

國立交通大學
工業工程與管理學系

碩士論文

以塔布搜尋法求解流線型製造單元排程

**A Tabu Search Approach to Scheduling
Flowshop Manufacturing Cell**



研究生：林耿漢

指導教授：巫木誠 博士

中華民國一百年六月

以塔布搜尋法求解流線型製造單元排程
**A Tabu Search Approach to Scheduling
Flowshop Manufacturing Cell**

研究生：林耿漢

Student：Keng-Han Lin

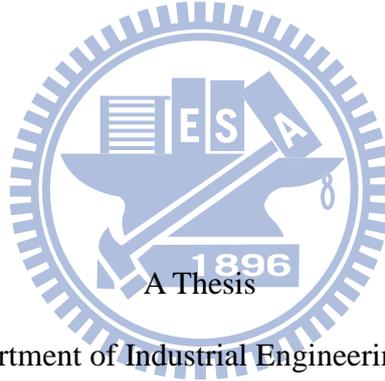
指導教授：巫木誠 博士

Advisor：Dr. Muh-Cherng Wu

國立交通大學

工業工程與管理學系

碩士論文



Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering

June 2011

Hsin-chu, Taiwan, Republic of China

中華民國一百年六月

A Tabu Search Approach to Scheduling Flowshop Manufacturing Cell

研究生：林耿漢

指導教授：巫木誠 博士

國立交通大學工業工程與管理研究所

中文摘要

巨集啟發式演算法-塔布搜尋法已經廣泛地應用於求解複雜的空間搜尋問題。過去的研究專注於如何應用或改善塔布搜尋法機制於不同的問題。相對於過去的研究，本研究則專注於新的研究方向-採取新的解表達方式是否可以改善塔布搜尋法的績效。本研究以固定序列流線型製造單元為排程背景，比較兩種不同的塔布搜尋演算法。值得注意的是，此兩種不同的塔布搜尋演算法在搜尋流程上是相同的，不同的地方在於使用了不同的解表達法，分別稱作舊表達法(S_{old})和新表達法(S_{new})。 S_{old} 為過去研究所使用的解表達法， S_{new} 為 Wu *et al.* (2011)使用的解表達法。使用 S_{old} 的演算法稱作 $Tabu-S_{old}$ ，使用 S_{new} 的演算法則稱作 $Tabu-S_{new}$ 。經過大量的數據實驗顯示，使用 S_{new} 的演算法 $Tabu-S_{new}$ 其實驗結果皆優於使用 S_{old} 的演算法 $Tabu-S_{old}$ 。這個發現顯示一個重要的新研究方向，即利用巨集啟發式演算法於不同空間求解問題時，亦可利用新的解表達法來改善空間求解問題。

關鍵詞：塔布搜尋，解表達法，排程。

A Tabu Search Approach to Scheduling Flowshop Manufacturing Cell

Student : Keng-Han Lin

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

The tabu-search mechanism, a type of meta-heuristic algorithms, has been widely used in solving complex space-search problems. Most prior research focused on how to apply or enhance the tabu-search mechanism to various problems. Aside from the traditional track, this research examines a new research issue—Can the adoption of a new solution representation scheme improve the performance of the tabu-search mechanism? A scheduling problem called flowshop manufacturing cell with permutation is used as the problem context, and two tabu-search algorithms are compared. The two algorithms, essentially the same in algorithmic flow, are distinct in using two different solution representation schemes (respectively called S_{old} and S_{new}). Noticeably, S_{old} was developed by prior studies and S_{new} is by Wu *et al.* (2011); the two algorithms are named *Tabu- S_{old}* and *Tabu- S_{new}* accordingly. Extensive numerical experiments reveal that *Tabu- S_{new}* comprehensively outperforms *Tabu- S_{old}* . This finding highlights an important new research track—exploring new solution representation schemes while applying meta-heuristic algorithms to various space-search problems.

Keywords: Tabu search, solution representation, scheduling

誌謝

本論文得以順利完成，首要感謝的是我的指導教授巫木誠教授的細心指導與教誨。每次與老師討論論文研究方向與問題時，老師除了在學術上細心指導外，也時常分享自己做事的經驗與做人的原則，並強調誠信的重要。而老師嚴謹與紀律的做事風格，使我在學術研究上獲益良多外，也學習到未來工作的做事態度與團隊合作精神，讓我在碩士班的兩年生活得到許多寶貴的經驗與做事態度。同時也感謝許錫美教授和彭德保教授在論文口試時，所給予的寶貴意見與指導，讓本論文更加完備。

研究所的兩年生活中，要感謝同實驗室的李奕勳、曾偉杰、潘冠銘與陳威宇，陪我一起修課、一起撰寫論文的的日子，並給我許多寶貴的意見與幫助，使我在交大碩士的學習過程中非常順利。還有同實驗的朱冠衡、連德凱、侯博鈞、洪正哲一起在實驗室奮鬥完成論文。以及碩一的學弟妹與博班的學長姐，規劃實驗室的聚餐與出遊，使我在交大的生活變的愉快而充實。在此並特別感謝陳振富學長的幫助，指導我程式的撰寫，並提供意見解決在研究中遇到的困難，使我得以順利完成研究。

我也要特別感謝我最親愛的家人的支持，特別是我的父母親林水鏡先生與張麗惠女士，感謝您們多年來的辛勞與關懷，讓我能無憂無慮的完成學業。也特別感謝我的女友胡婷琪，一直陪在我身邊幫我分擔論文的困難與煩惱，讓我能夠突破瓶頸完成論文。最後在此將此論文獻給我最敬愛的家人、師長、朋友與女友。

林耿漢 于 新竹交大

2011'7'1

目錄

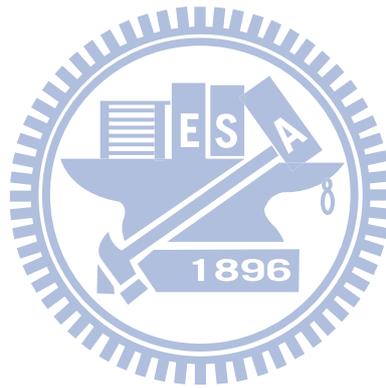
| | |
|-----------------------------|------|
| 中文摘要..... | ii |
| Abstract..... | iii |
| 誌謝..... | iv |
| 表目錄..... | vii |
| 圖目錄..... | viii |
| 第一章 緒論..... | 1 |
| 1.1 研究背景..... | 1 |
| 1.2 研究目的..... | 2 |
| 1.3 研究問題..... | 2 |
| 1.4 論文架構介紹..... | 3 |
| 第二章 文獻探討..... | 4 |
| 2.1 製造單元系統..... | 4 |
| 2.2 流線型生產分類..... | 5 |
| 2.3 家族整備時間分類..... | 7 |
| 2.4 解的表達法分類..... | 8 |
| 2.5 塔布搜尋法(Tabu Search)..... | 9 |
| 2.5.1 塔布搜尋法簡介..... | 9 |
| 2.5.2 塔布搜尋法應用於生產排程..... | 12 |
| 第三章 研究方法..... | 14 |
| 3.1 研究問題敘述..... | 14 |
| 3.2 研究限制與模型假設..... | 15 |
| 3.3 表達法設計與解讀..... | 16 |
| 3.4 舊表達法設計與解讀..... | 16 |
| 3.4.1 舊表達法設計..... | 16 |
| 3.4.2 舊表達法解讀..... | 17 |
| 3.5 新表達法設計與解讀..... | 17 |
| 3.5.1 新表達法設計..... | 17 |
| 3.5.2 新表達法解讀..... | 18 |
| 3.6 計算目標值..... | 19 |
| 3.7 塔布搜尋法流程..... | 19 |
| 3.7.1 塔布搜尋法基本元素應用..... | 20 |
| 3.7.2 塔布搜尋法求解步驟..... | 25 |
| 第四章 實驗情境與結果..... | 28 |
| 4.1 實驗目的..... | 28 |
| 4.2 實驗情境..... | 28 |
| 4.3 前測實驗..... | 29 |

| | |
|-----------------------------|----|
| 4.3.1 初始解設定..... | 29 |
| 4.3.2 終止條件代數設定..... | 32 |
| 4.4 實驗數據..... | 33 |
| 4.4.1 實驗總平均..... | 35 |
| 4.4.2 目標值平手與贏的次數比較..... | 35 |
| 4.4.3 目標值改善率..... | 37 |
| 4.5 統計檢定..... | 37 |
| 第五章 結論與未來研究方向..... | 40 |
| 5.1 結論..... | 40 |
| 5.2 未來研究方向..... | 41 |
| 參考文獻..... | 42 |
| 附錄一、各個情境下 ds_i 值計算表格..... | A |



表目錄

| | |
|--------------------------------|----|
| 表 2.1 製造單元系統文獻..... | 5 |
| 表 2.2 固定序列流線型分類..... | 7 |
| 表 2.3 家族整備時間..... | 8 |
| 表 2.4 解表達法..... | 9 |
| 表 2.5 塔布搜尋法..... | 13 |
| 表 3.1 塔布列表大小..... | 24 |
| 表 4.1 初始解比較..... | 31 |
| 表 4.2 初始解比較-單一情境(LSU1010)..... | 31 |
| 表 4.3 不同結束代數-目標值..... | 32 |
| 表 4.4 實驗數據結果..... | 34 |
| 表 4.5 實驗數據總平均..... | 35 |
| 表 4.6 統計檢定結果..... | 39 |



圖目錄

| | |
|-----------------------------|----|
| 圖 1.1 流線型生產..... | 3 |
| 圖 2.1 固定序列..... | 6 |
| 圖 2.2 非固定序列..... | 6 |
| 圖 2.3 塔布列表運作方式..... | 11 |
| 圖 3.1 求解製造單元系統過程..... | 14 |
| 圖 3.2 計算流程..... | 16 |
| 圖 3.3 舊表達法..... | 17 |
| 圖 3.4 舊表達法解讀..... | 17 |
| 圖 3.5 新表達法..... | 18 |
| 圖 3.6 新表達法工件族解讀..... | 18 |
| 圖 3.7 新表達法工件族內工件的解讀..... | 19 |
| 圖 3.8 舊表達法初始解-工件族順序..... | 20 |
| 圖 3.9 舊表達法初始解-工件族內工件順序..... | 21 |
| 圖 3.10 舊表達法的初始解..... | 21 |
| 圖 3.11 新表達法鄰近解..... | 22 |
| 圖 3.12 舊表達法鄰近解..... | 23 |
| 圖 3.13 塔布演算法流程..... | 27 |
| 圖 4.1 SSU 下的次數比較..... | 36 |
| 圖 4.2 MSU 下的次數比較..... | 36 |
| 圖 4.3 LSU 下的次數比較..... | 36 |
| 圖 4.4 目標值改善率..... | 37 |

第一章 緒論

1.1 研究背景

工廠能否順利在客戶所要求的時間下，完成產品達交，好的生產排程規劃占了很大的因素。好的生產規劃能夠提高工廠的生產效率，使工廠能夠順利完成產品又能降低生產上不必要的浪費，達到最大的有效生產效率。

在許多生產排程方法中，製造單元系統(Manufacturing Cell System)為群組技術(Group Technology)與生產排程的重要應用之一，過去的相關研究指出，製造單元系統具備著流線型(Flowshop)生產的效率與零工式(Jobshop)生產的彈性。除了結合了流線型生產與零工式生產的優點外，製造單元系統亦具有降低設置時間、減少再製品存貨、簡化生產排程規劃與提高機器使用率等優點。因此製造單元生產排程問題(Manufacturing Cell Scheduling Problem)近年來引起學者與業界的注意，並開始廣泛的應用於工廠的生產規劃排程，使得製造單元生產排程問題成為生產排程領域的核心研究問題。

製造單元系統其概念為群組技術的應用，在生產規劃時，工件(Jobs / Parts)的規劃並不像傳統的生產排程規劃，以單一工件為單位進行加工，而是須以群組的概念加以處理，亦即將具有相同製造性質的工件以工件族(Jobs / Parts Family)的型態進行加工處理。因此規劃製造單元生產排程時，必須考慮兩個層面，即工件族之間(Among Family)的加工順序與工件族內工件(Within Family)的加工順序，所以在規劃生產排程上，比起傳統的生產排程問題更加複雜，加上製造單元排程問題本身即為 NP-hard 問題，即隨著工件數越多，越難利用數學規劃方法在合理時間內求解。為了有效率求解，近年來學者多以巨集啟發式演算法解決求解上的問題。雖然巨集啟發式演算法只能求得近似解，無法像數學規劃法求得最佳解，但由於求解時間快，能夠在合理時間內找到近似最佳解，因此廣泛的被學者所利用，而常見解決製造單元排程問題的巨集啟發式演算法，例如基因演算法(Genetic Algorithms; GA)、塔布搜尋法(Tabu Search; TS)、模擬退火法(Simulated Annealing;

SA)、濶集演算法(Memetic Algorithms ; MA)等巨集啟發式演算法。本論文為了解決製造單元排程系統在求解上的困難，也是使用巨集啟發式演算法求解，利用塔布搜尋法模擬整個流程。

1.2 研究目的

對於製造單元排程問題，過去使用巨集啟發式演算法作為求解方式的學者，主要研究集中於演算法進化機制的探討，例如使用基因演算法求解，討論不同染色體交配方式的優劣。這些演算法進化機制的研究，對於製造單元排程問題能夠有效的解決此問題的困難點和求得不錯的求解品質，但對於巨集啟發式演算法的使用，解決問題的困難點與改善求解品質也可以從解的表達方式解決，也就是一組工件的加工順序亦可利用不同的解表達方式表達，在相同的模擬流程下，亦可得到不同的求解結果。對於解的表達法研究，過去學者較少著重於此區塊，本論文以 Lin *et al.* (2009)使用之舊表達法與 Wu *et al.* (2011)使用之新表達法作為本論文的研究依據。比較舊表達法與新表達法對於求解品質的差異，並對於 Wu *et al.* (2011)提出，在基因演算法下，新表達法優於舊表達法之論點，驗證塔布搜尋法下是否也能得到相同之論點。

1.3 研究問題

本篇論文主要探討，如何應用兩種不同的解表達法應用於製造單元排程背景，其排程背景再加工流程上為流線型生產排程，流線型生產的概念為工件依機台順序依序在機台上加工處理，如圖 1.1 所示，每個工件皆從機台 1 進入，依機台順序，經過機台 1、機台 2 與機台 3 加工，最後皆由機台 3 產出，即為流線型生產排程。製造單元排程背景本身為群組技術的重要應用，在工件的加工順序與整備時間上，皆必須考慮工件族與工件族內工件兩個層面。對於加工順序，我們考慮固定序列(Permutation)特性的加工模式，其概念即工件進入時，第一部機台即決

定工件族與工件族內工件的加工順序，後續機台皆使用相同加工順序。而對於每部機台其工件族與工件族內工件的設置時間，我們考慮順序相依家族整備時間 (Sequence Dependent Family Setup Time)，而不是順序獨立家族整備時間 (Sequence Independent Family Setup Time)，主要在於，順序獨立家族整備時間對於工件族的加工順序不同，無法表達整備時間不同所帶來生產上的差異，但在實務上而言，對於相同機台，工件族加工順序的不同，會影響機台的整備時間，進而影響整個生產的完工時間與生產進度，因此我們考慮順序相依家族整備時間，即工件族的整備時間會因為前一個不同的工件族，而有不同的整備時間。對於同一工件族內工件的整備時間，由於同一工件族內工件的製造性質相同，所以在同一部機台，同一工件族內的工件，無整備時間。因此本篇論文研究的排程背景為具順序相依家族整備時間與固定序列的流線型製造單元排程問題。



圖 1.1 流線型生產

1.4 論文架構介紹

本論文內容總共五個章節。第二章則針對我們研究的議題，進行相關文獻的回顧與探討。第三章為本研究的研究方法，敘述舊表達法與新表達法的差異，並描述本研究的整個模擬流程。第四章則為利用演算法模擬後所得到之實驗結果，分析舊表達法與新表達法在求解品質的優劣。第五章則為本研究未來繼續研究的方向。

第二章 文獻探討

本章將分成五個小節進行探討，首先針對本研究議題，說明何謂製造單元系統；接著針對本研究的製造單元的特性加以描述，分成兩節，第二節，流線型(Flowshop)生產系統分類；第三節，家族整備時間分類；對於本論文研究主題的求解方式進行簡單描述，分成兩節，第四節，解的表達法分類；第五節，本研究所使用之塔布演算法探討。

2.1 製造單元系統

製造單元系統屬於群組技術的重要應用。工件的分類會依工件的製造性質，將具有相同製造性質的工件群組成同一個工件族，以工件族為單位進行加工，例如需要使用相同的機台工具、整備時間或相同加工流程的工件，將其群組一起加工。因此製造單元系統的排程須決定兩個層面，第一個層面為工件族之間的加工順序(Among Family Sequence)，另一個層面為工件族內工件的加工順序(Within Family Sequence)，故製造單元系統在許多文獻上皆已證明為一 NP-hard 問題。

求解製造單元系統可分成最佳解法與近似解法，最佳解法主要以數學規劃法為主，優點在於可求得最佳解，缺點在於模型建構不易，求解時間過久等問題，因此在求解製造單元系統，主要是以近似解法為主，近似解法可分成 Heuristic 與 Meta-heuristic，其中又以 Meta-heuristic 為主要求解方法，相關求解製造單元文獻整理於表 2.1。

| 主要概念 | 作者(年份) | 演算法 |
|----------------------|---------------------------------|---|
| Optimization Method | Webster & Azizoglu (2001) | Dynamic Programming |
| | Janiak <i>et al.</i> (2005) | Linear Programming |
| | Das & Canel (2005) | Branch and Bound |
| Approximation Method | Schaller <i>et al.</i> (2000) | Genetic algorithm, Memetic algorithm |
| | Franca <i>et al.</i> (2005) | Tabu search |
| | Hendizadeh <i>et al.</i> (2008) | Tabu search, |
| | Lin <i>et al.</i> (2009) | Tabu search, Genetic algorithm |
| | Wu <i>et al.</i> (2011) | Genetic algorithm |

表 2.1 製造單元系統文獻

2.2 流線型生產分類

對於流線型生產系統，在機台上加工順序的排列可分為固定序列與非固定序列，兩者的差異在於，固定序列在不同機台上，其加工順序皆一樣，如圖 2.1 所示，而非固定序列可為不一樣的加工順序，如圖 2.2 所示。針對這兩種不同的排程方式，本篇論文主要探討固定序列排程問題。又固定序列排程問題，工件的排序上，可分成具家族分類與不具家族分類，而製造單元排程問題屬於群組技術的應用，須將工件依其製造性質分類成不同的工件族群，因此本篇論文考量了家族分類的概念。其相關文獻分類如表 2.2。

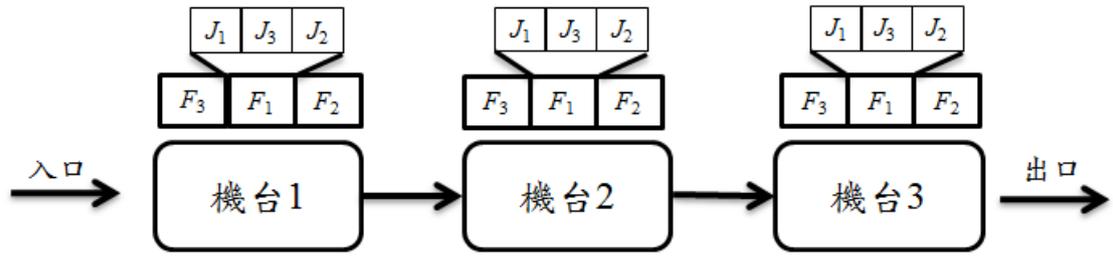


圖 2.1 固定序列

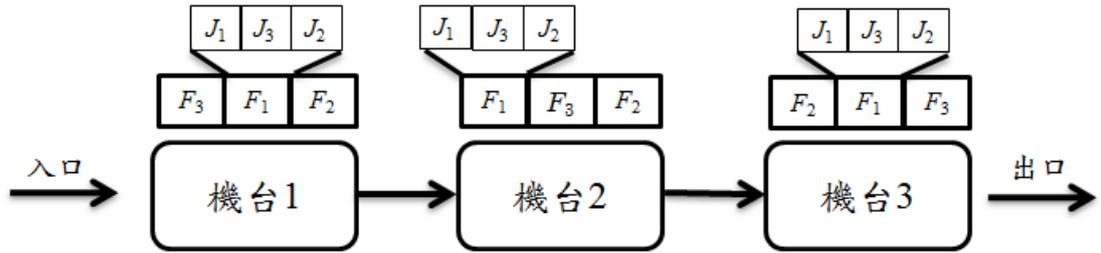


圖 2.2 非固定序列

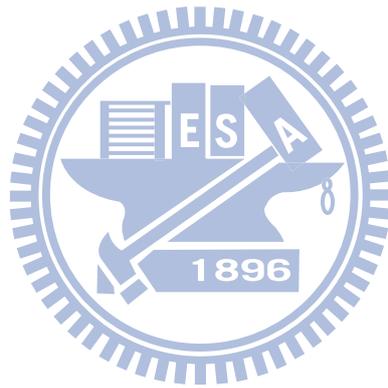


表 2.2 固定序列流線型分類

| 主要概念 | 作者(年份) | 演算法 | 目標值 |
|------------|---------------------------------|---|-----------------------------|
| non-family | Grabowski & Wodecki (2004) | Tabu search | makespan |
| | Jarboui <i>et al.</i> (2008) | Particle swarm algorithm | makespan, total flowtime |
| | Zhang <i>et al.</i> (2009) | Genetic algorithm | total flowtime |
| | Tseng & Lin (2009) | Genetic algorithm | makespan |
| | Naderi & Ruiz (2010) | Genetic algorithm | makespan, total flowtime |
| family | Schaller <i>et al.</i> (2000) | 派工法則 | makespan |
| | Franca <i>et al.</i> (2005) | Genetic algorithm, Memetic algorithm | makespan |
| | Hendizadeh <i>et al.</i> (2008) | Tabu search | makespan |
| | Lin <i>et al.</i> (2009) | Tabu search, Genetic algorithm | makespan |
| | Wu <i>et al.</i> (2011) | Genetic algorithm | makespan |

2.3 家族整備時間分類

根據 Cheng *et al.* (2000)整理，流線型生產整備時間可分類成四種：順序獨立工件整備時間(Sequence Independent Job Setup Time)、順序相依工件整備時間(Sequence Dependent Job Setup Time)、順序獨立家族整備時間 (Sequence Independent Family Setup Time) 和順序相依家族整備時間(Sequence Dependent Family Setup Time)。

本論文在排序加工流程時，先針對工件依其製造性質先分類成同一家族，再

排序家族內的工件。對於工件族的順序，機台整備時間應該是由正在加工的工件族與下一個被加工的工件族共同決定，所以順序相依家族整備時間比順序獨立家族整備時間更貼近實務情況。而同一工件族內的工件，因為製造性質相近，故機台整備時間相對於不同工件族轉換的機台整備時間小很多，可以忽略不計。其家族整備時間可分類如表 2.3。

表 2.3 家族整備時間

| 主要概念 | 作者(年份) | 演算法 | 目標值 |
|--------------|------------------------------------|---|----------------|
| 順序獨立 整備時間 | Cheng & Wang (1998) | 動態規劃法 | makespan |
| | Gupta & Schaller (2006) | 派工法則 | tatal flowtime |
| 順序相依 整備時間 | Schaller <i>et al.</i> (2000) | 派工法則 | makespan |
| | Franca <i>et al.</i> (2005) | Genetic algorithm, Memetic algorithm | makespan |
| | Hendizadeh <i>et al.</i> (2008) | Tabu search | makespan |
| | Lin <i>et al.</i> (2009) | Tabu search, Genetic algorithm | makespan |
| | Wu <i>et al.</i> (2011) | Genetic algorithm | makespan |

2.4 解的表達法分類

對於製造單元排程系統，規劃加工順序時，需考慮工件族與工件兩個層面，其加工順序即代表一組解的順序，可利用基因演算法中染色體的概念來表示。染色體的表達方式分成舊染色體表達法與新染色體表達法，舊染色體表達法以 Lin *et al.* (2009)使用的染色體為依據，本論文稱作舊表達法。新染色體表達法則以 Wu *et al.* (2011)使用之染色體為依據，本論文稱作新表達法。舊表達法每格基因

格存放工件族或工件編號，可區分成 $F+1$ 個區段， F 為工件族數。而舊表達法的第一個區段為工件族加工順序，後續區段依序為工件族 1、工件族 2、工件族 3 到工件族 F 內工件的加工順序。相對於舊表達法，新表達法不必分成許多區段，只需以一個區段表達即可，每格基因格存放工件的編號，每個工件會附帶一個屬性，其屬性即每個工件屬於哪個工件族群。對於舊表達法與新表達法在第三章有更進一步的介紹。表達法應用於排程中的文獻整理如表 2.4。

表 2.4 解表達法

| 主要概念 | 作者(年份) | 演算法 | 目標值 |
|------|-----------------------------|---|----------|
| 舊表達法 | Franca <i>et al.</i> (2005) | Genetic algorithm, Memetic algorithm | makespan |
| | Lin <i>et al.</i> (2009) | Tabu search, Genetic algorithm | makespan |
| 新表達法 | 呂佳玟 (2009) | Genetic algorithm | makespan |
| | 戴邦豪 (2010) | Genetic algorithm | makespan |
| | Wu <i>et al.</i> (2011) | Genetic algorithm | makespan |

2.5 塔布搜尋法(Tabu Search)

本論文的模擬流程即應用巨集啟發式演算法-塔布搜尋法模擬，其塔布搜尋法的介紹與應用將於 2.5.1 與 2.5.2 詳細介紹。

2.5.1 塔布搜尋法簡介

塔布搜尋法由 Glover (1977) 提出，為一種搜尋近似解的啟發式演算法，其原理主要來至，人類做決策時，會透過之前經驗的記憶，決定下一次決策方向的思維而創造出塔布搜尋法。塔布搜尋法已廣泛應用於不同的領域中，例如路徑指派問題(Path Assignment)、流線型排程(Flowshop Scheduling)、零工式排程(Jobshop Scheduling)、旅行者問題(TSP)與流線型製造單元排程問題(Flowshop

Manufacturing Cell Scheduling)等方面皆可利用塔布搜尋法快速求解。其搜尋方式是先從一組初始解開始，搜尋初始解附近的所有鄰近解，從此鄰近解中找尋一組最佳的鄰近解作為下次搜尋的起點，以此不斷的重複搜尋，直到達到搜尋終止條件。而在塔布搜尋法中，其經驗的記憶即塔布列表(Tabu List)的應用，塔布列表存放每次決策的移步(Move)，搭配塔布列表的長度設定，採取先進先出的方式，每做一次決策即更新塔布列表的移步，使移步在經過一定次數後，跳脫塔布列表的限制。因此透過塔布列表的記憶機制，可避免重複搜尋相同的區域與落入區域最佳解的機會，進而提高搜尋到全域最佳解的機會。塔布搜尋法的基本組成元素包括初始解(Starting Solution)、鄰近解(Neighbor Solution)、移步(Move)、塔布列表(Tabu List)、破禁準則(Aspiration Criterion)與結束準則(Stopping Criterion)，其基本元素介紹如下。

1. 初始解(Starting Solution)

即進行塔布搜尋法時，作為最一開始搜尋起點的解，稱為初始解，一般而言可透過其他已知巨集啟發式演算法取得。

2. 鄰近解(Neighbor Solution)

對於任意可行的初始解，鄰近起始解的可行解皆稱為鄰近解。而鄰近解的定義會依每個使用者的定義有所不同。例如：在某一機台上需處理工件 i 、工件 j 與工件 k ，起始加工順序先做工件 i 再加工工件 j 最後再加工工件 k ，故起始解為 $i \rightarrow j \rightarrow k$ ，而鄰近解的定義為左右工件互換，所以鄰近解即為 $\{j \rightarrow i \rightarrow k\}$ 與 $\{i \rightarrow k \rightarrow j\}$ 兩組鄰近解。塔布搜尋法會透過每次鄰近解的搜尋，對於使用者所設定的目標值，排序所有鄰近解的優劣，並先從最佳鄰近解開始找起，找尋適合下一次搜尋的起始解。

3. 移步(Move)

由目前解移動到另一個解的移動路徑，稱作移步。移步的方式可由問題的特性加以設計，常見的移動方式為兩點互換法(Swap)或插入法(Insert)。例如：起始解為 $i \rightarrow j \rightarrow k$ ，移步方式為左右互換，鄰近解 $\{j \rightarrow i \rightarrow k\}$ 與 $\{i \rightarrow k \rightarrow j\}$ ，若 $\{j \rightarrow i$

→ k 為較佳解，則移步即為 $\{i, j\}$ 。

4. 塔布列表(Tabu List)

為塔布搜尋的記憶機制，紀錄每次由一組目前解移動至另一組解的移步，在塔布列表中的移步，被禁止當作可行解，其目的在於禁止找尋在塔布列表中的移步，避免重新搜尋上一步或之前的解，造成迴圈的發生，陷入區域解中。而塔布列表的長度，不能為無限的長度，這樣會造成所有可行的移步皆在塔布列表中，學者 Glover 建議，塔布列表長度為 7 或 \sqrt{n} ，但可以針對不同的問題，調整其塔布列表的長度。對於塔布列表內的移步，每搜尋到一個移步即會更新塔布列表，並採取先進先出的方式，以便當塔布列表已經填滿時，會將最先進入的移步，解除在塔布列表中的限制，更新塔布列表。如圖 2.3 所示。



圖 2.3 塔布列表運作方式

5. 破禁準則(Aspiration Criterion)

透過塔布列表的設計，可以避免搜尋陷入迴圈與區域最佳解中，可是在搜尋過程中，某些在塔布列表中的移步，可以產生很好的可行解，但因為被塔布列表所

限制，以至於我們不能採取此組可行解，為了避免這種情況發生，故設定條件，使得移步可以跳脫塔布列表，往更好的可行解方向搜尋。例如：塔布名單中有 $\{\{1,2\}, \{4,5\}, \{3,7\}\}$ ，若移步 $\{4,5\}$ 為此鄰近解中的最優解，且大於目前最佳解時，就符合破禁準則，解除此移步之限制，使其成為可接受的解。而破禁準則一般而言可分成三類。

(1) 最小限制的破禁原則

當所有移步均被列入塔布列表中，且其它移步破禁準則也不符合時，可採取此方式，即將最先進入的移步解除在塔布列表的限制。

(2) 目標值改善破禁原則

其原則為此移步的可行解所產生的目標值優於最佳解時，即可將此移步解除限制，接受此移步。

(3) 固定搜尋方向

假設只往某一固定方向進行搜尋，在此方向，即使移步在塔布列表內，能可接受此移步，不受塔布列表限制。

6. 結束準則(Stopping Criterion)

即整個演算法的停止條件。一般而言，可分成三種結束準則。

- (1) 搜尋計算次數達到設計者所設定之總次數。
- (2) 無法更新最佳解之次數，達到設計者所設定之次數。
- (3) 扣除在塔布列表中的可行解，其他可行的鄰近解為空集合。

2.5.2 塔布搜尋法應用於生產排程

求解生產排程的方法有許多種方式，常見可分成兩種，一為利用數學方法求解最佳化，例如線性規劃法(Linear Programming)、動態規劃法(Dynamic Programming)、分枝界限法(Branch and Bound)等其他數學方式。另一種則為利用巨集啟發式演算法求解近似解，例如基因演算法(Genetic algorithm)、塔布搜尋法(Tabu Search)等其他已知巨集啟發式演算法。對於簡易的生產排程問題，可透

過數學方式建構數學模式求解最佳解，然而，實際的生產環境，需考慮的條件相當多，例如工件的數量、產品類別與資源等條件皆須仔細考慮。若此時使用數學方式求解，則會造成求解模式不易建構與求解時間過長等缺點。巨集啟發式演算法可解決求解時間過長與求解更複雜的生產系統，並找到可容許的誤差範圍下的近似解，因此在求解複雜的生產系統時，巨集啟發式演算法較為被其他學者所使用。本論文亦是使用巨集啟發式演算法求解具順序相依家族整備時間與固定序列流線型製造單元排程問題，所使用之演算法為塔布搜尋法，其塔布搜尋法的相關文獻整理如表 2.5。

表 2.5 塔布搜尋法

| 作者(年份) | 演算法 | 目標值 | 內容簡述 |
|---------------------------------|-------------|--------------|---|
| Grabowski & Wodecki (2004) | Tabu search | Makespan | 針對流線型生產的一些特性，建構一快速求解的布搜尋法 |
| Hendizadeh <i>et al.</i> (2008) | Tabu search | Makespan | 塔布搜尋為基礎的演算法，求解具順序相依家族整備時間與固定序列流線型製造單元系統 |
| Lin <i>et al.</i> (2009) | Tabu search | Makespan | 以塔布搜尋法求解具順序相依家族整備時間與非固定序列流線型製造單元系統 |
| Xu <i>et al.</i> (2010) | Tabu search | Dudate-based | 使用於單一機台的排程，透過資源的利用控制加工時間 |

第三章 研究方法

本章節將介紹具順序相依家族整備時間與固定序列流線型製造單元系統，如何利用舊表達法與新表達法表示工件加工順序，解決此問題。本章節分成三個章節，第一節，研究問題描述；第二節，研究限制與模型假設；第三節，解的表達法設計與解讀；第四節，舊表達法設計與解讀；第五節，新表達法設計與解讀；第六節，計算目標值；第七節，塔布搜尋法流程。

3.1 研究問題敘述

本論文研究的排程背景為具順序相依家族整備時間與固定序列流線型製造單元排程問題，製造單元系統如同 2.1 所敘述，在規劃生產排程時必須同時決定工件族之間的加工順序與工件族內工件的加工順序，因此就製造單元系統，已經是一種 NP-hard 問題，又本研究考慮了順序相依家族整備時間與固定序列的想法，使整個問題更加趨於複雜。因此對於工件與工件族在機台上加工流程的排序，需以簡單又適合的表達法代表加工順序，以降低在排序工件與工件族加工順序的複雜。因此表達法的應用即為本論文的重點所在。其求解具順序相依家族整備時間與固定序列流線型製造單元系統的順序如圖 3.1 所示。

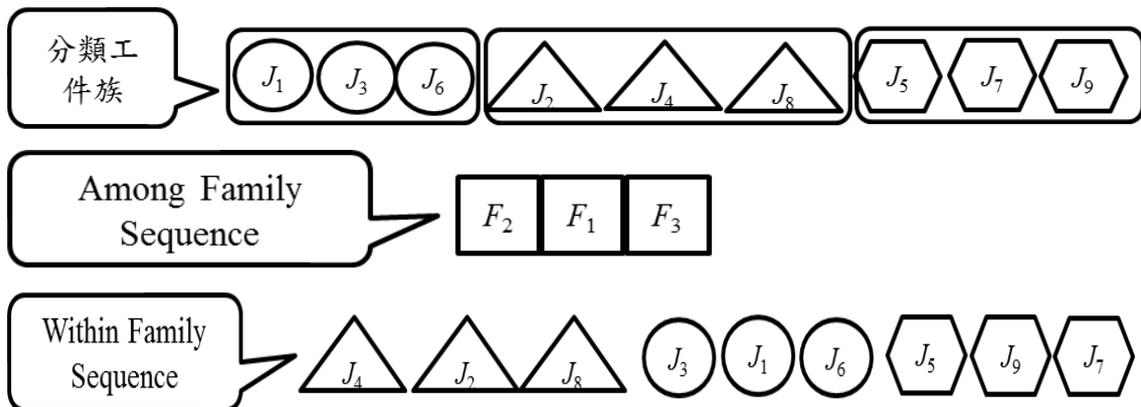


圖 3.1 求解製造單元系統過程

3.2 研究限制與模型假設

本論文的排程背景其研究假設與研究模型的建構說明如下所示。

- (1) 本研究是屬於靜態分析，即工件與工件族的數量、工件在機台上的加工時間與工件族在不同機台上的整備時間皆已知，且為非負整數。
- (2) Ready Time=0，即各機台最一開始的工件一到，即可馬上加工處理。
- (3) 每個工件一次只能經由一部機台加工，每部機台一次只能加工一個工件。
- (4) 同一工件族內的工件不可被分割，即必須處理完同一個工件族內的工件後，才可處理下一個工件族。
- (5) 機台無當機現象。

在以上的研究限制下，本研究的目標為求解最小化最大完工時間

(makespan)，其研究模型建構如下：

考慮 n 個工件的集合 $N = \{J_1, \dots, J_n\}$ 與 m 部機台的順序相依家族整備時間與固定序列之流線型製造單元排程問題，其工件會依製造性質，將使用相同技術的工件分類成同一個工件族，所以可將所有工件分類成 q 個互斥且獨立的工件族集合 $F = \{f_1, \dots, f_q\}$ ，工件族 f_k 會有 n_k 個工件 ($k = 1, 2, \dots, q$)，工件按照工件編號依序分類至工件族內，例如前 n_1 個工件屬於第 f_1 個工件族，尚未排序的工件再依序編號排入工件族內，也就是，前 n_2 個工件屬於第 f_2 個工件族，依此排序，最後 n_q 個工件屬於第 f_q 個工件族。因此總工件數 $n = n_1 + n_2 + \dots + n_q$ 。工件 J_i 在機台 j 的加工時間為 p_{ij} 。而我們考慮了順序相依家族整備時間，在機台 j 工件族 x 轉換為工件族 y 其整備時間為 s_{xy}^j 。若兩工件皆屬於工件族 x ，則在機台 j 上的整備時間為 $s_{xx}^j = 0$ ，亦即只有在不同工件族交換時，才需考慮機台整備時間。

因此我們考慮一個工件排程順序 $\sigma = \{\sigma(1), \dots, \sigma(n)\}$ ， $\sigma(i)$ 在機台 j 的完工時間 $C(\sigma(i), j)$ ，計算方式如公式(1)：

$$C(\sigma(i), j) = \max\{C(\sigma(i), j-1), C(\sigma(i-1), j) + s_{xy}^j\} + p_{\sigma(i), j} \quad (1)$$

where $\sigma(i) \in f_y, \sigma(i-1) \in f_x$, and $C(\emptyset, j) = C(\sigma(i), 0) = 0$ for all i and j

透過公式(1)，即可求出此組工件加工順序的完工時間，而我們的目標為求解最小化最大完工時間(makespan)，即求解 $C(\sigma(n), m)$ 。

3.3 表達法設計與解讀

本論文的研究重點在於，相同的塔布流程下，以不同的表達法表示工件族與工件的加工順序，評估不同的表達法對於目標值的優劣。其表達法，本論文分成舊表達法與新表達法兩種，舊表達法以 Lin *et al.* (2009)為依據，而新表達法則以 Wu *et al.* (2011)為依據。不論是哪一種表達法來表達工件加工順序，其求解目標值時皆須經過表達法設計、表達法解讀與計算目標值三個步驟，其流程如圖 3.2 所示。對於舊表達法與新表達法的設計與解讀，將一一介紹。



圖 3.2 計算流程

3.4 舊表達法設計與解讀

本小節將詳細介紹舊表達法的組成與如何透過舊表達法的解讀，產工件的加工順序。

3.4.1 舊表達法設計

舊表達法由一條分成 $F+1$ 個區段所組成， F 為工件族數，而每一格可能為一個工件族編號或一個工件編號。每一個區段的內容皆不相同，第一個區段放置工件族的加工順序，後續的區段，依序為工件族 1 內的工件加工順序、工件族 2 內的工件加工順序、...、工件族 F 內的工件加工順序。因此一條舊表達法，其表達方式如圖 3.3 所示。

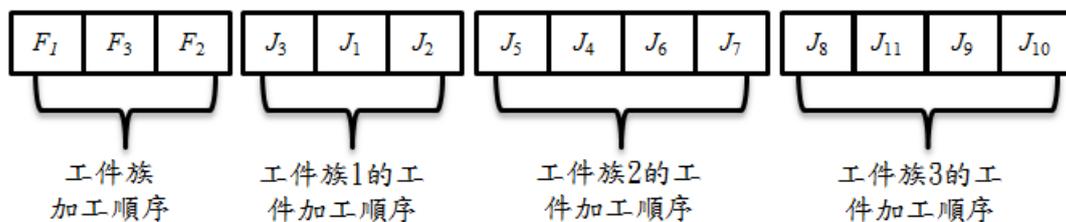


圖 3.3 舊表達法

3.4.2 舊表達法解讀

由於舊表達法第一區段即表示為工件族的加工順序，其餘區段代表每個工件族內的工件加工順序，因此在舊表達法解讀上，依照第一個區段所排的工件族加工順序，即可知道生產過程的加工順序。以圖 3.3 所示為例，在一個區段工件族加工順序為 F_1 、 F_3 最後 F_2 ，因此先加工工件族 1 內的工件，再加工工件族 3 內的工件，最後加工工件族 2 內的工件，即舊表達法的工件順序， J_3 - J_1 - J_2 - J_8 - J_{11} - J_9 - J_{10} - J_5 - J_4 - J_6 - J_7 。其解讀方式如圖 3.4 所示。

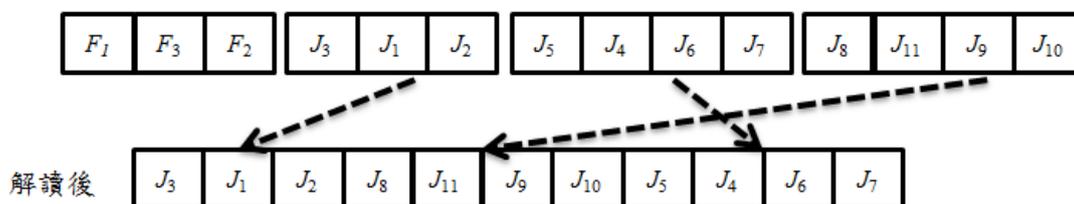


圖 3.4 舊表達法解讀

3.5 新表達法設計與解讀

本小節將詳細介紹新表達法的組成與舊表達法的差異，並說明如何透過新表達法的解讀，產工件的加工順序。

3.5.1 新表達法設計

新表達法相對於舊表達法而言，不必區分成許多區段，以一個區段表示即可，每格格子存放著工件編號，其每個工件亦會附帶一個屬性，屬性即代表該工件屬於哪個工件族，因此在表達法的解讀上，就可靠此屬性分類工件族。新表達法如

圖 3.5 所示。

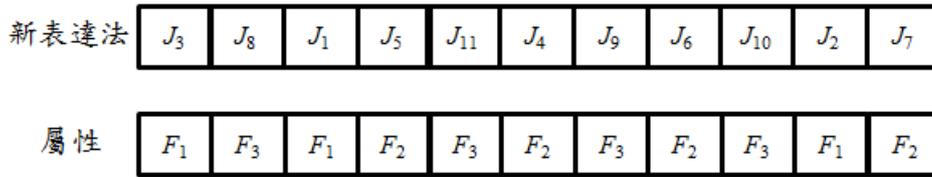


圖 3.5 新表達法

3.5.2 新表達法解讀

新表達法不像舊表達法，已知工件族的加工順序與各工件族內工件的加工順序，因此在解讀上須依靠所附帶的屬性加以解讀，可分成兩個步驟，步驟一為工件族的加工順序解讀，步驟二為工件族內工件的加工順序解讀。以圖 3.5 所示之新表達法為例。

步驟一：工件族加工順序

首先針對每個工件族所附帶的屬性，先分類其工件族加工順序。其方法為，由左至右依序找出第一次出現的工件族編號。因此由圖 3.5 所示為例，第一個出現的工件族編號為工件族 1，下一個出現的工件族編號為工件族 3，由於第一次出現，所以採取工件族 3，再下一個出現的工件族編號為工件族 1，但因為工件族 1 已經出現過，所以不採用，接著再依序尋找下一個工件族編號，下一個工件族編號為工件族 2，第一次出現，所以採取工件族 2，因此可以得到一組工件族的加工順序為 $F_1-F_3-F_2$ ，即排出工件族在機台上的加工順序。工件族解讀過程如圖 3.6 所示。

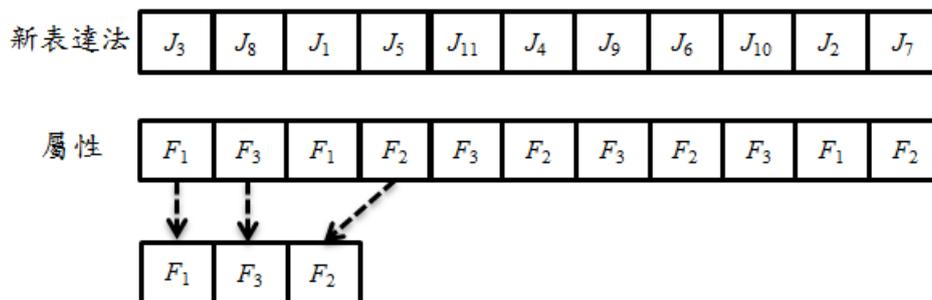


圖 3.6 新表達法工件族解讀

步驟二：工件族內工件的加工順序

決定工件族的加工順序後，即可排序工件族內工件的加工順序。其方法與找尋工件族加工順序一樣，由左至右，找尋該工件族第一次出現的工件，即該工件族先出現的工件先排入加工。由步驟一得知，先做工件族 1，再做工件族 3，最後做工件族 2，因此排序工件族 1 內的工件時，由左至右，依序出現的工件為 J_3 、 J_1 與 J_2 ，故工件族 1 內的工件加工順序為 J_3 - J_1 - J_2 。排序工件族 3 內的工件順序，與排序工件族 1 的方法一樣，由左至右，依序出現的工件為 J_8 、 J_{11} 、 J_9 與 J_{10} ，因此工件族 3 內工件加工順序為 J_8 - J_{11} - J_9 - J_{10} 。最後排序工件族 2 內工件順序，其順序為 J_5 - J_4 - J_6 - J_7 。因此即可得知各工件族內工件的加工順序。透過步驟一與步驟二，此排程的加工順序即 J_3 - J_1 - J_2 - J_8 - J_{11} - J_9 - J_{10} - J_5 - J_4 - J_6 - J_7 。其解讀過程如圖 3.7 所示。

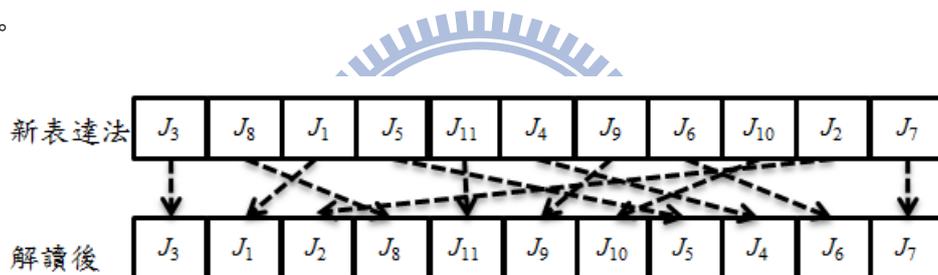


圖 3.7 新表達法工件族內工件的解讀

3.6 計算目標值

不論使用哪一種表達法表達工件加工順序，最後皆需計算其目標值評量此組加工順序是否為一組好的加工順序。本篇的目標值為最小化 makespan，求解 makespan 的公式為 3.1 的公式一。

3.7 塔布搜尋法流程

對於本論文研究主題，是採取巨集啟發式演算法模擬整個排程問題，而所採取的巨集啟發式演算法為塔步搜尋法，其演算邏輯引用 Glover (1989) 的文獻。塔布搜尋法的基本元素概念已在 2.5.1 介紹，本節將針對一些基本元素如何應用於

本論文加以說明。

3.7.1 塔布搜尋法基本元素應用

1. 初始解(Starting Solution)

使用巨集啟發式演算法時，初始解的產生通常都以隨機方式產生符合題意的解。本論文的解表達法有兩種不同的形式，若以隨機方式各自產生初始解，即使在相同的 random number 下產生，其初始解並不相同，可能會造成結果上的誤判。因此為了消除此影響因素，第一組初始解，必須使新表達法與舊表達法為同一組解。本論文使用的方法為，新表達法的初始解以隨機方式產生，舊表達法則是從新表達法轉換而來。其轉換步驟與新表達法解讀成加工順序步驟類似，以圖 3.8 為例，首先從新表達法解讀出工件族的加工順序，其方法為，由左至右，找尋第一次出現的工件族編號，因此可得工件族順序 F_1 - F_3 - F_2 。得到工件族順序後，再解讀各工件族內的工件順序，其方法與解讀工件族順序一樣，因此工件族 1 內的工件順序為 J_3 - J_1 - J_2 ，工件族 2 內的工件順序 J_5 - J_4 - J_6 - J_7 ，工件族 3 內的工件順序 J_8 - J_{11} - J_9 - J_{10} ，如圖 3.9 所示。解讀出工件族加工順序與各工件族內工件加工順序後，即可依舊表達法每各區段的意義，轉換成舊表達法的表達方式，如圖 3.10 所示。

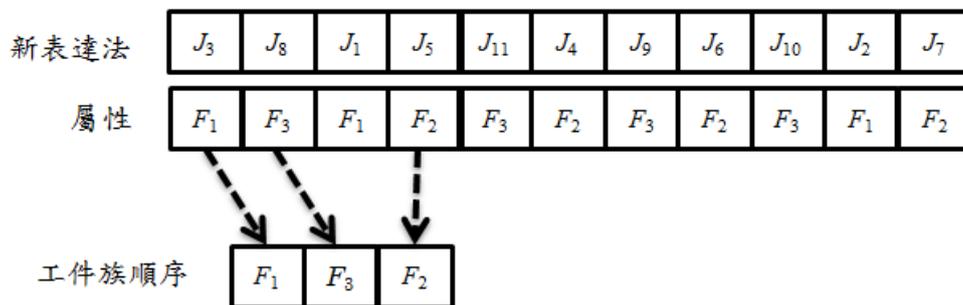


圖 3.8 舊表達法初始解-工件族順序



圖 3.9 舊表達法初始解-工件族內工件順序

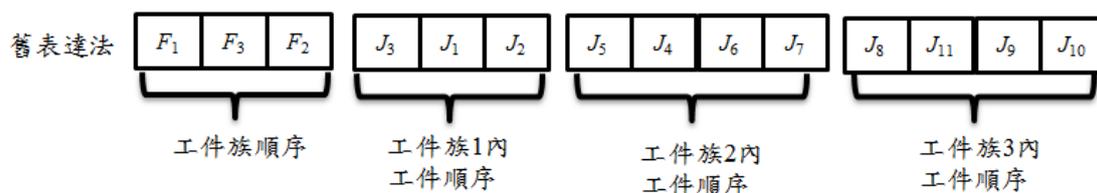


圖 3.10 舊表達法的初始解

2. 鄰近解(Neighbor Solution)

採取兩兩互換方式。例如：在某一機台上需處理工件 i 、工件 j 與工件 k ，起始加工順序先做工件 i 在加工工件 j 最後在加工工件 k ，故起始解為 $i \rightarrow j \rightarrow k$ ，依本論使用之鄰近解定義，鄰近解有 $\{j \rightarrow i \rightarrow k\}$ 、 $\{k \rightarrow j \rightarrow i\}$ 與 $\{i \rightarrow k \rightarrow j\}$ 三組鄰近解。本研究採取舊表達法與新表達法表示工件加工順序，兩種不同的表達法，在搜尋鄰近解上皆採用兩兩互換方式，但由於表達法不一樣，因此在鄰近解的搜尋上也有些不同，以下將介紹這兩種表達法搜尋鄰近解的方式。

(1) 新表達法的鄰近解

於 3.5 我們已經介紹了新表達法，且已知新表達法無須分段，以一區段表示即可，因此在鄰近解的搜尋上，只需兩兩互換即可找到所有鄰近解。例如：已知一組新表達法其工件順序為 $J_1-J_2-J_3$ ，依本論文的鄰近解定義，第一組鄰近解為 J_1 與 J_2 交換，因此可得第一組鄰近解 $J_2-J_1-J_3$ 。第二組鄰近解為 J_1 與 J_3 交換，因此第二組鄰近解 $J_3-J_2-J_1$ 。到此 J_1 與其他工件皆交換過，因此 J_1 已經可以不須交換了。而第三組鄰近解為 J_2 與 J_3 交換，得第三組鄰近解 $J_1-J_3-J_2$ 。因此對於新表達法，鄰近解只需工件兩兩互相交換即可找出所有鄰近解，而每一個工件又會附帶一個屬

性，再透過新表達法的解讀，即可排出該組鄰近解工件族的加工順序與工件族內工件的加工順序，最後在計算該組鄰近解的目標值。新表達法的鄰近解搜尋過程如圖 3.8 所示。



圖 3.11 新表達法鄰近解

(2) 舊表達法的鄰近解

相對於新表達法，舊表達法需分成許多區段，又因為每個區段所代表的意義不同，在做鄰近解的搜尋時，如果與新表達法相同方式，則會發生不合理的情況，例如，工件族 1 內的工件與工件族 2 內的工件交換，交換後，工件族 1 內的工件交換到工件族 2，而工件族 2 內的工件交換到工件族 1，但工件族 1 內的工件與工件族 2 內的工件，在製造性質上並不相同或類似，所以不能將這兩類工件族的工件放置一起加工，因此在搜尋舊表達法的鄰近解時，需分段做兩兩交換。由於需分段做兩兩交換，因此在找一組鄰近解時，如果同時每一段都各做兩兩交換，則鄰近解的空間實在太大，無法有效找鄰近解，因此為了解決此一問題，本論文採用某一區段交換，其餘區段不交換的方式，提高搜尋鄰近解的效率。做法為依區段順序，選定區段，先做選定區段內的兩兩交換，其餘區段不交換，做完該區段的兩兩交換後，再依區段順序選定下一個區段，以此重複，直至所有區段皆做完兩兩交換。以圖 3.9 所示為例，依區段順序，先選定第一個區段，及工件族加工順序的區段，其加工順序為 F_1 - F_2 ，

兩兩交換後加工順序為 F_2-F_1 ，至於其他區段則不必做交換，因此第一組鄰近解的工件族加工順序為 F_2-F_1 ，工件族 1 內工件的加工順序為 $J_3-J_1-J_2$ ，工件族 2 內工件的加工順序為 J_5-J_4 ，此鄰近解在機台上的加工順序即 $J_5-J_4-J_3-J_1-J_2$ 。到此第一個區段已交換完所有鄰近解，則依區段順序，做下一個區段做兩兩交換，其概念與做第一個區段的方式相同。下一個區段為工件族 1 內工件的加工順序，工件加工順序為 $J_3-J_1-J_2$ 。首先工件族 1 交換 J_3 與 J_1 ，其餘區段不做交換，所以第二組鄰近解，工件族加工順序為 F_1-F_2 ，工件族 1 內工件加工順序為 $J_1-J_3-J_2$ ，工件族 2 內工件加工順序 J_5-J_4 ，因此第二組鄰近解在機台上的加工順序即 $J_1-J_3-J_2-J_5-J_4$ 。依相同概念，第三組鄰近解在機台上的加工順序為 $J_2-J_1-J_3-J_5-J_4$ 。第四組鄰近解的加工順序為 $J_3-J_2-J_1-J_5-J_4$ 。此時交換完第二個區段內的工件，依序找下一個區段，因此第五組鄰近解加工順序為 $J_3-J_1-J_2-J_4-J_5$ 。以此方式即可快速找出舊表達法的鄰近解，最後每一組鄰近解再透過舊表達法解讀即可算出目標值。



圖 3.12 舊表達法鄰近解

3. 塔布列表

Glover(1977)學者建議塔布列表的長度為 7 或 \sqrt{n} ，由於本研究在鄰近解

的定義上為兩兩互相交換，所以一組初始解所產生的鄰近解相當多，故若只將塔布列表長度設定為 7 或 \sqrt{n} ，會過早收斂，陷入區域解中，因此無法有效利用塔布列表的功用，即避免陷入區域最佳解中。因此本論文採取 Hendizadeh *et al.* (2008) 提出的概念，其概念為，依鄰近解的數量乘以某倍數後的數值作為塔布列表的長度。本論文的應用在概念上相似，工件族數與工件數 10 個以下，其塔布列表的長度為，鄰近解數量乘以三分之一。工件數 10 個以上，則為工件數乘上某一倍數，倍數隨工件數的範圍分成 2 倍、3 倍、5 倍、7 倍、9 倍、11 倍與 13 倍。其塔布列表的長度如表 3.1 所示。由於有兩種不同的解表達方式，因此塔布列表的數量有所不同。對於新表達法，由於不必分區段，因此塔布列表只需一個即可。而舊表達法，由於需分成 $F+1$ 個區段，因此會有 $F+1$ 個塔布列表。

表 3.1 塔布列表大小

| 工件族數 | 塔布列表大小 | 工件數 | 塔布列表大小 | 工件數 | 塔布列表大小 |
|------|--------|-------|--------|-------|--------|
| 3 | 1 | 2 或 3 | 1 | 11~14 | 工件數*2 |
| 4 | 2 | 4 | 2 | 15~19 | 工件數*3 |
| 5 | 3 | 5 | 3 | 20~29 | 工件數*5 |
| 6 | 5 | 6 | 5 | 30~39 | 工件數*7 |
| 7 | 7 | 7 | 7 | 40~49 | 工件數*9 |
| 8 | 9 | 8 | 9 | 50~59 | 工件數*11 |
| 9 | 12 | 9 | 12 | >60 | 工件數*13 |
| 10 | 15 | 10 | 15 | | |

4. 破禁準則

此世代所選定的鄰近解，其目標值優於最佳解且在塔布列表內，即符合破禁準則。此時會先將此移步從塔布列表中移除，調整塔布列表，最後再將

此移步重新放入塔布列表內。

5. 結束準則

最佳解連續 2000 代不變即結束整個模擬搜尋。每一世代的定義為搜尋一次完整的塔布搜尋，即從一組初始解，找尋此初始解附近的鄰近解，透過鄰近解再找出下一次搜尋的起始解，即為一次完整的塔布搜尋。

3.7.2 塔布搜尋法求解步驟

對於相關基本元素定義後，即利用塔布搜尋法模擬本研究問題，其求解步驟如下所示，求解流程圖如圖 3.10 所示。

塔布搜尋法求解步驟：

步驟一

- i 產生一組可行的初始解。
- ii 此初始解為目前最佳解並求出目標值。
- iii 設定終止條件。
- iv 設定塔布列表大小並為空集合。



步驟二

- i 透過鄰近解定義，搜尋鄰近的鄰近解，若無可行解，則跳至步驟五。若有鄰近解，計算每組鄰近解的目標值。
- ii 依鄰近解的目標值，排序鄰近解的優劣，尋找出鄰近解中最佳解，並檢查此鄰近解的移步是否在塔布列表中。若不在塔布列表中，跳至步驟三。若在塔布列表中，則跳至步驟四。

步驟三

- i 若此鄰近解的目標值優於目前最佳解，則取代目前最佳解，反之，則不取代，保存目前最佳解。
- ii 將此鄰近解的移步列入塔布名單中，並將此鄰近解作為下次搜尋的參考點。

iii 若符合結束準則，跳至步驟五，反之，回步驟二。

步驟四

- i 若符合破禁準則，則解除此鄰近解的移步，並將此鄰近解的目標值更改為目前最佳解，更新塔布列表，將此鄰近解作為下次搜尋的參考點。若符合結束準則，跳至步驟五，反之，回步驟二，繼續搜尋。
- ii 若不符合破禁準則，則依步驟二所排序的鄰近解目標值優劣，找下一組較好的鄰近解，並檢查此鄰近解的移步是否於塔布列表中，若是，重覆步驟四，若不是，重覆步驟三。

步驟五

- i 若符合結束準則，最佳解連續 2000 代不變，結束塔布搜尋。
- ii 目前最佳解即為塔布搜尋之最佳解。



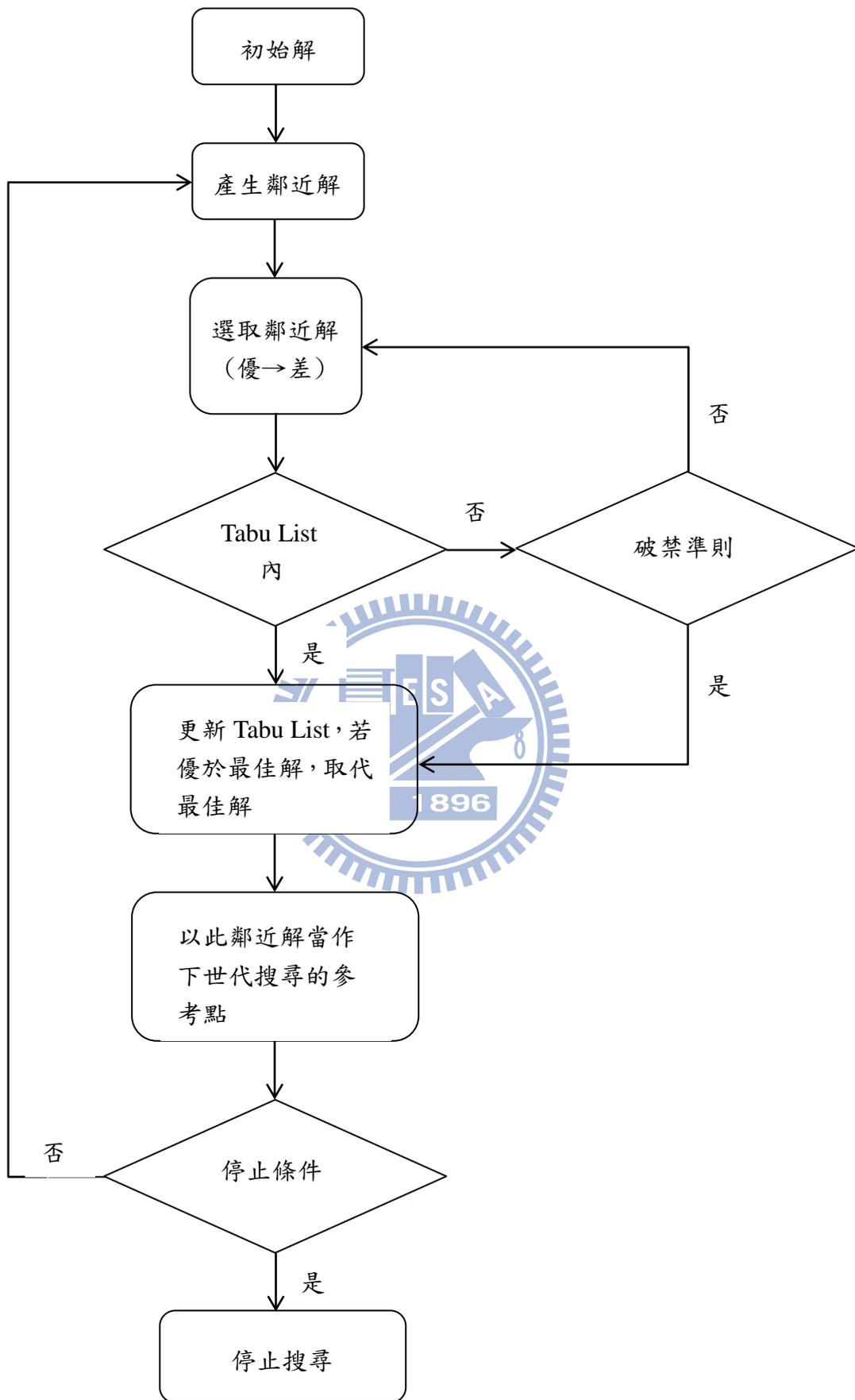


圖 3.13 塔布演算法流程

第四章 實驗情境與結果

本章節為利用塔布搜尋法，模擬實驗情境之結果。利用 c++ 撰寫程式，編譯器為 MSVC9.0，編寫環境為 Visual Studio 2008，使用的作業系統為 Windows 7 Enterprise 32-bits，記憶體為 4G RAM，中央處理器為 AMD Athlon(tm) II X4 640 Processor 3.00 GHz 的電腦。第一節：實驗情境說明；第二節：實驗期望達成的目的；第三節：參數設定實驗；第四節：利用塔布搜尋法之模擬結果分析；第五節：統計結果。

4.1 實驗目的

依據 Wu *et al.* (2011) 提出，在固定序列流線型製造單元系統且具順序相依家族整備時間下，相同基因演算流程，新表達法比舊表達法有較好的表現。因此本論文想驗證，在相同的排程背景下，以相同塔布搜尋流程，新表達法的表現是否還是優於舊表達法。比較基準為比較目標值 makespan 優劣。

4.2 實驗情境

本研究之實驗情境依據 Schaller *et al.* (2000) 的實驗情境，利用舊表達法與新表達法，透過相同的塔布搜尋流程模擬實驗情境，求解 makespan。其情境設定如下：

(1) 三種整備時間：

Small setups(SSU) : $U[1, 10]$

Medium setups(MSU) : $U[1, 50]$

Large setups(LSU) : $U[1, 100]$

(2) 每種整備時間各有 10 種情境

(工件族數，機台數) : $\{(3, 3), (3, 4), (4, 4), (5, 5),$

$(5, 6), (6, 5), (6, 6), (8, 8), (10, 8), (10, 10)\}$ 。

因此總共有 30 種不同的情境。

(3) 每種情境有 30 個 instances

每一種 instance 即不同工件數、不同整備時間與不同加工時間。

(4) 每個 instance 產生 15 筆數，即 15 個不同的亂數起始點

(5) 工件處理時間： $U[1, 10]$

(6) 工件族內的工件數： $U[2, 10]$

本實驗有 3 種整備時間，每種整備時間有 10 種情境，每種情境 30 個 instances，每個 instance 產生 15 筆數據，利用兩種不同表達法，求解 makespan，所以本實驗次數總共 27000 次，即 $3 \times 10 \times 30 \times 15 \times 2 \times 1 = 27000$ 。

4.3 前測實驗

此小節將詳細說明終止條件為什麼要取 2000 代的設定，新表達法與舊表達法初始解為什麼要相同的數據實驗佐證。



4.3.1 初始解設定

在過往使用 Meta-heuristic 求解的演算法，其著重的地方在於演算法的進化機制比較，因此只要以相同的 random number 做為起點，其初始解皆會為同一組初始解。然而本論文著重點並不是在進化機制的比較，是著重於不同的解表達法比較，因此以相同的 random number 產生初始解，兩種不同的表達法其初始解並不相同。故為了瞭解初始解對實驗結果是否會有影響，我們進行此前測實驗來驗證，確保實驗結果的完整性。本研究測試了新舊表達法以不同初始解開始搜尋和新舊表達法以相同初始解開始搜尋，其最後的結果。實驗 30 個情境，每種情境各跑一個 instance，每個 instance 跑出 15 筆數據，終止條件先假設最佳解連續 2000 代不變。

我們以 Paired t-test 為統計檢定方式，比較方式為新表達法與新表達法相比，

舊表達法與舊表達法相比，也就是以不同初始解作為搜尋起點的新舊表達法結果與以相同初始解作為搜尋起點的新舊表達法結果相比。檢定前需先將資料進一步正規化，以至於我們可以做整體性的檢定。假設 $S : \{SSU33, SSU34, \dots, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 代表所有 30 個情境的集合。 $u_{s,i,1}$ 代表在情境 s 下，以不同初始解為搜尋起點的新舊表達法目標值平均，而 $u_{s,i,2}$ 代表在情境 s 下，以相同初始解為搜尋起點的新舊表達法目標值平均，其中 $s \in S, i = 1, 2$ (1 代表新表達法，2 代表舊表達法)。計算公式如下所示。

在情境 s 下新舊表達法的正規化樣本資料

$$d_{s,i}\% = (u_{s,i,1} - u_{s,i,2}) * 100 / u_{s,i,1} \quad (2)$$

新舊表達法在 30 個 instance 的樣本總平均：

$$\bar{d}_i\% = \frac{\sum_{s \in S} d_{s,i}}{30} \quad (3)$$

新舊表達法在 30 個 instance 的樣本標準差：

$$SD_i = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_i)^2}{29}} \quad (4)$$

計算 t_0

$$t_0 = \frac{\bar{d}_i}{SD_i / \sqrt{30}} \quad (5)$$

數據結果如表 4.1 所示，在 95% 的信心水準下，以相同初始解開始求解或以不同初始解開始求解，其新舊表達法的統計結果皆無顯著差異。因此未來研究以不同表達法的議題，其初始解的設定，以相同初始解為起點或以不同初始解為起點皆可以。

雖然統計結果並無顯著差異，但若以單一情境來看，其實不同的初始解還是會有所差異，如表 4.2 所示，以 LSU1010 為例。同 seed 代表新舊表達法初始解不相同所計算的目標值，同 solution 代表新舊表達法初始解相同所算的目標值。首先以新表達法來看，由於皆是由隨機方式產生且相同的 random number，故數值會相同。但對舊表達法而言，可以很明顯的看出，在相同結束條件與相同的

random number 下，15 筆目標值數據有些會變好有些會變差，因此初始解的不同對於結果其實是會有所影響。因此針對此問題，雖然統計結果無顯著差異，但本論文還是採取新舊表達法初始解相同。新表達法以隨機方式產生，舊表達法由新表達法轉換而得。

表 4.1 初始解比較

| 相同初始解 VS 不同初始解 | | | | |
|----------------|----------|-------|---------------------|--------|
| Scenario | 平均改善率(%) | t_0 | $t_{(\alpha, n-1)}$ | 統計結果 |
| 舊解表達法 | 0.01 | 0.06 | 2.05 | 沒有顯著差異 |
| 新解表達法 | 0.00 | 0.00 | 2.05 | 沒有顯著差異 |

表 4.2 初始解比較-單一情境(LSU1010)

| 結束條件 | 2000 | | | | | |
|------|--------|------------|-------|--------|------------|------|
| | 舊表達法 | | | 新表達法 | | |
| 表達法 | 同 seed | 同 solution | 差 | 同 seed | 同 solution | 差 |
| 1 | 971 | 931 | 40 | 893 | 893 | 0 |
| 2 | 959 | 979 | -20 | 959 | 959 | 0 |
| 3 | 956 | 912 | 44 | 953 | 953 | 0 |
| 4 | 961 | 978 | -17 | 894 | 894 | 0 |
| 5 | 912 | 951 | -39 | 911 | 911 | 0 |
| 6 | 969 | 936 | 33 | 927 | 927 | 0 |
| 7 | 936 | 972 | -36 | 914 | 914 | 0 |
| 8 | 942 | 943 | -1 | 901 | 901 | 0 |
| 9 | 969 | 973 | -4 | 921 | 921 | 0 |
| 10 | 959 | 989 | -30 | 967 | 967 | 0 |
| 11 | 927 | 960 | -33 | 934 | 934 | 0 |
| 12 | 966 | 982 | -16 | 954 | 954 | 0 |
| 13 | 939 | 965 | -26 | 957 | 957 | 0 |
| 14 | 963 | 936 | 27 | 921 | 921 | 0 |
| 15 | 998 | 956 | 42 | 912 | 912 | 0 |
| 平均 | 955.13 | 957.53 | -2.40 | 927.87 | 927.87 | 0.00 |

4.3.2 終止條件代數設定

對於代數的設定，我們實驗了最佳解連續 2000 代不變與 2500 代不變。實驗情境皆與 4.3.1 初始解設定相同，新舊表達法的初始解為同一組初始解。統計檢定的方式與 4.3.1 一樣，皆以 Paired t-test 為檢定方式，比較方式還是以新表達法相比，舊表達法相比的方式，也就是改成最佳解連續 2000 代不變的新舊表達法目標值與最佳解連續 2500 代不變的新舊表達法目標值相比，因此統計檢定的參數定義幾乎與 4.3.1 的統計檢定參數相同，只須修正 $u_{s,i,1}$ 與 $u_{s,i,2}$ 的定義。 $u_{s,i,1}$ 改成 $u_{s,i,2500}$ ，代表在情境 s 下，最佳解連續 2500 代不變的新舊表達法目標值平均。而 $u_{s,i,2}$ 改成 $u_{s,i,2000}$ ，代表在情境 s 下，最佳解連續 2000 代不變的新舊表達法目標值平均。 $s \in S, i = 1, 2$ (1 代表新表達法, 2 代表舊表達法)。其計算公式如下：
在情境 s 下新舊表達法的正規化樣本資料

$$d_{s,i}\% = (u_{s,i,2500} - u_{s,i,2000}) * 100 / u_{s,i,2500} \quad (6)$$

新舊表達法在 30 個 instance 的樣本總平均：

$$\bar{d}_i\% = \frac{\sum_{s \in S} d_{s,i}}{30} \quad (7)$$

新舊表達法在 30 個 instance 的樣本標準差：

$$SD_i = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_i)^2}{29}} \quad (8)$$

計算 t_0

$$t_0 = \frac{\bar{d}_i}{SD_i / \sqrt{30}} \quad (9)$$

統計結果如表 4.3 所示，結束條件最佳解連續 2000 代不變與最佳解連續 2500 代不變，新舊表達法皆無著差異。因此結速條件只需跑至 2000 代即可停止。

表 4.3 不同結束代數-目標值

| 結束條件(2000 VS 2500) | | | | |
|--------------------|----------|-------|---------------------|--------|
| Scenario | 平均改善率(%) | t_0 | $t_{(\alpha, n-1)}$ | 統計結果 |
| 舊解表達法 | 0.0000 | 0.00 | 2.05 | 沒有顯著差異 |
| 新解表達法 | 0.0243 | 1.67 | 2.05 | 沒有顯著差異 |

4.4 實驗數據

結果將分成兩部分來說明，第一部份為所有情境與不同整備時間情境的平均分析，第二部分為三種不同整備時間各自結果分析。整體實驗結果整理如表 4.4 所示。表格符號說明如下所示：

- N ：每種情境的 instances 數。
- N_e ：每種情境新表達法與舊表達法所計算之目標值相同的 instances 數。
- N_W ：每種情境新表達法所求之目標值贏舊表達法之 instances 數。
- C_n ：每種情境新表達法在 30 個 instances 的平均值。
- C_t ：每種情境舊表達法在 30 個 instances 的平均值。
- $Ave.(\%)$ ：每一種情境，新表達法對於舊表達法的改善率。
- T_n ：新表達法於 30 個 instances 的平均求解時間。
- T_t ：舊表達法於 30 個 instances 的平均求解時間。
- Gap ：新表達法的求解時間與舊表達法的求解時間的差值。



表 4.4 實驗數據結果

| ID | Scenario | makespan | | | | | | | Computation Time(sec) | | |
|--------------------------|----------|-----------|------------------------------------|----------------------|----------------------|----------------------|----------------------|-----------------|-----------------------|----------------------|---------------|
| | | <i>N</i> | <i>N_w+N_e</i> | <i>N_e</i> | <i>N_w</i> | <i>C_n</i> | <i>C_t</i> | <i>Ave. (%)</i> | <i>T_n</i> | <i>T_t</i> | <i>Gap</i> |
| 1 | SSU33 | 30 | 30 | 3 | 27 | 134.48 | 137.27 | 2.04 | 0.94 | 0.33 | -0.61 |
| 2 | SSU34 | 30 | 30 | 1 | 29 | 151.10 | 153.98 | 1.92 | 1.40 | 0.42 | -0.98 |
| 3 | SSU44 | 30 | 30 | 0 | 30 | 186.55 | 191.57 | 2.65 | 2.64 | 0.61 | -2.02 |
| 4 | SSU55 | 30 | 30 | 0 | 30 | 243.25 | 250.31 | 2.81 | 7.01 | 1.05 | -5.96 |
| 5 | SSU56 | 30 | 30 | 0 | 30 | 254.87 | 262.76 | 3.04 | 7.30 | 1.17 | -6.13 |
| 6 | SSU65 | 30 | 30 | 0 | 30 | 288.18 | 296.44 | 2.79 | 12.35 | 1.50 | -10.85 |
| 7 | SSU66 | 30 | 30 | 0 | 30 | 296.83 | 305.63 | 2.88 | 13.43 | 1.66 | -11.77 |
| 8 | SSU88 | 30 | 30 | 0 | 30 | 407.41 | 419.07 | 2.77 | 43.40 | 3.50 | -39.90 |
| 9 | SSU108 | 30 | 30 | 0 | 30 | 489.27 | 504.36 | 2.99 | 95.58 | 5.30 | -90.27 |
| 10 | SSU1010 | 30 | 30 | 0 | 30 | 530.67 | 546.58 | 2.90 | 128.91 | 6.83 | -122.08 |
| 11 | MSU33 | 30 | 30 | 6 | 24 | 160.00 | 162.41 | 1.41 | 0.73 | 0.28 | -0.45 |
| 12 | MSU34 | 30 | 30 | 3 | 27 | 184.71 | 187.60 | 1.53 | 1.09 | 0.38 | -0.71 |
| 13 | MSU44 | 30 | 30 | 0 | 30 | 238.34 | 245.51 | 2.95 | 2.71 | 0.60 | -2.11 |
| 14 | MSU55 | 30 | 30 | 0 | 30 | 309.52 | 318.11 | 2.70 | 6.09 | 1.03 | -5.06 |
| 15 | MSU56 | 30 | 30 | 0 | 30 | 319.91 | 329.14 | 2.85 | 6.02 | 1.08 | -4.95 |
| 16 | MSU65 | 30 | 30 | 0 | 30 | 366.90 | 378.18 | 3.00 | 10.96 | 1.43 | -9.53 |
| 17 | MSU66 | 30 | 30 | 0 | 30 | 385.34 | 398.26 | 3.23 | 12.25 | 1.64 | -10.61 |
| 18 | MSU88 | 30 | 30 | 0 | 30 | 521.90 | 540.04 | 3.36 | 36.51 | 3.30 | -33.21 |
| 19 | MSU108 | 30 | 30 | 0 | 30 | 640.46 | 661.39 | 3.16 | 86.43 | 5.42 | -81.00 |
| 20 | MSU1010 | 30 | 30 | 0 | 30 | 672.82 | 692.26 | 2.80 | 94.50 | 6.12 | -88.38 |
| 21 | LSU33 | 30 | 30 | 4 | 26 | 228.09 | 233.21 | 2.24 | 1.06 | 0.37 | -0.69 |
| 22 | LSU34 | 30 | 30 | 4 | 26 | 239.06 | 243.85 | 2.05 | 0.88 | 0.31 | -0.57 |
| 23 | LSU44 | 30 | 30 | 0 | 30 | 324.89 | 333.18 | 2.62 | 2.41 | 0.55 | -1.86 |
| 24 | LSU55 | 30 | 30 | 0 | 30 | 421.59 | 434.62 | 3.03 | 5.70 | 0.96 | -4.75 |
| 25 | LSU56 | 30 | 30 | 0 | 30 | 442.82 | 455.67 | 2.84 | 6.84 | 1.13 | -5.71 |
| 26 | LSU65 | 30 | 30 | 0 | 30 | 500.06 | 518.11 | 3.48 | 10.63 | 1.47 | -9.16 |
| 27 | LSU66 | 30 | 30 | 0 | 30 | 524.49 | 540.73 | 3.02 | 10.93 | 1.50 | -9.43 |
| 28 | LSU88 | 30 | 30 | 0 | 30 | 722.17 | 745.71 | 3.18 | 33.00 | 3.18 | -29.83 |
| 29 | LSU108 | 30 | 30 | 0 | 30 | 880.50 | 910.78 | 3.31 | 71.70 | 5.02 | -66.68 |
| 30 | LSU1010 | 30 | 30 | 0 | 30 | 935.79 | 968.47 | 3.37 | 85.92 | 6.10 | -79.82 |
| Average of all Scenarios | | 30 | 30.00 | 0.70 | 29.30 | 400.07 | 412.17 | 2.76 | 26.65 | 2.14 | -24.50 |

4.4.1 實驗總平均

表 4.5 所代表的意義為所有情境的總平均，平均 30 個 instances，新表達法所求得的目標值可贏 29.3 個 instances，平手 0.7 個 instances，贏加平手的次數為 30 個 instances，改善率為 2.76%，由此可知，以所有實驗情境來看，新表達法對於舊表達法能夠求得較優的解，求得的目標值不是贏舊表達法就是與舊表達法有相同的目標值，立於不敗之地。在以時間來看，新表達法所需時間較久，舊表達法求解時間較快，但兩者的平均值相差 30 秒以內，差距並不大，是我們可以接受的範圍。

表 4.5 實驗數據總平均

| Average of all Scenarios | | | | | | | | | |
|--------------------------|------------------------------------|----------------------|----------------------|----------------------|----------------------|-----------------|----------------------|----------------------|-----------------|
| <i>N</i> | <i>N_w+N_e</i> | <i>N_e</i> | <i>N_w</i> | <i>C_n</i> | <i>C_t</i> | <i>Ave. (%)</i> | <i>T_n</i> | <i>T_t</i> | <i>Gap(sec)</i> |
| 30.00 | 30.00 | 0.70 | 29.30 | 400.07 | 412.17 | 2.76 | 26.65 | 2.14 | -24.50 |

4.4.2 目標值平手與贏的次數比較

由 4.4.1 實驗結果總平均得知，新表達法有優於舊表達法的現象。對於三種不同整備時間的結果，將透過此小節與 4.4.3 小節檢視不同整備時間是否有不同的結果。由圖 4.1、4.2 與 4.3 所示，不論是哪一種整備時間，機台數與家族數為 (3.3)與(3.4)，新表達法在 30 個 instances 還無法全部皆贏，代表在機台數家族數較小的情境下，舊表達法也可以找到不錯的解，但隨著機台數與家族數越來越多，新表達法的目標值在 30 個 instances 下皆贏舊表達法，代表機台數與家族數越來越大的情境下，新表達法亦有較佳的結果。值得注意的是，三種不同的整備時間，在任何情境下，新表達法目標值贏加平手的次數皆為 30，代表新表達法最差的目標值結果也會與舊表達法一樣，求得的目標值不是贏舊表達法就是打成平手。

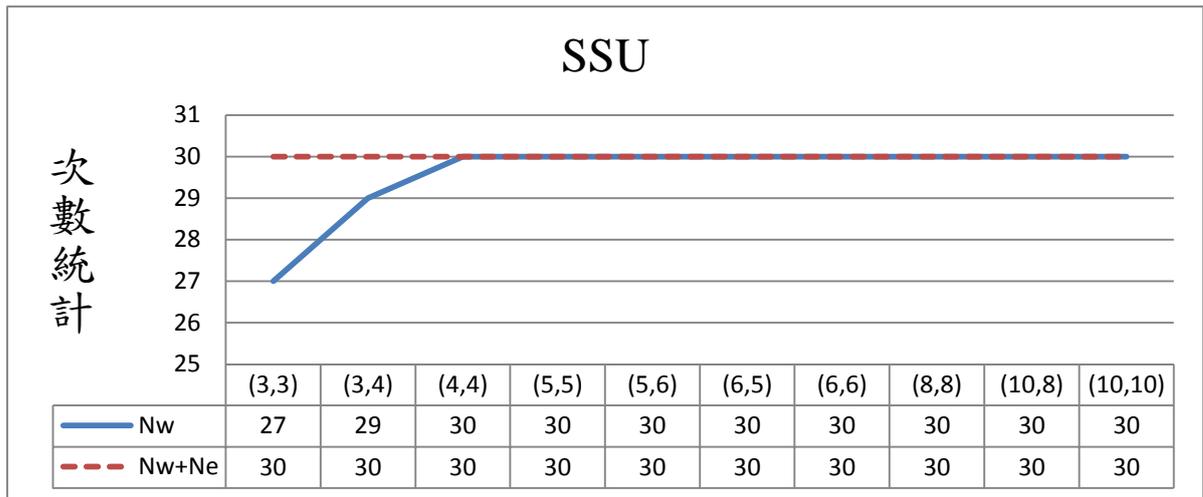


圖 4.1 SSU 下的次數比較

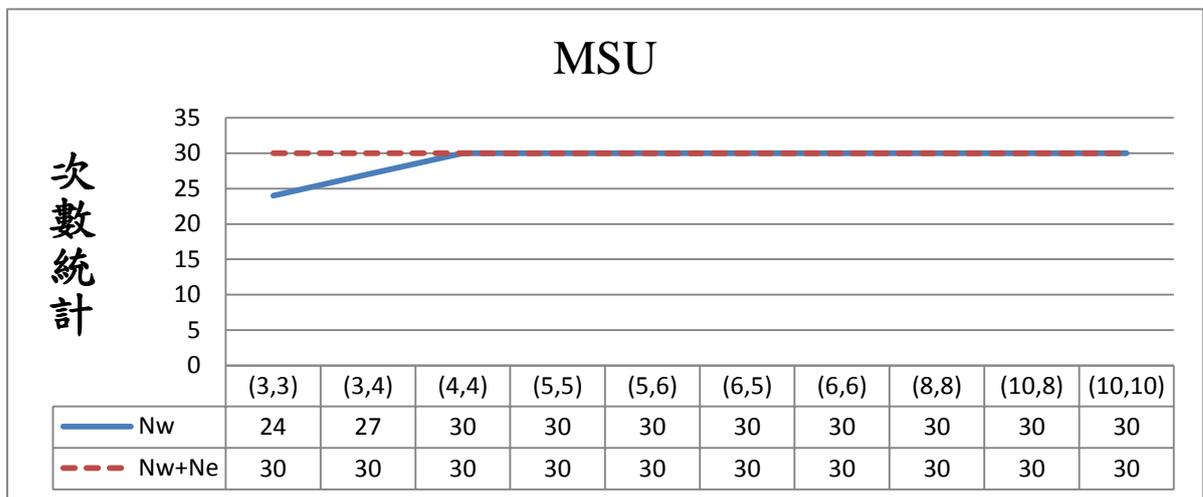


圖 4.2 MSU 下的次數比較

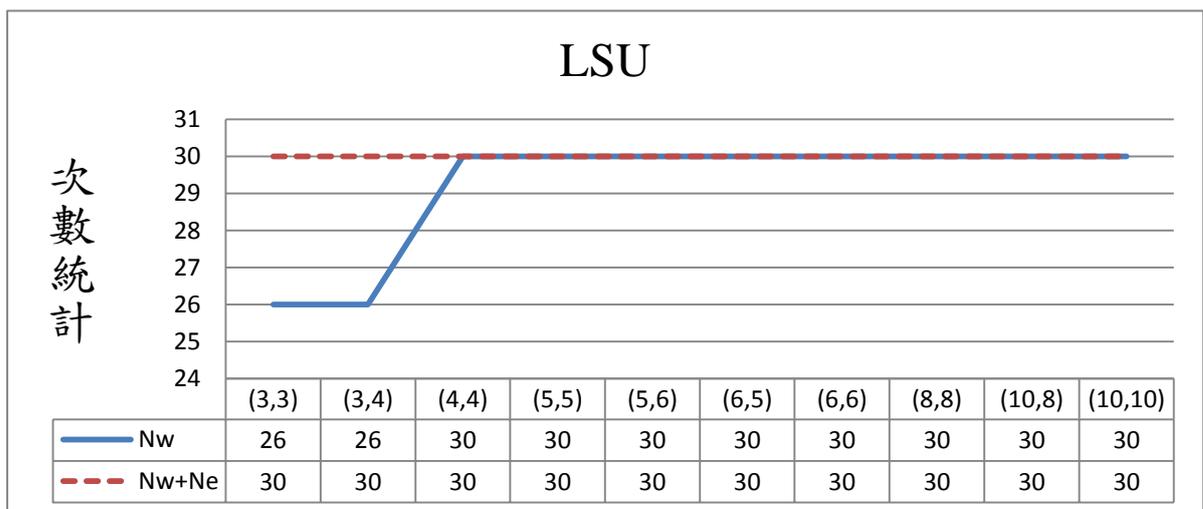


圖 4.3 LSU 下的次數比較

4.4.3 目標值改善率

目標值改善率算法 = $\frac{(\text{舊表達法目標值} - \text{新表達法目標值})}{\text{舊表達法目標值}}$ 。由圖 4.4 得知，對於不同的整備時間，其目標值改善率皆為正值，亦即新表達法能夠比舊表達法找到更好的工件加工順序與目標值。整體而言，機台數與家族數越大的情境，目標值改善率相對於機台數與家族數較小的情境改善得越多。以整備時間 MSU 為例，在最小的情境(3,3)改善率為 1.41，最大情境(10,10)改善率為 2.8，其增加幅度可以達到最小情境的 2 倍。因此在情境越大的情況下，新表達法將有更明顯的效益。所有情境，最大改善率在情境 MSU(8,8)的 3.36%，最小改善率在情境 MSU(3,3)的 1.41%。

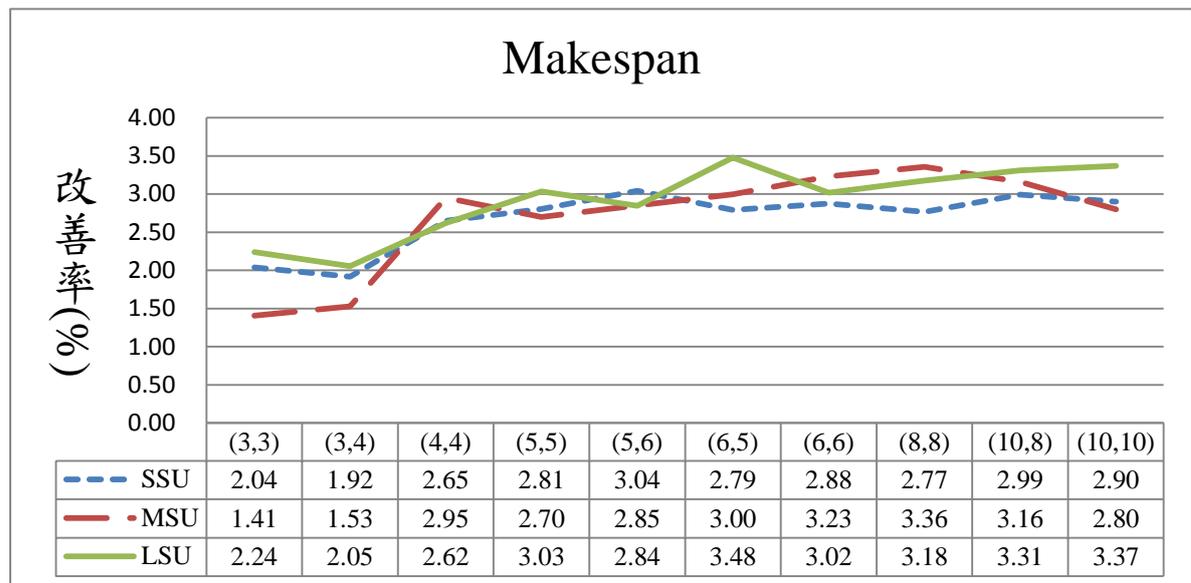


圖 4.4 目標值改善率

4.5 統計檢定

由 4.4 實驗數據結果得知，在任何情境下，新表達法求得的目標值皆優於舊表達法，為了更確認數據結果，因此利用統計檢定方式，確認兩種不同表達法求得的目標值是否有顯著差異。利用 Paired t-test 檢定方法來驗證。檢定四種實驗結果，即 SSU、MSU、LSU 三種整備時間情境及所有情境。檢定的方法為檢定

新表達法相對於舊表達法的表現是否顯著，並以相對改善率為樣本資料。檢定前需先將資料進一步正規化，以至於我們可以做整體性的檢定。假設 $S:\{SSU33, SSU34, \dots, MSU33, MSU34, \dots, LSU33, LSU34, \dots, LSU1010\}$ 代表所有 30 個情境的集合，而 $u_{new,s,i}$ 及 $u_{old,s,i}$ 分別代表在情境 s 下第 i 個實驗新表達法或舊表達法的目標值平均，其中 $s \in S, i = 1, 2, \dots, 30$ (每種情境有 30 個 instances)。

詳細計算資料結果如附錄一所示，計算公式如下所示。

在情境 s 下的第 i 個正規化樣本資料

$$d_{s,i}\% = (u_{old,s,i} - u_{new,s,i}) * 100 / u_{old,s,i} \quad (10)$$

不同整備時間情境及所有情境的樣本平均：

$$\bar{d}_k\% = \frac{\sum_{s \in k} \sum_{i=1}^{30} d_{s,i}}{300}, k \in \{SSU, MSU, LSU\} \quad (12)$$

$$\bar{d}_{Total}\% = \frac{\sum_{s \in S} \sum_{i=1}^{30} d_{s,i}}{900} \quad (13)$$

不同整備時間情境及所有情境的樣本標準差：

$$SD_k = \sqrt{\frac{\sum_{s \in k} (d_{s,i} - \bar{d}_k)^2}{299}}, k \in \{SSU, MSU, LSU\} \quad (14)$$

$$SD_{Total} = \sqrt{\frac{\sum_{s \in S} (d_{s,i} - \bar{d}_{Total})^2}{899}} \quad (15)$$

不同整備時間情境及所有情境檢定用 t 值：

$$t_k = \frac{\bar{d}_k}{SD_k / \sqrt{300}}, k \in \{SSU, MSU, LSU\} \quad (16)$$

$$t_{Total} = \frac{\bar{d}_{Total}}{SD_{Total} / \sqrt{900}} \quad (17)$$

根據公式計算，可得表 4.5。結果可分四個部分來看，LSU、MSU、SSU 三種不同整備時間的各自情境與整體情境。在 95% 信心水準 ($\alpha = 0.05$) 下，四個部份的統計結果皆為顯著贏，代表未經過統計檢定前的實驗結論，新表達法的目標值優於舊表達法，經過統計檢定後，更能確立新表達法贏舊表達法。新表達法贏舊表達法目標值的差距，為有意義的贏。因此不論是數據結果還是統計結果，新

表達法所求得的目標值皆能優於舊表達法。

表 4.6 統計檢定結果

| Scenario | 平均改善率(%) | t_0 | $t_{(\alpha, n-1)}$ | 統計結果 |
|----------|----------|-------|---------------------|------|
| LSU | 2.91 | 24.24 | 1.96 | 顯著贏 |
| MSU | 2.70 | 28.61 | 1.96 | 顯著贏 |
| SSU | 2.68 | 42.19 | 1.96 | 顯著贏 |
| Total | 2.76 | 50.05 | 1.96 | 顯著贏 |



第五章 結論與未來研究方向

本論研究重點在於解的表達方式，以具順序相依家族整備時間與固定序列流線型製造單元系統作為排程背景。兩種不同的表達方式搭配相同的塔布搜尋法流程，其結果有顯著上的差異。本章節將分成兩小節，第一節，針對第四章的實驗結果做總結；第二節，提出未來研究方向。

5.1 結論

過往使用 Meta-heuristic 求解的演算法，主要著重於演算法的進化機制比較，例如，使用相同的基因演算法，比較不同交配方式的績效。除了利用進化機制來改善績效外，其實也可以利用解的表達法來改善。因為一組工件加工順序其實就代表一組解的表達，因此不同的解表達亦可能產生不一樣的結果。在解的表達方式，可以分成新表達法與舊表達法兩種，其形式如同 2.4、3.4 與 3.5 介紹所示。本研究即比較這兩種不同的表達方式，利用相同 Tabu Search 流程求解，並以具順序相依家族整備時間與固定序列的流線型製造單元系統作為排程背景。

實驗此研究問題前，先針對初始解與終止條件作前測實驗，初始解的前測實驗目的即為了瞭解，以相同初始解為起點和不同初始解為起點，其結果是否有顯著差異。利用 Paired t-test 檢定後，其結果並無顯著上的差異，因此以相同初始解為起點和不同初始解為起點皆可以。但我們再以單一情境的 15 筆數據來看，初始解不同，舊表達法所求得的解並不相同，因此為了讓新舊表達法比較基準點一樣，我們還是採用新表達法初始解與舊表達法初始解為同一組初始解。終止條件的前測實驗目的即測試何時停止搜尋結果，測試最佳解連續 2000 代不變與最佳解連續 2500 代不變，最後統計結果並無顯著差異，因此我們採取最佳解連續 2000 代不變時，即停止搜尋。

實驗結果顯示，以三種不同整備時間的情境與整體情境來看，雖然新表達法求解時間較長，但新表達法的目標值皆優於舊表達法的目標值。最後以統計檢定

方式確認結果，其統計結果為顯著贏，代表新表達法在各三種整備時間與整體情境皆顯著的優於舊表達法。因此，經過大量的實驗與統計檢定，在此種排程情境下，以新表達法作為解表達是較佳的解表達方式。

5.2 未來研究方向

到目前的研究為止，本論文研究的議題僅研究固定序列的排程背景，未來可研究在非固定序列下或其他的排程背景，其舊與新表達法的績效結果。對於績效指標，目前只做 makespan 單一績效指標，可增加其它實務上重要的績效指標，驗證新表達法的通用性。針對實驗結果，新表達法在任何情境下皆顯著的優於就表達，因此針對此結論，可以分析這兩種不同的表達方式與 Tabu Search 演算法之間的關係，是否在相同的演算法下，舊表達法是較易陷入區域最佳解內而新表達法較不易陷入區域最佳解的範圍內。也可以思考是否可能是此原因造成新表達法求解時間較長的原因，這幾點都值得未來研究的方向與重點。



參考文獻

英文文獻

- Cheng, T.C.E., Gupta, N.D.J., and Wang, G., 2000. A view of flowshop scheduling research with setup times. *Production and Operations Management*, 9, 262-282.
- Das, S.R., and Canel, C., 2005. An algorithm for scheduling batches of parts in a multi-cell flexible manufacturing system. *International Journal of Production Economics*, 97, 247-262.
- Franca, P.M., Gupta, J.N.D., and Mendes, A.S., 2005. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computer & Industrial Engineering*, 48, 491-506.
- Glover, F., 1989. Tabu search Part I. *ORSA Journal of Computing*, 1, 190-206.
- Glover, F., 1990. Tabu search Part II. *ORSA Journal of Computing*, 2, 4-32.
- Grabowski, J. and Wodecki, M., 2004. A very fast tabu search algorithm for the permutation flowshop problem with makespan criterion. *Computer & Operations Research*, 31, 1891-1909.
- Hendizadeh, S.H., Faramarzi, H., Mansouri, S.A., Gupta, J.N.D., and ElMekawy, T.Y., 2008. Meta-Heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, 111, 593-605.
- Jarboui, B., Ibrahim, S., Siarry, P., and Rebai, A., 2008. A combinatorial partical swarm optimization for solving permutation flowshop problems. *Computer & Industrial Engineering*, 54, 526-538.
- Janiak, A., Kovalyov, M.Y., and Portmann, M.C, 2005. Single machine group scheduling with resource dependent setup and processing times. *European Journal of Operational Research*, 162, 112-121.

- Lin, S.W., Ying, K.C., and Lee, Z.J., 2009. Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence family setup times. *Computer & Operations Research*, 36, 1110-1121.
- Naderi, B. and Ruiz, R., 2010. The distributed permutation flowshop scheduling problem. *Computer & Operations Research*, 37, 754-768.
- Schaller, J.E., Gupta, J.N.D., and Vakharia, A.J., 2000. Scheduling a flowline manufacturing cell with sequence dependent family setup times. *European Journal of Operational Research*, 125, 324-339.
- Tseng, L.Y. and Lin, Y.T., 2009. A hybrid genetic local search algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 198, 84-92.
- Wu, M.C., Tai, P.H., and Chiou, C.W., 2011. A Comparison of Two Chromosome Representation Schemes Used in Solving a Family-Based Scheduling Problem. *to be presented in International Conference of Flexible Automation & Intelligent Manufacturing (FAIM)*, June, 2011, Taiwan.
- Webster, S., and Azizoglu, M., 2001. Dynamic programming algorithms for scheduling parallel machines with family setup times. *Computers & Operations Research*, 28, 127-137.
- Xu, K., Feng, Z., and Jun, K., 2010. A tabu-search algorithm for scheduling jobs with controllable processing times on a single machine to meet due-dates. *Computer & Operations Research*, 37, 1924-1938.
- Zhang, Y., Li, X., and Wang, Q., 2009. Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization. *European Journal of Operational Research*, 196, 869-876.

中文文獻

呂佳玟，「應用基因演算法與家族式派工於傳輸整合步進機在小批量情境下之排程問題」，國立交通大學工業工程與管理學系，碩士論文，民國 98 年。

戴邦豪，「應用混合式染色體表達法漁具順序相依家族整備時間之流線型製造單元排程」，國立交通大學工業工程與管理學系，碩士論文，民國 99 年。

馮正民、邱裕鈞，研究分析方法，1 版，新竹市：建都文化，民國 93 年。



附錄一、各個情境下 $d_{s,i}$ 值計算表格

| $d_{s,i}$ | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 | C17 | C18 | C19 | C20 | C21 | C22 | C23 | C24 | C25 | C26 | C27 | C28 | C29 | C30 |
|-----------|------|------|------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|------|------|------|-------|------|------|------|-------|------|------|------|
| LSU33 | 0.00 | 1.06 | 4.10 | 1.03 | 4.40 | 0.50 | 0.03 | 0.43 | 0.00 | 0.18 | 1.84 | 1.12 | 0.00 | 7.13 | 3.31 | 2.47 | 4.26 | 6.90 | 12.02 | 0.64 | 0.42 | 0.00 | 6.96 | 0.39 | 5.33 | 0.89 | 0.28 | 1.09 | 0.26 | 0.14 |
| LSU34 | 0.10 | 0.20 | 2.09 | 0.07 | 0.39 | 1.36 | 2.32 | 0.60 | 5.76 | 9.27 | 0.86 | 3.42 | 0.38 | 1.78 | 0.02 | 0.00 | 1.15 | 0.84 | 0.11 | 3.73 | 0.00 | 0.20 | 22.84 | 0.29 | 0.35 | 2.30 | 0.00 | 0.07 | 1.09 | 0.00 |
| LSU44 | 1.77 | 1.97 | 7.17 | 2.26 | 3.14 | 3.09 | 2.86 | 2.01 | 4.04 | 2.03 | 2.96 | 7.11 | 0.16 | 0.28 | 2.78 | 5.32 | 1.38 | 2.38 | 0.76 | 2.66 | 0.94 | 2.20 | 3.46 | 3.12 | 0.68 | 5.28 | 2.47 | 2.93 | 1.27 | 0.09 |
| LSU55 | 2.06 | 4.49 | 7.45 | 2.71 | 2.70 | 6.12 | 0.92 | 1.72 | 2.19 | 3.59 | 1.38 | 1.24 | 3.63 | 2.56 | 1.48 | 2.70 | 2.97 | 3.77 | 4.07 | 6.23 | 2.12 | 1.89 | 4.22 | 2.35 | 5.16 | 5.42 | 1.70 | 1.83 | 1.75 | 0.62 |
| LSU56 | 3.52 | 2.67 | 1.23 | 1.38 | 3.39 | 4.54 | 1.44 | 4.66 | 4.50 | 2.76 | 1.70 | 1.59 | 2.59 | 2.74 | 1.94 | 2.12 | 2.58 | 6.57 | 1.49 | 3.13 | 3.44 | 3.34 | 1.90 | 2.59 | 1.74 | 0.87 | 5.09 | 2.05 | 4.10 | 3.68 |
| LSU65 | 5.13 | 3.77 | 0.87 | 1.44 | 3.22 | 3.97 | 4.25 | 2.91 | 2.66 | 2.54 | 4.07 | 3.01 | 3.83 | 6.32 | 1.88 | 6.13 | 1.86 | 2.95 | 4.75 | 1.65 | 3.54 | 2.37 | 3.18 | 3.49 | 7.51 | 4.16 | 3.18 | 2.23 | 4.00 | 3.52 |
| LSU66 | 5.62 | 1.76 | 3.36 | 2.35 | 5.71 | 1.97 | 1.36 | 3.18 | 3.75 | 2.71 | 3.40 | 3.41 | 5.72 | 1.74 | 5.33 | 3.02 | 1.50 | 1.52 | 1.83 | 2.17 | 2.52 | 2.05 | 1.53 | 4.12 | 3.09 | 4.56 | 2.80 | 0.57 | 2.02 | 5.87 |
| LSU88 | 2.43 | 3.92 | 4.33 | 2.98 | 5.05 | 2.86 | 3.26 | 2.78 | 2.35 | 2.29 | 2.76 | 4.03 | 3.20 | 2.57 | 2.51 | 2.22 | 3.60 | 2.51 | 4.66 | 2.38 | 1.22 | 2.87 | 2.21 | 4.38 | 5.47 | 2.60 | 2.68 | 4.07 | 3.84 | 3.26 |
| LSU810 | 2.36 | 2.96 | 3.47 | 2.44 | 3.80 | 3.19 | 4.51 | 3.11 | 5.31 | 4.03 | 3.65 | 2.11 | 2.46 | 3.21 | 3.72 | 4.53 | 2.30 | 3.49 | 3.57 | 2.87 | 3.42 | 3.35 | 2.72 | 3.25 | 3.08 | 3.04 | 4.89 | 2.32 | 2.92 | 3.18 |
| LSU1010 | 3.10 | 5.06 | 3.62 | 1.57 | 4.39 | 3.99 | 2.29 | 3.23 | 3.36 | 3.02 | 3.45 | 3.75 | 2.40 | 3.68 | 2.50 | 3.92 | 3.33 | 2.41 | 4.49 | 3.83 | 3.28 | 2.71 | 4.37 | 3.48 | 3.35 | 2.80 | 2.87 | 4.82 | 2.50 | 3.53 |
| MSU33 | 0.61 | 0.77 | 0.49 | 11.44 | 0.57 | 2.89 | 0.58 | 0.51 | 1.53 | 0.90 | 1.21 | 1.27 | 0.00 | 2.33 | 0.00 | 0.47 | 2.09 | 0.04 | 0.93 | 0.00 | 2.59 | 0.22 | 3.90 | 0.00 | 0.64 | 0.00 | 0.90 | 0.00 | 4.74 | 0.56 |
| MSU34 | 1.86 | 3.02 | 4.58 | 1.29 | 1.74 | 0.00 | 0.68 | 1.38 | 0.64 | 1.78 | 3.18 | 0.00 | 2.52 | 0.00 | 0.94 | 0.90 | 0.73 | 2.12 | 0.45 | 0.32 | 3.36 | 2.17 | 1.26 | 1.98 | 1.70 | 3.88 | 0.94 | 1.53 | 0.12 | 0.75 |
| MSU44 | 1.70 | 3.24 | 0.52 | 9.74 | 0.24 | 2.35 | 3.95 | 2.04 | 1.14 | 7.56 | 6.72 | 3.72 | 2.49 | 3.93 | 1.92 | 1.95 | 0.96 | 3.99 | 0.87 | 2.11 | 2.16 | 4.56 | 1.06 | 0.45 | 0.88 | 3.18 | 10.42 | 0.60 | 3.09 | 0.96 |
| MSU55 | 2.89 | 2.31 | 3.79 | 1.13 | 1.64 | 2.43 | 3.09 | 0.93 | 1.60 | 2.83 | 3.19 | 4.16 | 2.39 | 4.57 | 1.54 | 3.06 | 2.65 | 1.15 | 4.26 | 1.60 | 2.76 | 7.13 | 2.65 | 1.83 | 2.96 | 2.48 | 2.39 | 2.58 | 2.06 | 2.88 |
| MSU56 | 3.17 | 7.44 | 3.92 | 1.81 | 6.93 | 3.64 | 0.64 | 3.83 | 3.78 | 4.80 | 2.87 | 3.69 | 3.38 | 2.19 | 3.02 | 0.91 | 2.69 | 2.71 | 2.26 | 2.67 | 1.80 | 2.23 | 1.79 | 1.64 | 0.66 | 2.81 | 1.88 | 3.75 | 1.14 | 1.51 |
| MSU65 | 4.80 | 1.62 | 2.79 | 2.66 | 2.85 | 2.42 | 0.92 | 5.55 | 3.69 | 4.26 | 2.95 | 2.73 | 3.75 | 3.10 | 2.09 | 1.65 | 3.68 | 2.03 | 2.67 | 3.20 | 2.11 | 2.91 | 3.79 | 4.26 | 2.32 | 2.39 | 4.30 | 4.84 | 1.76 | 1.77 |
| MSU66 | 2.52 | 2.20 | 2.73 | 2.87 | 1.93 | 2.03 | 1.75 | 4.13 | 4.55 | 2.57 | 4.50 | 3.60 | 4.23 | 7.37 | 5.52 | 2.05 | 3.95 | 3.30 | 4.28 | 2.42 | 1.64 | 2.34 | 2.95 | 3.10 | 4.67 | 2.46 | 2.64 | 2.36 | 3.86 | 2.40 |
| MSU88 | 4.15 | 2.27 | 3.18 | 3.19 | 4.43 | 3.88 | 2.48 | 3.43 | 1.67 | 2.63 | 3.16 | 4.84 | 4.03 | 4.15 | 2.78 | 1.72 | 1.67 | 3.97 | 3.64 | 2.15 | 2.46 | 3.79 | 4.86 | 2.21 | 2.26 | 3.99 | 8.82 | 3.72 | 2.65 | 2.53 |
| MSU810 | 3.35 | 2.73 | 2.63 | 3.89 | 1.68 | 2.14 | 3.42 | 2.15 | 3.25 | 2.96 | 4.31 | 4.44 | 2.58 | 3.01 | 2.47 | 2.05 | 2.62 | 4.29 | 2.95 | 1.95 | 3.62 | 3.16 | 4.84 | 3.28 | 3.46 | 4.83 | 3.29 | 3.57 | 2.35 | 3.44 |
| MSU1010 | 3.10 | 2.71 | 3.06 | 2.19 | 3.66 | 3.59 | 3.29 | 2.50 | 1.67 | 3.01 | 2.00 | 2.90 | 2.53 | 2.91 | 2.44 | 2.32 | 1.63 | 3.06 | 3.19 | 1.95 | 2.87 | 2.96 | 3.10 | 2.24 | 2.90 | 3.99 | 4.31 | 2.03 | 3.65 | 2.28 |
| SSU33 | 0.00 | 2.17 | 4.15 | 2.32 | 2.29 | 4.37 | 0.44 | 3.08 | 2.91 | 0.62 | 0.00 | 6.71 | 2.54 | 5.43 | 2.37 | 0.51 | 1.14 | 1.45 | 4.35 | 0.12 | 0.31 | 1.48 | 0.00 | 2.57 | 0.75 | 0.18 | 3.93 | 0.53 | 3.57 | 0.85 |
| SSU34 | 0.31 | 0.48 | 0.96 | 0.29 | 0.74 | 1.80 | 4.55 | 0.11 | 1.22 | 1.72 | 0.00 | 3.01 | 1.66 | 7.01 | 0.44 | 1.09 | 1.66 | 1.40 | 1.26 | 1.17 | 4.22 | 4.50 | 1.24 | 4.43 | 1.86 | 1.57 | 2.33 | 4.37 | 0.99 | 1.16 |
| SSU44 | 5.00 | 2.20 | 2.28 | 2.51 | 2.60 | 3.00 | 3.28 | 2.67 | 1.78 | 3.87 | 1.38 | 2.73 | 1.77 | 1.56 | 3.62 | 2.28 | 2.69 | 1.96 | 4.79 | 1.54 | 1.85 | 3.30 | 2.28 | 2.79 | 1.43 | 2.79 | 3.87 | 4.99 | 0.50 | 2.21 |
| SSU55 | 2.95 | 2.78 | 3.84 | 2.67 | 4.01 | 2.04 | 2.04 | 2.50 | 1.13 | 3.02 | 3.17 | 2.66 | 1.41 | 2.90 | 2.22 | 5.84 | 3.57 | 3.21 | 3.92 | 4.75 | 1.97 | 2.03 | 3.34 | 2.33 | 2.55 | 2.54 | 1.12 | 1.60 | 2.63 | 3.45 |
| SSU56 | 2.62 | 3.67 | 4.76 | 4.03 | 4.13 | 2.24 | 2.86 | 4.24 | 2.85 | 3.07 | 1.82 | 2.56 | 1.59 | 3.13 | 2.59 | 3.17 | 3.65 | 2.92 | 2.23 | 1.61 | 3.46 | 3.20 | 2.96 | 2.59 | 1.37 | 2.47 | 2.09 | 3.75 | 5.95 | 3.62 |
| SSU65 | 2.10 | 2.66 | 3.28 | 1.83 | 1.82 | 2.15 | 3.97 | 2.75 | 2.59 | 2.20 | 2.98 | 3.06 | 3.76 | 3.35 | 2.43 | 2.03 | 3.52 | 2.80 | 2.98 | 2.16 | 2.16 | 3.35 | 3.92 | 2.25 | 1.82 | 3.15 | 4.49 | 2.94 | 2.69 | 2.49 |
| SSU66 | 3.76 | 3.01 | 3.28 | 2.70 | 3.76 | 3.34 | 2.70 | 2.78 | 2.89 | 3.32 | 3.27 | 3.57 | 1.69 | 3.05 | 2.39 | 1.97 | 2.98 | 2.12 | 3.11 | 3.51 | 3.93 | 2.58 | 3.24 | 2.18 | 2.00 | 3.06 | 2.60 | 3.18 | 1.95 | 2.33 |
| SSU88 | 2.57 | 3.34 | 3.03 | 3.10 | 4.53 | 2.76 | 2.43 | 2.06 | 3.52 | 3.33 | 2.24 | 1.22 | 2.55 | 3.53 | 1.91 | 3.15 | 2.77 | 3.06 | 3.74 | 2.35 | 1.56 | 2.95 | 3.13 | 2.05 | 2.74 | 2.17 | 2.29 | 3.32 | 3.04 | 2.62 |
| SSU810 | 3.35 | 2.11 | 3.08 | 2.97 | 3.33 | 2.66 | 3.08 | 3.24 | 2.52 | 2.49 | 3.29 | 2.94 | 3.33 | 2.17 | 2.42 | 3.32 | 4.66 | 3.28 | 2.63 | 3.29 | 3.79 | 3.38 | 2.88 | 3.06 | 2.79 | 2.60 | 3.09 | 2.37 | 2.97 | 2.63 |
| SSU1010 | 3.50 | 2.86 | 2.91 | 3.57 | 2.96 | 3.05 | 3.80 | 3.09 | 3.40 | 2.87 | 2.32 | 2.40 | 2.54 | 2.45 | 2.86 | 2.69 | 2.96 | 2.90 | 2.76 | 2.60 | 3.36 | 2.20 | 3.60 | 1.76 | 2.43 | 2.78 | 3.22 | 2.92 | 2.65 | 3.51 |