Kusy(1985)提出之方法簡稱 G&K，本研究提出之方法以作者英文姓氏(Yao,Huang,Yang)簡稱為 YHY。

定義本研究與 Goyal 和 Kusy(1985)之平均總成本差異度為

$$I_{G\&K} = \frac{Cost_{G\&K} - Cost_{YHY}}{Cost_{YHY}} \times 100\% \quad (7.1)$$

另外，本研究與 Yao(2005)之平均總成本差異度為

$$I_{Yao} = \frac{Cost_{Yao} - Cost_{YHY}}{Cost_{YHY}} \times 100\% \quad (7.2)$$

由隨機實驗結果可得到下列結果：隨著機器數目 n 的增加，本研究與 Goyal 和 Kusy(1985)之平均總成本的差異度越大，但是在 n 從 25 增加到 50 時，改善幅度卻沒有明顯增加，有可能當 $n=50$ 時，所得到的維修乘數為 1，因此成本改善幅度並不大。另外，本研究與 Goyal 和 Kusy(1985)之平均總成本的差異度並未隨 M 的增加或減少而有明顯的差異。

Yao(2005)提出之方法，雖然和本研究之平均總成本的差異度不大，但是在每一種 n 與 M 組合下所產生的 100 組隨機試驗中，可行解的個數卻不超過二成，尤其在 $n=25$ 與 $n=50$ 時，完全得不到可行維修排程。

整體而言，本研究提出之方法之求解品質比 Goyal 和 Kusy(1985)與 Yao(2005)提出之方法佳，除了保證可以得到可行維修排程之外，還可以得到較低的平均總成本。本研究提出之搜尋演算法求得的最佳解，可作為決策者安排機器維修排程的參考依據。

7.3 敏感度分析

本小結希望藉由針對不同的 e 值(操作機器的變動成本中對時間的冪次數)，了解操作成本函數的非線性對於目標函數的影響程度。

選取機器數目 $n=5$ 、主要整備成本 ($M = 50, 100, 200, 500, 750, 1000$) 與 $e = 1, 2, 3, 4$ ，共產生 24 種 M 與 e 組合。針對每一種 M 與 e 組合，運用均勻分配函數產生 m_i, f_i, v_i ，如表 7.4 所示與維修時間 X_i ，參見附錄 C 所示，分別隨機產生 100 組數據，實驗結果如表 7.6 所示。

由實驗結果得到：本研究與 Goyal 和 Kusy(1985)之平均總成本的差異度，隨著冪次數 e 的增加，二者之間的差異度越明顯，也就是非線性對於平均總成本的差異度之影響程度越大。另外，本研究與 Yao(2005)之平均總成本的差異度，由於在 $M = 50, 100, 200$ 時，產生之可行解個數極少，因此不容易觀察到不同 e 值對目標函數的影響；在 $M = 500, 750$ 時，二者之平均總成本的差異度隨 e 值的增加而增加；在 $M = 1000$ 時， e 值對於平均總成本的差異度並沒有明顯的呈現遞增或是遞減的趨勢。整體而言，本研究與文獻之平均總成本的差異度是隨著 e 值的增加而呈線性增加。

表 7.5 在 100 組隨機試驗中可行解個數與本研究在可行解成本之改善

		本研究方法較 Goyal 和 Kusy 之改進($I_{G\&K}$)						本研究方法較 Yao 之改進 (I_{Yao})			
n	M	$\frac{\text{可行解個數}}{100\text{組}}$	Min (%)	Max (%)	Avg. (%)	n	M	$\frac{\text{可行解個數}}{100\text{組}}$	Min (%)	Max (%)	Avg. (%)
3	50	63	0.0013	0.81	0.17	3	50	0	-	-	-
	100	65	0.008	1.96	0.26		100	0	-	-	-
	200	81	0.012	2.00	0.26		200	3	0.16	2.08	1.23
	500	87	0.16	1.47	0.56		500	3	0.63	1.90	1.01
	750	97	0.03	2.12	0.38		750	14	0.07	1.01	0.94
	1000	98	0.02	1.21	0.54		1000	17	0.40	1.76	1.80
5	50	64	0.007	2.47	1.27	5	50	0	-	-	-
	100	79	0.002	3.38	1.36		100	1	1.93	1.93	1.93
	200	81	0.05	3.40	1.35		200	2	0.97	4.03	2.32
	500	87	0.002	3.23	1.11		500	6	0.31	1.10	0.71
	750	95	0.003	3.16	0.90		750	11	0.16	3.95	1.99
	1000	94	0.005	4.11	0.87		1000	19	0.35	3.70	1.56
7	50	70	0.15	4.52	2.60	7	50	0	-	-	-
	100	66	0.20	4.82	2.74		100	0	-	-	-
	200	78	0.09	4.77	2.38		200	0	-	-	-
	500	81	0.20	5.03	2.83		500	3	0.43	2.33	1.19
	750	91	0.08	5.23	2.19		750	7	0.05	1.62	0.59
	1000	96	0.05	5.18	2.47		1000	18	0.02	2.18	1.51

表 7.5 在 100 組隨機試驗中可行解個數與本研究在可行解成本之改善(續)

		本研究方法較 Goyal 和 Kusy 之改進($I_{G\&K}$)						本研究方法較 Yao 之改進 (I_{Yao})			
n	M	$\frac{\text{可行解個數}}{100\text{組}}$	Min (%)	Max (%)	Avg. (%)	n	M	$\frac{\text{可行解個數}}{100\text{組}}$	Min (%)	Max (%)	Avg. (%)
10	50	72	0.32	6.43	3.16	10	50	0	-	-	-
	100	71	0.006	6.78	3.49		100	0	-	-	-
	200	75	0.17	6.69	3.95		200	0	-	-	-
	500	88	0.004	6.85	3.73		500	8	0.13	0.38	0.24
	750	87	0.13	7.23	4.02		750	6	0.14	0.73	0.43
	1000	94	0.02	6.74	3.57		1000	15	0.40	1.18	0.72
25	50	72	0.92	6.31	4.05	25	50	0	-	-	-
	100	68	0.13	8.78	4.20		100	0	-	-	-
	200	68	0.26	8.12	3.93		200	0	-	-	-
	500	76	0.15	8.92	4.51		500	0	-	-	-
	750	78	0.17	9.70	4.48		750	0	-	-	-
	1000	84	0.023	9.79	4.89		1000	0	-	-	-
50	50	68	0.48	9.32	4.48	50	50	0	-	-	-
	100	77	0.36	9.56	4.65		100	0	-	-	-
	200	80	0.27	10.2	4.83		200	0	-	-	-
	500	83	0.023	9.67	4.15		500	0	-	-	-
	750	90	0.11	9.92	4.55		750	0	-	-	-
	1000	87	0.24	10.23	4.82		1000	0	-	-	-

表 7.6 敏感度分析

		本研究方法較 Goyal 和 Kusy 之改進($I_{G\&K}$)						本研究方法較 Yao 之改 進(I_{Yao})			
n	M	e	Min. (%)	Max. (%)	Avg. (%)	n	M	e	Min. (%)	Max. (%)	Avg. (%)
5	50	1	0.005	0.16	0.08	5	50	1	0.59	0.59	0.59
		2	0.09	1.08	0.65			2	-	-	-
		3	0.13	2.64	1.57			3	-	-	-
		4	0.29	3.74	2.12			4	-	-	-
	100	1	0.04	0.33	0.14		100	1	-	-	-
		2	0.07	1.05	0.62			2	3.36	3.36	3.36
		3	0.09	2.59	1.55			3	1.73	1.73	1.73
		4	0.12	3.98	1.93			4	-	-	-
	200	1	0.02	0.47	0.15		200	1	0.35	2.84	1.14
		2	0.09	1.31	0.63			2	4.73	6.03	5.01
		3	0.21	2.92	1.58			3	-	-	-
		4	0.07	4.36	1.86			4	-	-	-
	500	1	0.01	0.88	0.12		500	1	0.12	0.12	0.12
		2	0.01	1.61	0.58			2	0.07	3.73	1.37
		3	0.01	2.94	1.38			3	0.36	7.74	2.02
		4	0.05	3.91	1.59			4	4.19	4.19	4.19
	750	1	0.008	0.35	0.10		750	1	0.06	5.78	1.94
		2	0.03	1.44	0.63			2	0.08	1.57	4.27
		3	0.01	2.74	1.24			3	0.33	7.51	4.42
		4	0.06	2.87	1.22			4	1.89	6.87	4.25
1000	1	0.01	0.84	0.10	1000	1	0.02	6.08	2.45		
	2	0.004	1.33	0.51		2	0.46	8.87	3.83		
	3	0.02	2.76	1.08		3	0.04	5.11	2.99		
	4	0.005	3.32	1.34		4	2.46	2.46	2.46		

第八章 結論與未來研究方向

8.1 結論

為探討考慮維修產能限制下之群組機器維修排程問題，本研究除了推導其數學模式進行完整理論分析外，並以理論結果提出一套有效的搜尋演算法。另外，運用一個啟發式解法產生一組可行維修排程。經數據實驗可知，本研究提出的求解方法，除了比文獻中的解法有更好的求解品質，並可保證產生可行的維修排程。因此，本研究可以提供管理機器設備之經理人作決策之依據。

8.2 未來研究方向

在本研究所探討的維修相關操作，目前以單一維修設備的情境作為探討的基礎，雖然符合部分的實務情境，但是在實務操作環境中，另外常見如流程式(flowshop type)的維修操作方式，其維修工作必須經過多站式的維修操作。在此情境下的維修排程勢必更為複雜，將在未來繼續作為研究方法的研究延伸。

參考文獻

1. 姚銘忠，最佳化專題上課講義，東海大學工業工程與經營資訊研究所，台中，2005。
2. 張智星，MATLAB 程式設計與應用，清蔚科技，新竹市，2000。
3. 陳英欽，「以遺傳演算法求解一般整數策略下之經濟批量排程問題」，碩士論文，東海大學工業工程與經營資訊研究所，2002。
4. Arkin E., Joneja, D., Roundy, R., “Computational Complexity of Uncapacitated Multi-Echelon Production Planning Problems,” *Operations Research Letters*, 8, 61–6, (1989).
5. Anily, S, C. A. Glass, and R. Hassin, “The Scheduling of Maintenance Service”, *Discrete Applied Mathematics*, 82, 27-42 (1998).
6. Anily, S., C. A. Glass and R. Hassin, “Scheduling of Maintenance Services to Three Machines”, *Annals of Operations Research*, 86, 375-391 (1999).
7. Amotz, B.N., R. Bhatia, Naor, J. and B. Schieber, “Minimizing Service and Operation Costs of Periodic Scheduling”, *Mathematics of Operations Research*, 27, 518-544 (2002).
8. Christer, A. H., T. Doherty, “Scheduling Overhauls of Soaking Pits”, *Operational Research Quarterly*, 28(4), 915-926 (1977).
9. Chan, M. Y. and F. Chin, “General Schedulers for the Pinwheel Problem based on double-integer reduction”, *IEEE Transactions on Computers*, 41(6), 755-768 (1992).
10. Chan, M. Y. and F. Chin, “Schedulers for Larger Classes of Pinwheel Instances”, *Algorithmica*, 9,425-462 (1993).
11. Cho, D. I. and Parlar, M., “A Survey of Maintenance Models for Multi-unit Systems”, *European Journal of Operational Research*, 51, 1-23 (1991).
12. Dekker, R., R. Wildeman and F. van der Duyn Schouten, “A Review of Multi-component Maintenance Models with Economic Dependence”, *Mathematical Methods of Operations Research*, 45, 411-435 (1997).
13. Goyal, S.K., “Economic Packaging Frequency for Items Jointly Replenished,” *Operations Research*, 21, 644–647 (1973).
14. Goyal, S. K. and A. Gunasekaran, “Determining Economic Maintenance Frequency of a Transportation Fleet”, *International Journal of Systems Science*, 23(4), 655-659 (1992).
15. Goyal, S. K. and M. I. Kusy, “Determining Economic Maintenance Frequency for a Family of Machines”, *Journal of the Operational Research Society*, 36(12), 1125-1128 (1985).
16. Glass, C. A., “Feasibility of Scheduling Lot Sizes of Three Products on One Machine”, *Management Science*, 38, 1482-1494 (1992).
17. Glass, C. A., “Feasibility of Scheduling Lot Sizes of Two Frequencies on One Machine”, *European Journal of Operational Research*, 75, 354-364 (1994).

18. Hassin, R. and N. Megiddo, "Exact Computation of optimal inventory policy over an unbounded horizon", *Mathematics of Operations Research*, 16, 534-546 (1991).
19. Holte, R., Z. Rosier, I. Tulchinsky and D. Varvel, "Pinwheel Scheduling with two distinct numbers", *Theoretical Computer Science*, 100, 105-135 (1992).
20. Huang, J.Y., "New Search Algorithm for Solving the Maintenance Scheduling Problem for a Family of Machines," *Optimization Methods and Software*, v 21, n 3, Jun 1, 2006, p 461-477
21. Krämer, F. J. and S. X.. Bai, "Optimal Control of a Production System with Periodic Maintenance", *Optimal Control Applications and Methods*, 17, 281-307 (1996).
22. Luss, H. and Z. Kander, "Preparedness Model Dealing With N Systems Operating Simultaneously", *Operations Research*, 22(1), 117-128 (1974).
23. Luss, H., "Maintenance Policies When Deterioration can Be Observed by Inspections", *Operations Research*, 24(2), 359-366 (1976).
24. Lee, F.C., Yao, M.J., "A Global Optimum Search Algorithm for the Joint Replenishment Problem under Power-of-Two Policy," *Computers and Operations Research*, 30, 1319-1333 (2003).
25. Nocturne, D.J., "Economic Ordering Frequency for Several Items Jointly Replenished," *Management Science*, 19(9), 1073-99 (1973).
26. Shu, F.T., "Economic Ordering Frequency for Two Items Jointly Replenished," *Management Science*, 17(6), B406-410 (1971).
27. Sule, D. R. and B. Harmon, "Determination of Coordinated Maintenance Scheduling Frequencies for a Group of Machines", *AIIE Transactions*, 11(1),48-53 (1979).
28. Su, C, and L. Tassiulas, "Broadcast Scheduling for Information Distribution", *Proceeding of IEEE Infocom (Infocom 97)*, Kobe, Japan, 109-117 (1997).
29. Yao, M.J., "On the Feasibility Testing of the Economic Lot Scheduling Problem Using the Extended Basic Period Approach," *Journal of the Chinese Institute of Industrial Engineers*, 18(3), 13-20 (2002).
30. Yao, M.J., "On Determining the Optimal Maintenance Frequency for a Family of Machines," *Journal of the Chinese Institute of Industrial Engineers*, 22(3), 199-209, (2005).
31. Van Eijs, M. J. G., "A Note on the Joint Replenishment Problem under Constant Demand", *Journal of the Operational Research Society*, 44, 185-191 (1993).
32. Viswanathan, S., "A New Optimal Algorithm for the Joint Replenishment Problem", *Journal of the Operational Research Society*, 47, 936-944 (1996).

附錄A 數學證明

證明 $\Phi_i(k_i^*(\bar{T})-1, \delta_i(k_i^*(\bar{T})-1)) > \Phi_i(k_i^*(\bar{T}), \delta_i(k_i^*(\bar{T})))$

[Proof] :

當 $e=1$ 時，左式-右式

$$= \sqrt{(m_i - f_i X_i) v_i k_i} \left(\frac{\sqrt{k_i^2} - \sqrt{k_i^2 - 1}}{k_i \sqrt{k_i - 1}} \right) + \sqrt{\frac{(m_i - f_i X_i)}{2k_i} \left(\frac{\sqrt{(k_i + 1)^3} - \sqrt{k_i^2(k_i - 1)}}{\sqrt{k_i^2 - 1}} \right)} > 0$$

，原式成立。

設 $e=n$ 時，原式成立，即

$$\text{左式-右式} = \frac{m_i - f_i X_i}{\left[\frac{m_i(n+1)}{k_i v_i} \right]^{\frac{1}{n+1}}} \cdot A + \frac{v_i \left[\frac{(m_i - f_i X_i)(n+1)}{k_i v_i} \right]^{\frac{n}{n+1}}}{n+1} \cdot B > 0$$

$$\text{其中 } A = \frac{1}{(k_i - 1) \left\{ \frac{1}{(k_i - 1)[k_i^n - (k_i - 1)^n]} \right\}^{\frac{1}{n+1}}} - \frac{1}{k_i \left\{ \frac{1}{(k_i + 1)[(k_i + 1)^n - k_i^n]} \right\}^{\frac{1}{n+1}}}$$

$$B = (k_i + 1)^n \left[\frac{1}{(k_i - 1)[k_i^n - (k_i - 1)^n]} \right]^{\frac{n}{n+1}} - k_i^n \left[\frac{1}{(k_i + 1)[(k_i + 1)^n - k_i^n]} \right]^{\frac{n}{n+1}}$$

則 $e=n+1$ 時，

$$\text{左式-右式} = \frac{m_i - f_i X_i}{\left[\frac{(m_i - f_i X_i)(n+2)}{k_i v_i} \right]^{\frac{1}{n+2}}} \cdot A + \frac{v_i \left[\frac{(m_i - f_i X_i)(n+2)}{k_i v_i} \right]^{\frac{n+1}{n+2}}}{n+1} \cdot B > 0$$

$$\text{其中 } A = \frac{1}{(k_i - 1) \left\{ \frac{1}{(k_i - 1)[k_i^{n+1} - (k_i - 1)^{n+1}]}\right\}^{\frac{1}{n+2}}} - \frac{1}{k_i \left\{ \frac{1}{(k_i + 1)[(k_i + 1)^{n+1} - k_i^{n+1}]} \right\}^{\frac{1}{n+2}}}$$

$$B = (k_i + 1)^{n+1} \left[\frac{1}{(k_i - 1)[k_i^{n+1} - (k_i - 1)^{n+1}]} \right]^{\frac{n+1}{n+2}} - k_i^{n+1} \left[\frac{1}{(k_i + 1)[(k_i + 1)^{n+1} - k_i^{n+1}]} \right]^{\frac{n+1}{n+2}}$$

原式亦成立

故由數學歸納法得証，原式對於任意自然數均成立。■

附錄B-1 Proc. IS之流程圖

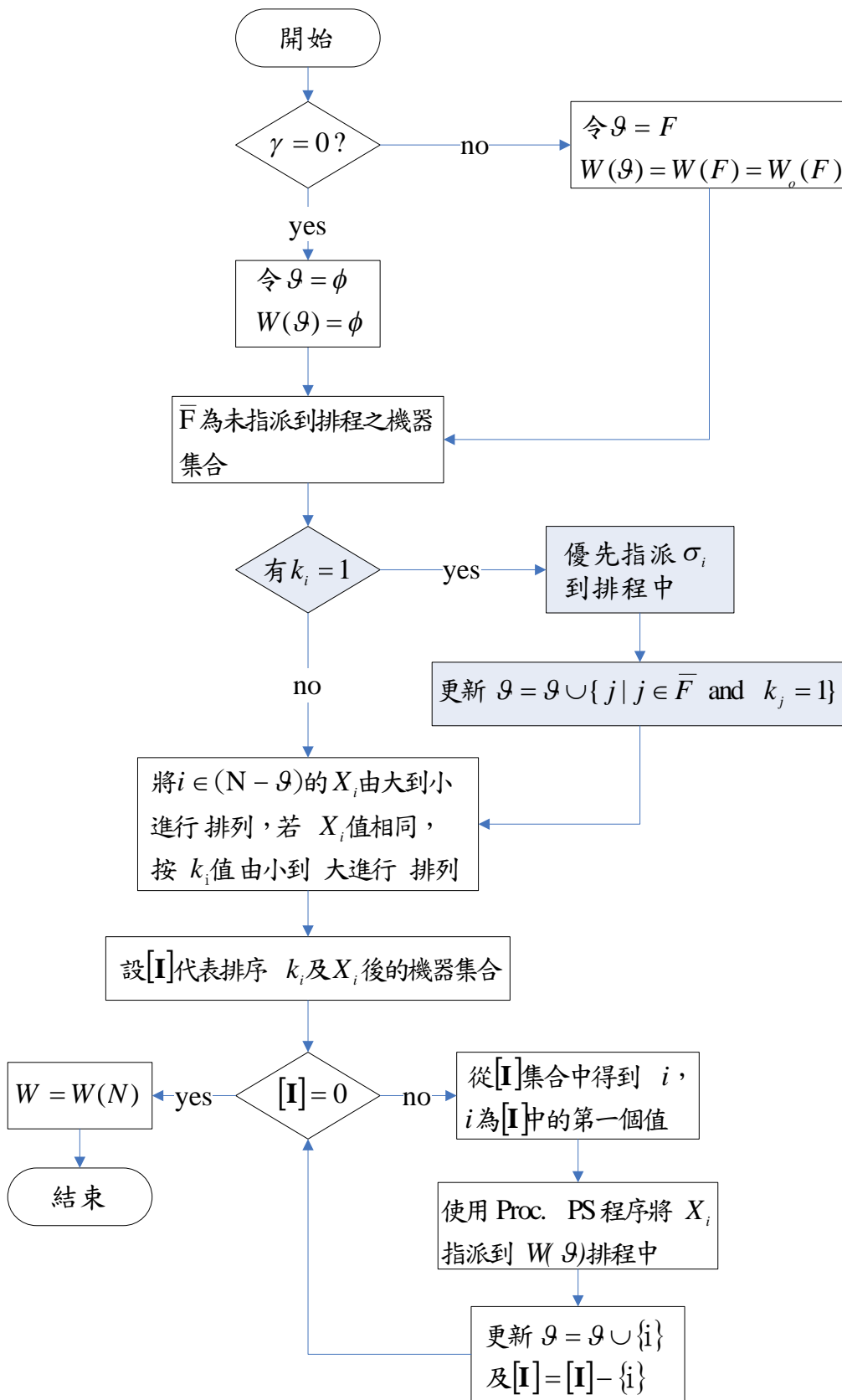


圖 B.1 Proc. IS 之流程圖

附錄B-2 Proc. PS之流程圖

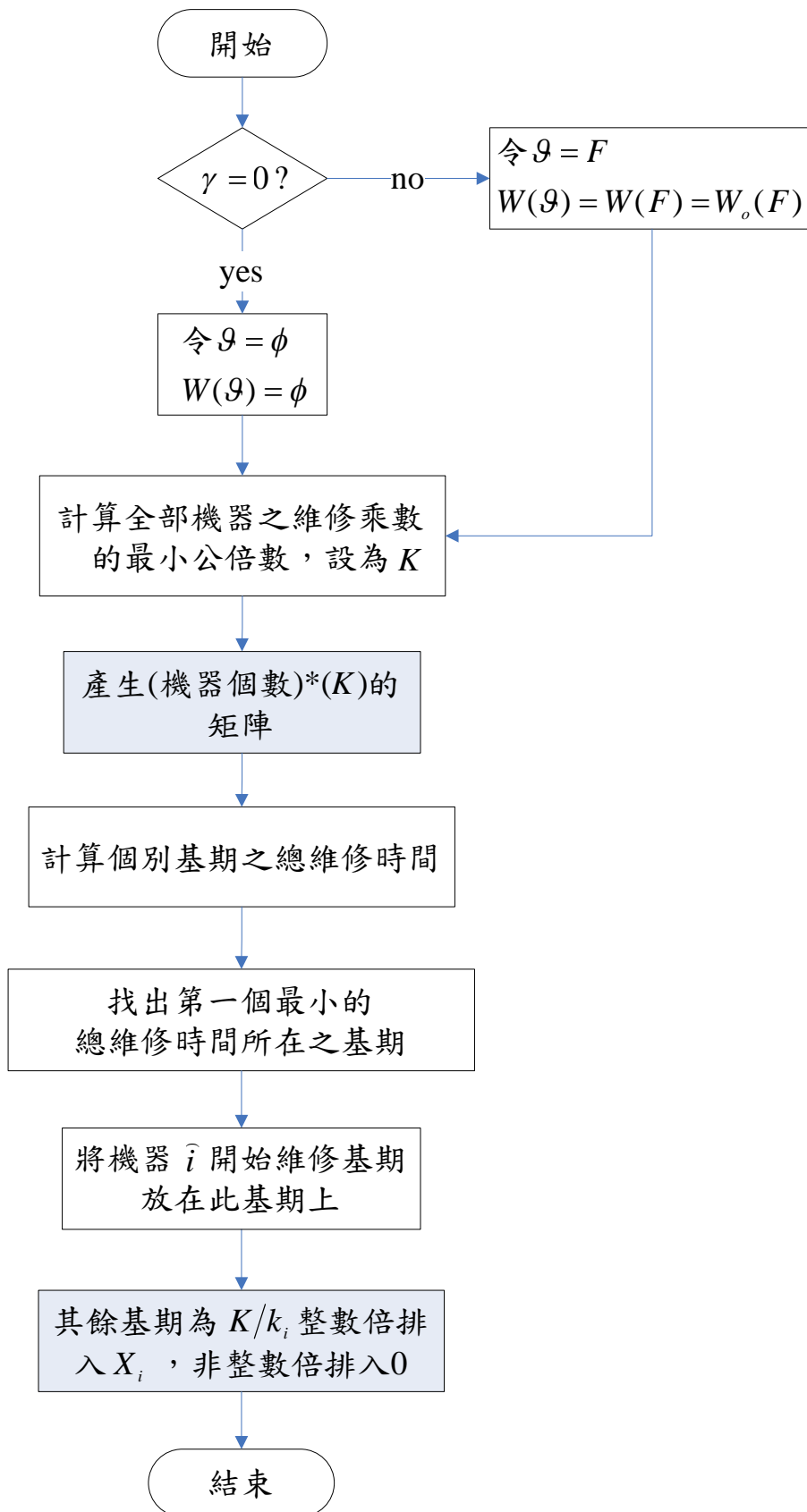


圖 B.2 Proc. PS 之流程圖

附錄B-3 移出之流程圖

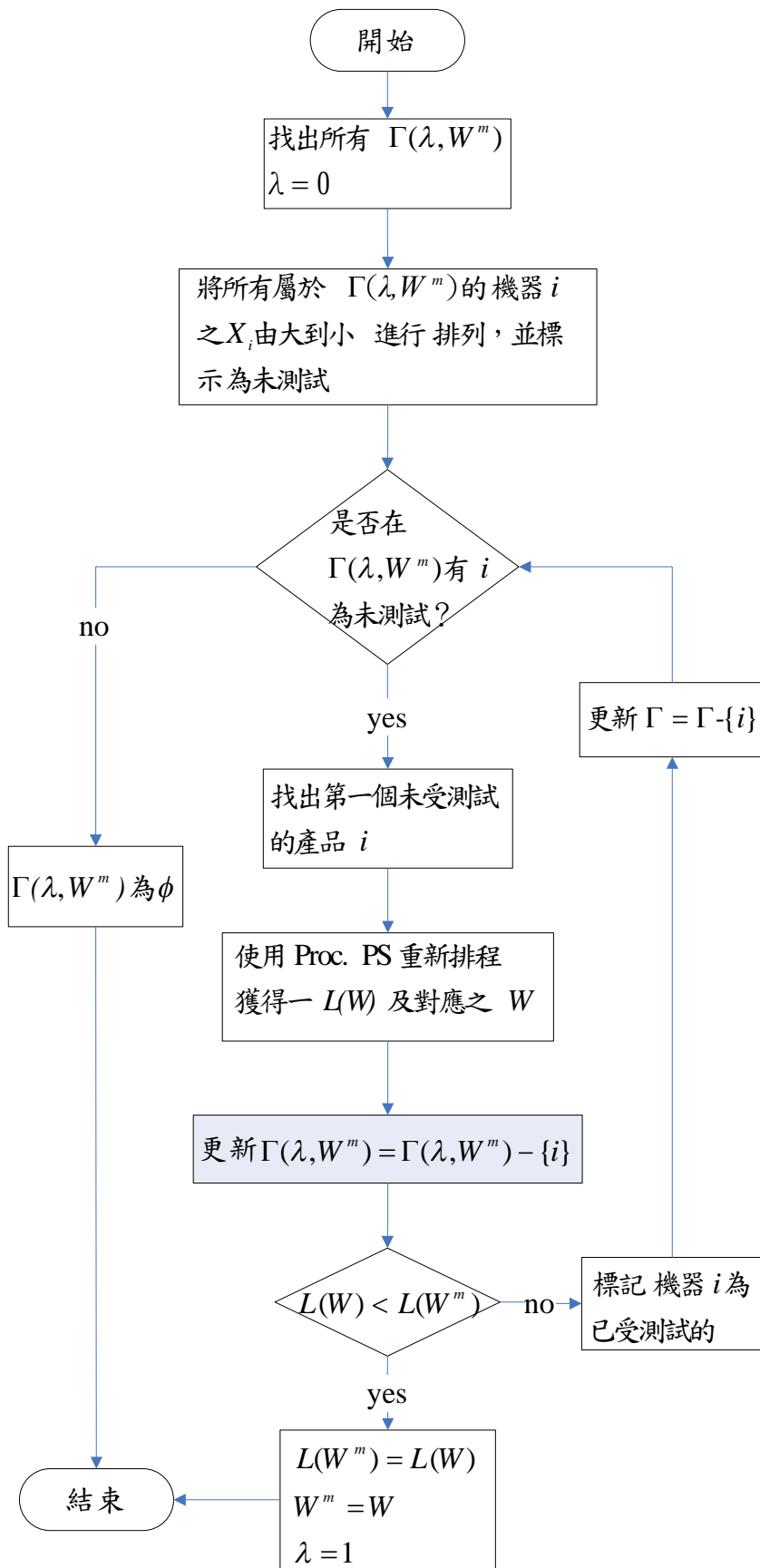


圖 B.3 移出之流程圖

附錄B-4 1對1互換之流程圖

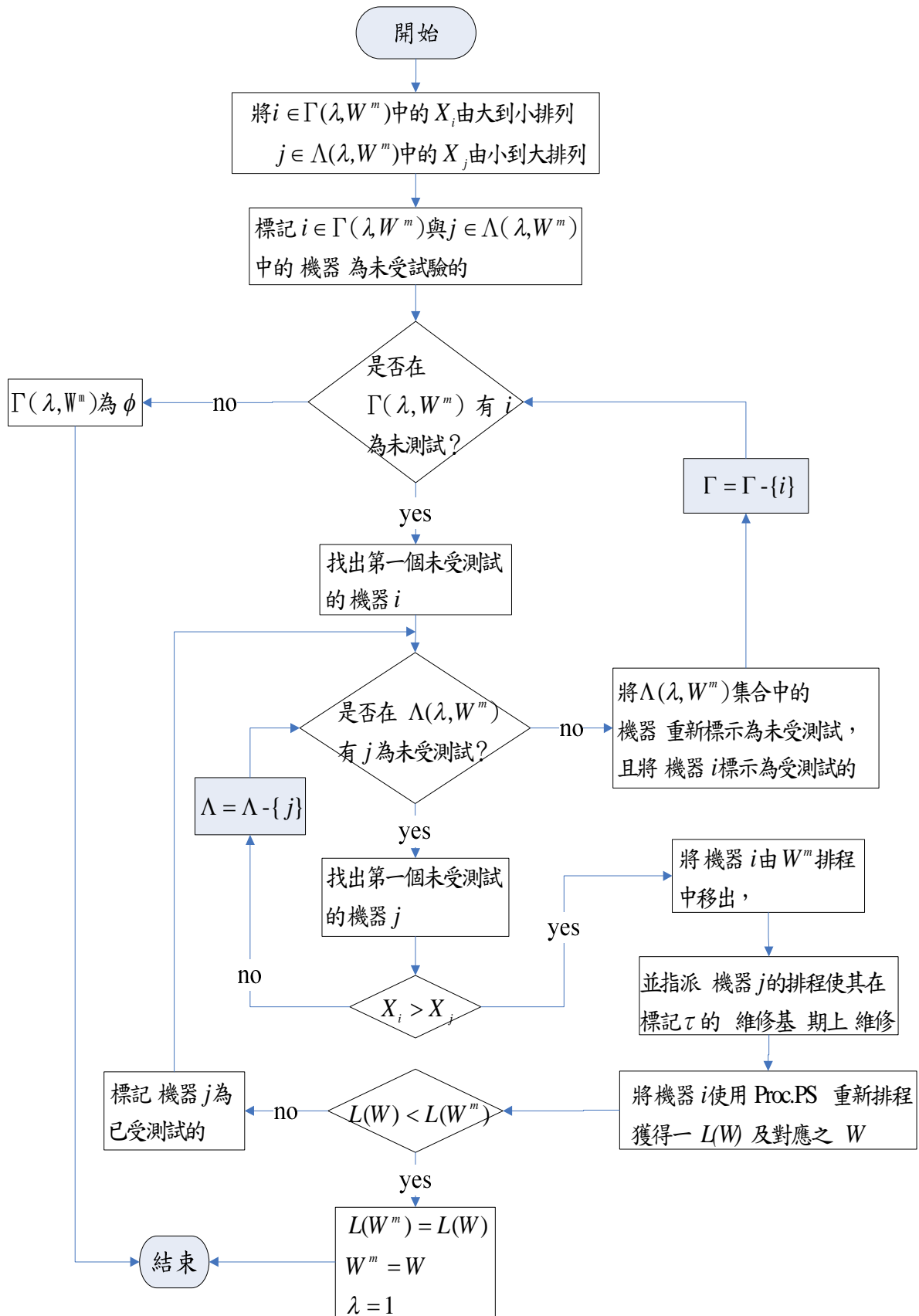


圖 B.4 1對1互換之流程圖

附錄B-5 2對1互換之流程圖

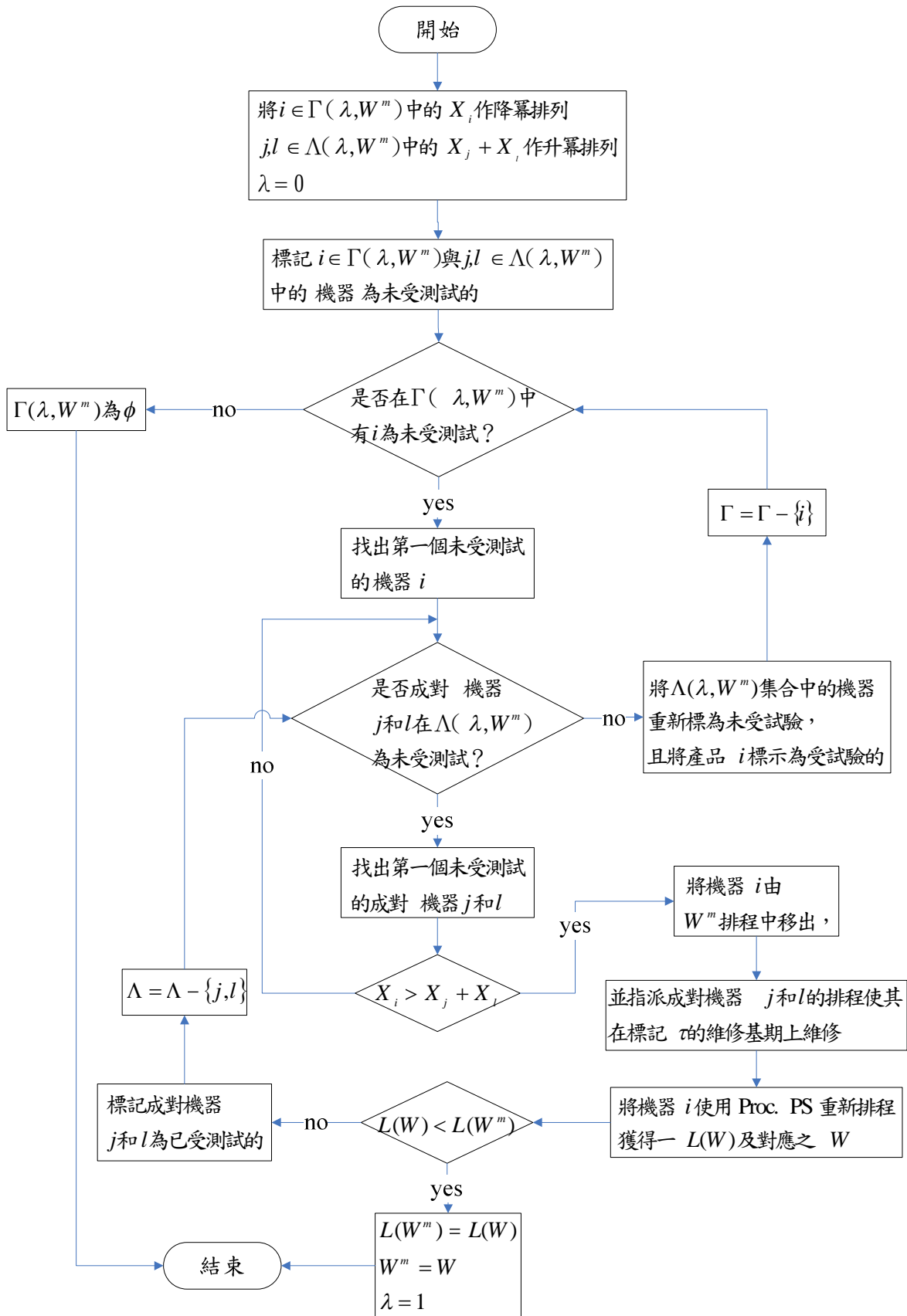


圖 B.5 2對1互換之流程圖

附錄C 維修時間範圍之決定

在進行第 7.2 節隨機實驗之前，我們必須先對維修時間 X_i 之設定作說明。假如維修時間設定的太大時，可能造成本研究的最佳維修乘數皆為 1，也就是每台機器享有相同的維修週期，以某些程度而言這並非是品質佳的答案，除此之外也有可能造成文獻提出之方法求得的最佳解無法產生可行維修排程，如此一來則無法說明本研究提出之求解方法所得的答案之品質較佳。相反地，假如維修時間設定的太小時，文獻提出之方法求得的最佳解所產生的維修排程皆為可行排程，同樣無法說明本研究提出之求解方法所得的答案之品質較佳。有鑒於此，我們將提出一套啟發式方法產生一組維修時間。

我們運用經濟訂購批量(Economic Order Quantity, EOQ)得到最佳訂購量的作法產生本研究問題之維修時間：令 $T_i = k_i T$ ，式(3.1)將被改寫成式(C.1)，如下所示：

$$\begin{aligned} Z(k_1, \dots, k_n, T) &\approx \frac{M}{T} + \sum_{i=1}^n \left(\frac{m_i - f_i X_i}{k_i T} + \frac{v_i}{e+1} (k_i T)^e + f_i \right) \\ &= \frac{M}{T} + \sum_{i=1}^n \left(\frac{m_i - f_i X_i}{T_i} + \frac{v_i}{e+1} (T_i)^e + f_i \right) \\ &= \frac{M}{T} + \sum_{i=1}^n Z_i(T_i) \end{aligned} \quad (C.1)$$

其中

$$Z_i(T_i) = \frac{m_i - f_i X_i}{T_i} + \frac{v_i}{e+1} (T_i)^e + f_i \quad (C.2)$$

將式(C.2)對 T_i 進行一階導數並令其為零，可得到

$$T_i = \frac{\left(\frac{m_i - f_i X_i}{e} \right)^{\frac{1}{e+1}}}{\left(\frac{v_i}{e+1} \right)} \leq \frac{\left(\frac{m_i}{e} \right)^{\frac{1}{e+1}}}{\left(\frac{v_i}{e+1} \right)} = T_i^* \quad (C.3)$$

所以假設維修時間為

$$X_i = T_i^* \cdot \text{scaling_factor} \quad (C.4)$$

其中 *scaling_factor* 為調整因子，也就是維修時間佔維修週期的百分比。

假設我們希望經由 Goyal 和 Kusy(1985)之方法求得的最佳解其可行維修排程至少佔七成左右，則由數據分析的結果，參見表 C.2 所示，我們會

依照表 C.1 所示，當作是設定維修時間的基準。因此，接下來我們會根據表 C.1 所示，產生維修時間以進行第 7.2 節的隨機實驗。

表 C.1 調整因子之選擇基準

機器數 n	$scaling_factor$
3	0.4
5	0.225
7	0.1625
10	0.1125
25	0.046
50	0.0225

表 C.2 在 100 組隨機問題中可行解個數

$scaling_factor=0.5$						
$M \backslash n$	50	100	200	500	750	1000
3	0	3	7	18	48	68
5	0	0	0	0	0	0
7	0	0	0	0	0	0
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor=0.4</i>						
$M \backslash n$	50	100	200	500	750	1000
3	67	70	78	92	95	98
5	0	0	0	0	0	0
7	0	0	0	0	0	0
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor=0.3</i>						
$M \backslash n$	50	100	200	500	750	1000
3	94	90	92	100	99	100
5	0	0	1	5	21	26
7	0	0	0	0	0	0
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor=0.25</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	47	42	57	75	80	84
7	0	0	0	0	0	0
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor=0.225</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	99	98	100	100	100	100
5	73	77	84	86	97	97
7	0	0	0	0	0	0
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor</i> =0.2						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	99	99	100	100	100	100
5	81	84	90	94	98	100
7	0	1	1	13	19	41
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor</i> =0.175						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	97	93	100	98	99	100
7	48	44	63	79	76	89
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor=0.1625</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	95	96	98	99	100	100
7	73	73	79	86	91	96
10	0	0	0	0	0	0
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor=0.15</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	98	98	99	100	100	100
7	85	84	91	94	98	95
10	0	0	0	0	0	2
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor</i> =0.125						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	37	38	38	51	66	78
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor</i> =0.1125						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	66	69	67	78	86	96
25	0	0	0	0	0	0
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor=0.1</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	92	92	91	97	97	100
25	0	0	0	0	0	0
50	0	0	0	0	0	0
<i>scaling_factor=0.05</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	34	24	30	37	46	52
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor</i> =0.046						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	72	71	78	80	77	85
50	0	0	0	0	0	0
<i>scaling_factor</i> =0.04375						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	83	84	82	87	88	92
50	0	0	0	0	0	0

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor=0.025</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	100	100	100	100	100	100
50	32	27	28	38	39	38
<i>scaling_factor=0.0225</i>						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	100	100	100	100	100	100
50	76	80	69	78	79	86

表 C.2 在 100 組隨機問題中可行解個數(續)

<i>scaling_factor</i> =0.02						
<i>n</i> \ <i>M</i>	50	100	200	500	750	1000
3	100	100	100	100	100	100
5	100	100	100	100	100	100
7	100	100	100	100	100	100
10	100	100	100	100	100	100
25	100	100	100	100	100	100
50	100	98	99	99	99	99