

國立成功大學  
製造工程研究所  
碩士論文

考慮工件到達時間與機台限制之二階段成批排  
程問題

Scheduling hybrid flowshops with parallel batch,  
release time, and machine eligibility constraints

研究生：張聿邦

指導教授：楊大和

王逸琳

中華民國九十八年六月

# 考慮工件到達時間與機台限制之二階段成批排程問題

張聿邦

國立成功大學製造工程研究所

指導教授：楊大和、王逸琳 博士

## 摘要

本研究主要探討半導體及電子製造產業中常見的多機台流線式製程 (Flow Shop with Multiple Processors ; FSMP)，求解工件到達時間不同與機台限制之二階段成批排程以最小化總完工時間。其中第一階段為序列加工 (Serial) 而第二階段為成批加工 (Batch) 的平行機台。針對此一困難的排程問題，我們首先建構一混整數規劃模型，再依據機台所能加工的工件數提出減少模型變數之技巧，以在較短時間內求解原模型無法處理之較大規模排程問題。根據測試得知，此技巧仍難以即時處理現實之大規模排程，因此我們依照工件的處理特性來調整工件成批加工順序，提出以傳統的 FIFO 派工法則為基礎的 BFIFO 派工法則；然而，該派工法則雖可快速排程，其排程品質卻可能因機台限制而變差。於是，我們再以該排程結果為基礎，提出數種機制以改善排程品質，其中包括先以 BFIFO 法則決定成批工件，再以混整數規劃模型規劃各成批工件的機台及其順序；或是將所得之排程進行各階段機台內及機台間之工件或批量的安插與互換，以在時限內收斂得更佳之排程，此即 Translocation、Interchange 及 Transposition 改善機制 (簡稱 TIT 改善機制)。在數值測試方面，我們嘗試將本研究提出之數學規劃法、BFIFO 派工法則以及相關的數種改善求解機制與 Yang *et al.* (2004) 所提出的禁忌搜尋法來進行效能與效率的比較以及敏感度分析，以驗證各解法之求解品質與效率。實驗結果顯示，本研究所提的 BFIFO 派工法則結合 TIT 改善機制大都能在較短的時間得到較佳的排程結果，因此本研究建議相關產業可嘗試使用此求解機制來解決其即時性排程需求。

**關鍵字：**多機台流線式製程；排程；派工法則；啟發式方法；禁忌搜尋法；總完工時間

# Scheduling hybrid flowshops with parallel batch, release time, and machine eligibility constraints

Yu-Bang Chang

Institute of Manufacturing Engineering National Cheng Kung University

Advisor : Dr. Taho Yang

Co-advisor : Dr. I-Lin Wang

## Abstract

This paper investigates a difficult scheduling problem on a specialized two-stage hybrid flow shop with multiple processors (FSMP) that often appears in semiconductor manufacturing industry, where the first and second stages process serial and parallel batches, respectively. The objective is to seek job-machine, job-batch, and batch-machine assignments such that makespan is minimized, while considering parallel batch, release time, and machine eligibility constraints. We first propose a mixed integer programming (MIP) formulation for this problem, then give a heuristic to estimate the upper bound on the number of jobs processed by a machine to reduce the size of original MIP. In order to handle real-world large-scale scheduling problems, we propose an efficient dispatching rule called BFIFO that assigns jobs or batches to machines based on first-in-first-out principle. To further improve the quality of the calculated schedule, we propose and test several re-optimization techniques using MIP and local search heuristics involving translocation, interchange and transposition (TIT) between assigned jobs. We compare our algorithms with the tabu search method by Yang *et al.*(2004). Computational experiments indicate our proposed re-optimization techniques are very efficient and effective. In particular, our method can produce good solutions for scheduling up to 160 jobs on 40 machines in both stages within 10 minutes.

**Keywords** : Flow Shop with Multiple Processors; Scheduling; Dispatching; Heuristic Algorithm; Tabu Search; Makespan

## 誌謝

研究所生涯隨著論文的完成，終告結束。回想過去兩年的日子，首先要特別感謝指導教授 楊大和老師與 王逸琳老師，給予我專業的知識及研究過程的悉心指導，並培養學生獨立思考的態度，亦感激老師對我的教誨與鼓勵，使學生在課業研究及做人處事上收穫匪淺，得以順利完成學業，謹此對兩位老師致上最誠摯的感謝與敬意。在定義問題的過程中，也感謝案例公司的同仁協助我了解晶圓廠內運作情形與細節，並提供我暑期實習的機會，讓我在進入職場前，對半導體產業有更深一層的認識。此外亦要感謝中原大學工業工程學系 陳建良博士及興國管理學院科技管理學系 郭宜雍博士於口試期間對本論文提出指正及建議，使本論文更為完善，特此表達最誠摯的敬意。

在研究期間，感謝育安學長、元婷學姊、秋文學姊、俊臣學長及瓊嬉學姊對於研究及課業上的協助，並分享實驗室裡的生活點滴；感謝同窗好友鈺瀚、正誼、韋成、馨儀及佳伶的互相扶持與激勵，讓兩年的研究生涯更為深刻及難忘。以及感謝閔智、沁璇、艾靈及宜嫻等學弟妹們的幫助與關心。此外，還要特別感謝好友荃詠、志偉、俊賢、修鶴、定謙、勝凱、易樞與碩彥，在兩年的研究生活裡，並總是在旁給予我最適時的協助與鼓勵，希望大家未來都能更有成就，祝福你們。

最後，將此論文獻給最摯愛的雙親 張振明先生與 洪夙惠女士，感謝父親在我一度想放棄升學的那年不斷地鼓勵我、開導我，感謝您們提供我無後顧之憂的求學生涯，您們無怨無悔的支持是我唸書最大的原動力，更感謝您們二十多年來的養育之恩，未來必將湧泉以報。還有長輩 洪維謙先生、哥哥宏曄這些年對我的支持與鼓勵，若少了你們的關懷，今日不會有如此完美的結果，感謝你們。畢業在即，未來也將邁入另外一個人生階段，不論遇到任何的困難與挑戰，將以虛心學習及獨立思考的態度來面對，也期望自己能發揮所長來貢獻社會、造福人群。在此祝福所有關心我的師長以及支持我的同學和朋友們平安、幸福。

張聿邦 謹誌

民國九十八年七月于 成大

# 目錄

摘要 .....	i
Abstract.....	ii
誌謝 .....	iii
目錄 .....	iv
圖目錄 .....	vi
表目錄 .....	viii
1. 緒論 .....	1
1.1 研究背景與動機 .....	1
1.2 研究目的 .....	3
1.3 研究流程 .....	3
1.4 論文架構 .....	4
2. 文獻探討 .....	6
2.1 半導體生產排程方法相關文獻 .....	6
2.1.1 數學規劃方法 .....	7
2.1.2 派工法則 .....	8
2.1.3 電腦模擬方法 .....	9
2.1.4 智慧搜尋方法 .....	10
2.1.5 人工智慧方法 .....	11
2.2 多機台流線式製程排程研究 .....	11
2.3 禁忌搜尋法 .....	13
2.4 全域下界法 .....	17
3. 研究方法 .....	19
3.1 問題描述 .....	19

3.2 混整數規劃模型建構 .....	20
3.3 BFIFO 派工法則 .....	27
3.4 BFIFO 派工法則結合混整數規劃模型 .....	30
3.5 BFIFO 派工法則結合 TIT 改善機制 .....	33
3.5.1 Translocation 改善機制 .....	36
3.5.2 Interchange 改善機制 .....	41
3.5.3 Transposition 改善機制 .....	44
3.6 禁忌搜尋法 .....	46
3.6.1 起始解 .....	46
3.6.2 移步 .....	46
3.6.3 免禁準則 .....	48
3.6.4 集中化及分散化 .....	48
3.6.5 禁忌搜尋法參數設定 .....	52
4. 實驗結果分析 .....	56
4.1 作業環境 .....	56
4.2 模式資料設定 .....	56
4.3 衡量指標 .....	57
4.4 實驗分析與討論 .....	59
4.4.1 實驗結果分析 .....	59
4.4.2 敏感度分析 .....	66
4.5 小結 .....	69
5. 結論與建議 .....	71
5.1 結論 .....	71
5.2 未來研究與建議 .....	72
參考文獻 .....	74

## 圖目錄

圖 1.1 研究流程圖 .....	4
圖 2.1 禁忌搜尋法大致流程圖 .....	16
圖 3.1 問題概略示意圖 .....	19
圖 3.2 減少混整數規劃變數之技巧流程圖 .....	25
圖 3.3 減少混整數規劃變數之技巧 (1) .....	26
圖 3.4 減少混整數規劃變數之技巧 (2) .....	26
圖 3.5 減少混整數規劃變數之技巧 (3) .....	27
圖 3.6 BFIFO 派工法則流程圖 .....	29
圖 3.7 BFIFO 派工法則結合混整數規劃模型 .....	32
圖 3.8 BFIFO 派工法則結合 TIT 改善機制流程圖 .....	33
圖 3.9 前工作站不同改善機制之先後順序排列的求解收斂情況 .....	35
圖 3.10 後工作站不同改善機制之先後順序排列的求解收斂情況 .....	36
圖 3.11 後工作站的 Translocation 改善機制 .....	37
圖 3.12 後工作站的 Translocation 改善機制流程圖 .....	38
圖 3.13 前工作站使用後工作站的 Translocation 對求解品質的影響 .....	39
圖 3.14 前工作站的 Translocation 改善機制示意圖 .....	39
圖 3.15 前工作站的 Translocation 改善機制流程圖 .....	40
圖 3.16 前工作站不同 Translocation 改善機制測試 .....	41
圖 3.17 Interchange 改善機制示意圖 .....	42
圖 3.18 Interchange 改善機制流程圖 .....	43
圖 3.19 Transposition 改善機制示意圖 .....	44
圖 3.20 Transposition 改善機制流程圖 .....	45
圖 3.21 鄰近交換法示意圖 .....	47

圖 3.22 兩兩交換法示意圖 .....	47
圖 3.23 禁忌搜尋法之長期記憶 1 (修改自 郭宜雍, 民 94) .....	50
圖 3.24 禁忌搜尋法之長期記憶 2 (修改自 郭宜雍, 民 94) .....	50
圖 3.25 禁忌搜尋法之長期記憶 3 (修改自 郭宜雍, 民 94) .....	51
圖 3.26 禁忌搜尋法之長期記憶 4 (修改自 郭宜雍, 民 94) .....	51
圖 3.27 不同禁忌期限之收斂情況 .....	53
圖 3.28 禁忌搜尋法之流程 (Yang <i>et al.</i> , 2004) .....	54
圖 4.1 BFIFO 結合 TIT 三種改善求解機制與全域下界法之比較.....	59
圖 4.2 九種求解機制之時間 .....	60
圖 4.3 九種求解機制之總完工時間 .....	61
圖 4.4 七種啟發式演算法之求解時間 .....	61
圖 4.5 七種啟發式演算法之總完工時間 .....	62
圖 4.6 BFIFO、BFIFO+MIP 以及 BFIFO+MIPH 之總完工時間.....	63
圖 4.7 BFIFO、BFIFO+TIT 與禁忌搜尋法之求解時間比較.....	63
圖 4.8 BFIFO、BFIFO+TIT 與禁忌搜尋法之總完工時間比較.....	64
圖 4.9 FIFO、BFIFO、BFIFO+TIT 與禁忌搜尋法之求解時間比較 .....	67
圖 4.10 FIFO、BFIFO、BFIFO+TIT 與禁忌搜尋法之總完工時間比較 .....	67



## 表目錄

表 3.1 各機台權重之計算 .....	26
表 4.1 求解時間之數據比較 .....	65
表 4.2 總完工時間之數據比較 .....	65
表 4.3 敏感度分析表 .....	66
表 4.4 求解時間之數據比較（敏感度分析） .....	68
表 4.5 總完工時間之數據比較（敏感度分析） .....	68



# 1. 緒論

二十世紀發明電晶體之後，在 IC 技術方面由積體電路逐級提升至大型積體電路以及超大型積體電路，而半導體工業也跟著快速的發展進步。受到 IC 產品功能多樣化及複雜化發展的影響，半導體製程更趨微縮精密。由於 IC 產品生命週期快速縮短，新產品變化速度加快，導致半導體製程加速演進，機台設備亦因而必須隨之汰舊換新，造成設備投資的資本支出更加擴大，致使許多國際性整合元件製造大廠（Integrated Device Manufacturer, IDM）之經營績效及競爭力逐漸受其製造部門拖累，進而增加委外製造代工比重。在全球半導體業朝向專業分工的發展趨勢下，台灣半導體大廠所擅長之晶圓代工業務因而蒸蒸日上，如何而能有效率地利用現有機台之產能以避免訂單遲交，進而縮短交貨日期，已成為目前半導體產業中非常重要的議題。

## 1.1 研究背景與動機

近十年來，半導體產業（Semiconductor Industry）已成為台灣資本與技術最密集的工業，而國內晶圓代工產業也隨著技術的發展，朝向高階製程為主要生產目標。2000 年起，由於 12 吋晶圓廠的平均生產成本已低於 8 吋晶圓廠，國內許多半導體產業從過去 8 吋晶圓時代走向 12 吋晶圓時代，也促進了半導體產業的蓬勃發展，並成為台灣最重要的製造業之一。電子化產品的應用領域日趨廣泛，台灣的半導體產業同樣面臨著全球性的競爭，除了要有創新的技術外，降低成本與提高系統績效實為半導體產業的當務之急。IC 製造業者除了持續提升製造的技術能力、擴充產能及設備外，更重要的是提高製造品質、縮短生產週期以及提高交期的達成率，進而提高客戶滿意度，以因應越來越激烈的競爭壓力。

近幾年已有多座 12 吋晶圓廠計畫於新竹與台南科學園區興建使用，而半導體設備相當昂貴且設備更新快速，動輒花費數千萬元甚至上億元，無塵室之空間利用更是寸土寸金。半導體製程複雜冗長，交期又短；為了提前切入市場，必須充分利用其設備產能，

因此半導體業者不僅會在建廠前做充分的產能規劃，更會在生產過程中尋找進一步改善效率或產能的方法。舉例來說，投料管制、在製品控制、現場監控及生產排程等等手段，都是為了要使設備達到最大的利用率而採取的措施。簡而言之，最佳的生產方式或產能利用一直是晶圓廠所追求的目標。

在晶圓廠中，由於半導體製程加工程序繁複，製造流程往往多達數百道工作站。由矽晶圓投入開始，經過一連串製程步驟，包括光學顯影、快速高溫製程、化學氣相沉積、離子植入、蝕刻、化學機械研磨等前段製程，如此週而復始的加工過程，使矽晶片表面產生晶粒，並經過不同產品的出貨規格檢驗，透過後段製程的測試、切割、封裝以完成一顆顆的積體電路。

晶圓製造的生產型態屬於多機台流線式製程（Flow Shop with Multiple Processors；FSMP）的型態，表示所有產品都必須通過相同的製程才能產出，亦即產品所經過機台順序是相同的，產品在加工流程中存在著一定的順序關係。因此在排程與管理上有其獨特之處。如何有效地充分利用產能將產品在時間內完成所需製程，並改善作業流程降低生產線上不必要的時間浪費，以縮短總完工時間（Makespan）並達到準時交貨的使命，不僅可為公司創造競爭優勢，亦能帶來良好商譽，創造隱性之利潤，因此已成為管理上的重要課題。

FSMP 之排程問題常見於半導體及電子製造產業中，屬於高複雜性的問題。學術界針對此類問題的求解方式通常先以作業研究領域的數學規劃手法加以模式化，接著求解一個混整數規劃問題。然而，混整數規劃問題極難求解，即使以目前最先進的數學規劃軟體求解仍僅能處理小規模的問題，根本無法滿足短時間內處理現實製程之需求；當工件數及機台數量較多時，求解該數學規劃模式之精確最佳解將耗時甚久，遑論半導體廠中的生產排程必須具備即時性的考量。反之，若是利用傳統的派工法則來求解本問題，雖然能夠快速得到好的解，但其求解品質仍有很大的改善空間。

本研究所探討的兩階段 FSMP 排程問題屬於半導體製程中的一環（譬如去光阻與蝕刻等兩階段製程），在多樣少量的生產環境下，後工作站的機台設計允許處理特性相同

的產品成批（最多兩個工件）加工；在各製程單獨排程與排序的情況下，假使前工作站的排程並未考慮後工作站成批加工的順序，將可能產生以下兩種缺失：

- （1）當工件於前工作站加工完成後，到達後工作站的機台必須耗費相當長的時間等待下一個工件成批加工。
- （2）如果工件於後工作站的機台等待成批時間過長，則直接單一工件進入機台加工，將導致機台產能利用率降低。

因此，如何能夠同時考量兩階段的排程影響以縮短完工時間或提高機台產能利用率，是一件非常重要的排程議題。

## 1.2 研究目的

針對以上數種求解 FSMP 排程的缺點以及現場實際發生的問題，本研究以台灣某座位於台南科學園區內之 12 吋晶圓廠為研究案例，以該晶圓廠之半導體製程中諸如去光阻與蝕刻等常見的兩階段製程之排程問題作為主要的研究對象，提出混整數規劃模型及傳統派工法來做實驗，並發展具有即時性且能有效改善求解品質的啟發式方法，以能在現實應用中可接受的時限內（譬如 10 分鐘）為一個具有 160 個工件、各階段皆有 40 部機台的兩階段 FSMP 問題，排出以總完工時間（Makespan）最小為目標之排程。

## 1.3 研究流程

本研究以半導體製造業中的兩階段 FSMP 排程問題為主體，首先探討目前產業背景，針對現況問題進行分析來定義出問題所在。在界定研究主題及範圍後，依據研究主題進行相關文獻之收集及探討；接著針對問題的特性分別建立本文所提出的混整數規劃模式、派工法則、啟發式演算法以及禁忌搜尋法，並進行實驗結果分析，最後進行結論、建議及探討未來可研究方向。研究流程如圖 1.1 所示。

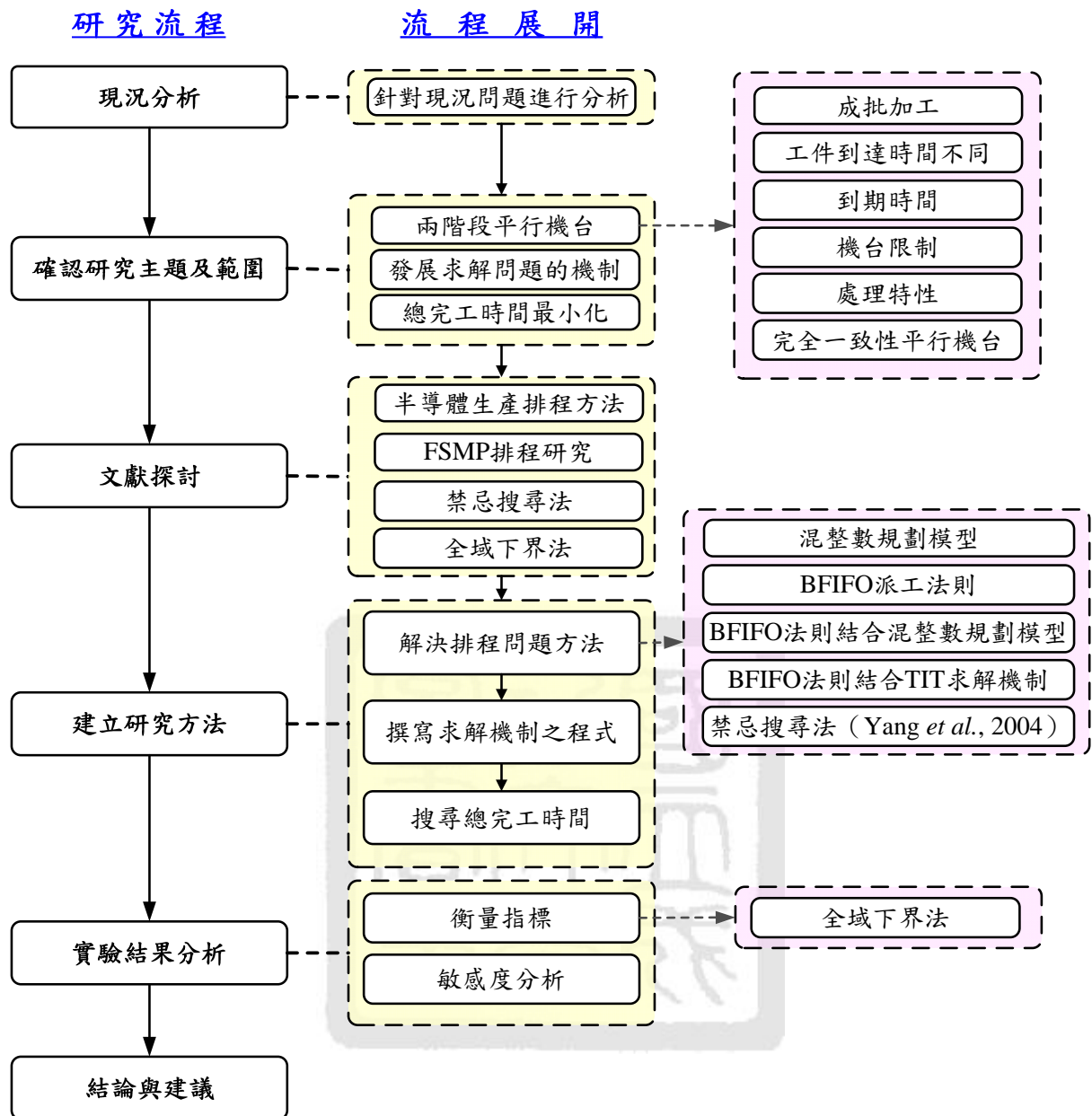


圖1.1 研究流程圖

## 1.4 論文架構

本研究共分為五章，各章節的架構簡述如下：

第一章：主要為確定研究主題，描述研究背景、動機、目的、研究流程以及論文架構。

第二章：進行相關的文獻探討，其目的在於透過文獻探討整理既有的理論與相關文獻，

包括半導體廠生產排程、多機台流線式製程排程、禁忌搜尋法以及全域下界法

等研究。

第三章：本章節為研究方法，首先針對問題進行描述與假設，接著進一步介紹本研究所應用的排程方法，分別為混整數規劃模型、BFIFO 派工法則、BFIFO 派工法則結合混整數規劃模型、BFIFO 派工法則結合 TIT 改善機制以及禁忌搜尋法。

第四章：主要為實驗結果分析，先介紹作業環境以及模式資料設定的參數，接著提出從過去文獻中所發展出的全域下界法來做為衡量指標，接著根據前一章解決排程問題方法所撰寫的程式來進行實驗與分析，來了解所建構的方法在求解效率與求解品質上的績效。

第五章：進行結論與建議，主要針對本研究的實驗結果作為最終的結論以及對未來研究發展方向提出建議，作為後續深入研究之參考。

本研究依照以上各章節的架構簡述，後續將內容詳細的介紹於後。



## 2. 文獻探討

本研究文獻探討主要分為三個部份，首先對半導體生產排程方法相關文獻來做簡介，然後對 FSMP 排程相關的文獻進行探討，並針對本論文使用到的禁忌搜尋法予以探討，接下來再針對過去所提出的全域下界法作文獻的收集。

### 2.1 半導體生產排程方法相關文獻

排程可定義為：在一段時間內，配置可用資源以處理一群工作，來達成所設立的目標 (Pinedo, 1995)。排程問題自 1950 年代以來，已有許多學術文獻發表。由於受到製造系統中的中或長期生產規劃流程所影響，排程必須考量到存貨水準、預測以及資源需求，才能在多種產品及長期的資源分配下最佳化系統的表現。

在排程領域的術語中，首先要先區分排序 (Sequence)、排程 (Schedule) 以及排程法則 (Scheduling Policy) 之間的不同。排序通常用來表示對工件的排列，或者工件在給定一特定機台下加工的順序。而排程是將工件在複雜的機台群組間進行配置。至於排程法則是用在隨機化的設定當中，利用一個法則針對系統中可能產生的狀況，來指定一個適切的行動 (Mosheiov and Oron, 2004)。

晶圓製造需要經過數百道工作站的加工步驟才能完成，每種產品都有其固定的加工程序，不同產品的加工程序也大致相同，故其生產流程可歸納為流程式生產 (Flow Shop)。流程式生產排程是一個常見的排程問題，而晶圓製造除了流程式生產的特性外，尚具有平行機台 (Parallel Machines) 以及機台限制 (Machine Eligibility) 等特性。

半導體廠中的製造流程分成許多階段，每一階段都有許多功能相同的機台，此即為平行機台；機台的加工無論是同產品不同步驟，或不同產品相同加工步驟，其處理特性 (Recipe) 也有不同，各工件將會因其處理特性的不同，在每個階段中只能在某些特定的機台上加工，此即為機台限制。而半導體廠中某幾道工作站機台具有成批加工的形式，成批加工是指一次可同時處理 2 到 6 個工件不等，而成批的工件會同時間

完成加工。

而半導體廠的生產排程問題，於平行機台中找尋求解組合式最佳化屬於 NP-Hard 問題。隨著問題規模的擴大與複雜度的增加，其可行解空間與求解時間都呈現指數或階層式的成長；舉例來說，假使一個工作站擁有 40 部機台，而一個工件可選擇 40 部機台中的任意一部機台來加工，則當產品增加到兩個工件時，其排程的方法將增加為  $40^2$  種可能性；而當產品增加到 160 個工件時，其機台選擇會增加為  $40^{160}$  種可能性，因此可能的排程組合數目將相當地可觀。

根據上述的情況，使用數學規劃方法（Mathematical Programming Approaches）求解的效率並不高，無法於短時間內獲得最佳解，因此近年來的研究為了能夠有效的來解決排程問題，除了數學規劃方法外，各種排程的方法及系統架構相繼被提出，其主要分類為派工法則（Dispatching Rule）、電腦模擬方法（Computer Simulation Approaches）、智慧搜尋方法（Intelligent Search Approaches）以及人工智慧方法（Artificial Intelligence Approaches）等，以下將個別說明之。

### 2.1.1 數學規劃方法

數學規劃方法將根據系統的限制條件和目標，利用作業研究的方法，求得最佳解。但利用此方法解決問題時，會有求解時間過長的情況發生。即使以目前最先進的數學規劃軟體求解，仍僅能處理小規模的問題，根本無法滿足短時間內處理現實製程之需求。若要在短時間內求得系統的最佳解，則需要結合不同的最佳化問題解決方法。

在數學規劃方法的文獻中，Dobson and Nambimadom（2001）提出單一機台並考慮成批的整數規劃模型，並證明此問題為 NP-hard 的問題。Hung（1998）在考慮成批的一階段平行機台，提出動態規劃法解決最小總延遲時間。Van der Velde（1993）以及 Horowitz and Sahni（1976）分別針對平行機台排程提出準確的演算法（Exact Algorithm），但是他們所提出的演算法在處理超過 5 台機台以及 50 個工件的排程時，其計算時間將會過長，導致無法及時提供一組最佳的排程解。Uzsoy（1995）針對



$Pm|B_j|C_{max}$  類排程問題，提出準確與近似最佳解的演算法。儘管數學規劃方法在問題夠小的情況下可能可以在時限內找到正確的最佳解，然而實際的排程問題通常規模皆不小，因此在實務上不可倚賴數學規劃方法在時限內求最佳解，但該法或可提供一個好的初始解。

## 2.1.2 派工法則

派工法則較適合於簡單的派工問題（如單機或多機單階段的情況），通常會針對工件或者資源的特性（如工件的到期日、處理時間、搬運時間、資源的負荷值等）來計算各作業的優先順序值，然後再根據此優先順序值大小，指派優先值順序最高的工件。由於派工法則使用數種貪婪（greedy）準則來進行工件的排序，其優點在於容易建構且較適用於動態問題，但對較複雜的排程問題則無法保證可求得最佳解。常見的派工法則大概有以下四種：最早到期日（Earliest Due Date；EDD）、最長加工時間法（Longest Processing Time First；LPT）、最短加工時間法（Shortest Processing Time First；SPT）、先進先出法（First In First Out；FIFO）。

在簡單派工法則的文獻中，Centeno and Armacost（1997）提到在此之前沒有關於  $Pm|r_j, M_j|L_{max}$  的研究，並將以最小彈性工件（Least Flexible Job；LFJ）、最小彈性機台（Least Flexible Machine；LFM）優先派工的結果與實際的資料作比對，發現有明顯的改善；Centeno and Armacost（2004）提到對 LPT、LFJ、LFM 與三者組合的派工法則進行比較，發現 LPT 的表現最佳。Chen and Vestjens（1997）證明 LPT 解在最壞情況下為最佳解的 3/2 倍。

在過去的文獻中，早期 Johnson（1954）針對雙機排程問題提出兩階段生產排程之啟發式解法（通稱為 Johnson's Rule），以求得總完工時間之最小值。然而，當機台數目大於 3 或是工作站數目大於 3 時，以 Johnson's Rule 所求得之解不一定是最佳解。Mathirajan and Sivakumar（2002）針對  $Pm|r_j, B_j|U_{max}$  類問題，提出一貪婪式派工法則來求解機台利用率最大化。Gupta *et al.*（2004）探討  $Pm|r_j, d_i|C_{max}$  類問題，提出兩種貪

婪式派工法則以解出最小化總完工時間之最佳化排程，並和既有的演算法進行經驗法則之效率和效能的比較。

### 2.1.3 電腦模擬方法

隨著電腦技術及設備規格的提昇，利用電腦來描述一個製造系統可能發生的狀況已變成一種實用的決策輔助方式。電腦在描述製造系統時，除了製造程序本身之外，也可以描述資源的限制以及每個決策點的策略。

由於真實的製造環境過於複雜，不容易透過理論模式以詳盡敘述系統特徵；反之，電腦模擬能夠利用高階的模擬語言來表現一個非常真實的系統，其優點在於能夠提供較為真實的介面給予專家參考，而缺點在於模擬結果並不能提供排程的決策，同時亦無法保證模式能夠求得最佳的排程結果。模擬所產生的結果僅能讓決策者知道所輸入的排程策略的表現狀況，且建模和模擬測試時間過長，間接形成模擬普遍化的障礙。

從實務角度而言，模擬是利用電腦模組化的邏輯設計來描述真實系統的運作，其目的是在於建立一個實驗平台，讓決策者可以快速了解在設定一系列的狀況下其系統可能產生之對應行為。以下使用三種分類來描述模擬的特性 (Kelton *et al.*, 2007)：

- 靜態或動態 (Static vs. Dynamic)：以時間的變動是否為主要影響系統的因子做為區分，若時間變動會造成變數改變的系統，則稱之為動態模擬系統，反之則為靜態模擬系統。
- 連續或離散 (Continuous vs. Discrete)：在連續性的模型中，系統狀態會隨著時間而持續地變化，如水庫的水位變化。而離散性的模型則是由事件來驅動系統狀態的改變，如物件的到達時間、加工時間及離開時間皆為特定時間。如果系統混合了連續和離散的模擬特性，則稱之為混合連續及離散性模擬模型 (Mixed Continuous-Discrete Models)。
- 確定性或隨機性 (Deterministic vs. Stochastic)：若沒有隨機因子則稱之為確

定性模型，如固定服務時間。反之，若模型有隨機因子的輸入，則此模型稱之為隨機性模型，如顧客隨機到達銀行且服務時間亦為隨機。

## 2.1.4 智慧搜尋方法

智慧搜尋方法與貪婪式派工法則的主要差別在於：(1) 貪婪式派工法則只在目標值有改善時才進行交換，常會陷入局部最佳解，而無法進一步找到更好的解；(2) 智慧搜尋法屬於巨集啟發式方法 (Meta-heuristics)，其主要以派工法則為基礎，並根據一些高階的搜尋策略指導下層的啟發式方法，以避免陷入局部最佳解。

在文獻上許多的智慧搜尋法中，最常被用來解決排程問題的為基因演算法 (Genetic Algorithm, GA)、模擬退火 (Simulated Annealing, SA) 法以及禁忌搜尋 (Tabu Search, TS) 法等三種，此三種搜尋法在適當的結合到排程問題之後，往往有不錯的表現。GA 在參數組合的問題上有其應用性及穩健性；SA 有能力可以跳脫局部最佳解，並使得最終解的品質不易受初始解的影響；而 TS 在排序的問題上亦有其應用性及穩健性。

在智慧搜尋方法的文獻中，Monch *et al.* (2005) 考慮  $Pm|r_j, B_j, incompatible| \sum wT$  下，利用 GA 提出兩種方法：(1) 先成批，利用 GA 來決定批量於哪部機台，最後再對批量作排序；(2) 利用 GA 先決定工件於哪部機台，再將各機台的工件成批，最後再對批量作排序。Jou (2005) 針對瓶頸站使用平行機台之問題，應用 GA 來找出最佳化排程，來達到製造成本最小化之績效目標。Gupta *et al.* (2005) 使用 SA 來求解完全一致性平行機台，文中表示搜尋鄰近解的個數會隨著工件數目及機台數目而增加。Malve and Uzsoy (2007) 提出一 GA 來求解  $Pm|r_j, B_j, incompatible|L_{max}$  的問題，並與四種派工法則作比較。Kashan *et al.* (2008) 考慮  $1|B_j|C_{max}$  問題，提出啟發式基因演算法，來求得總完工時間最小化之排程，並與 Melouk *et al.* (2004) 針對單一機台提出的 SA 來做比較。

由於本研究將以 Yang *et al.* (2004) 所提出的禁忌搜尋法，來進行效能與效率的比較以及敏感度分析，因此關於 TS 的原理及流程將於本章 2.3 節作更詳細的描述。

### 2.1.5 人工智慧方法

人工智慧方法對於複雜和類似的系統，必須先利用歷史資料的訓練和專家知識，藉由人類的經驗判斷與解題技巧，作有效率的搜尋而得到良好的績效。因此，執行之前的樣本訓練以及專家知識的取得，皆對人工智慧方法有重要的影響。

一般以人工智慧解決排程問題的方法有專家系統 (Expert System) 以及類神經網路 (Neural Network) 兩種：專家系統主要是將排程專家在排程上的經驗表達成知識，然後再將這些知識以邏輯的方式表達並描述於知識庫之中。當產生排程問題時，則於知識庫中尋找適當的邏輯來進行排程。類神經網路主要是模仿人腦學習的過程，假設類神經像是一個黑盒子 (Black Box) 一樣，給予一個特定的輸入 (排程問題)，這個黑盒子將會產生一組輸出 (排程結果)，然而類神經網路在能夠產生輸出之前，首先必須經過一個學習的過程，利用以往排程的結果作為訓練的資料，該訓練資料包含過去的排程決策以及其對應的排程結果。經由學習之後，該網路即能針對特定的輸入產生相對應的輸出。Min and Yih (2003) 提出以模擬模式結合類神經網路的方法流程，應用於半導體產業的派工決策並解決多目標問題。

## 2.2 多機台流線式製程排程研究

多機台流線式製程 (Flow Shop with Multiple Processors, FSMP) 是指在一流線式 (Flow Shop) 的生產線中，每個工作站皆包含一台 (含) 以上的機台。雖然在同一工作站的機台不盡相同，但其功能上卻相同。此種類型的生產環境相當普遍並且有不同的應用，包括半導體產業、電子製造業以及石化之相關產業等皆是屬於此一類型的製造生產系統。多機台流線式製程之排程屬於複雜性高的問題，且 FSMP 之生產型態與傳統的單

機台流線式製程 (Single Processor Flow Shop) 相似，差異在於 FSMP 中除了具備流線式製程的性質外，還具有平行機台的特性，因此在問題的複雜度上更甚於單機台流線式製程。

有關利用數學規劃方法來解決 FSMP 排程問題的文獻方面，Santos (1996) 利用數學模式建構 FSMP 之製造系統，並用來比較五種排序演算法的優劣。而該文獻中實驗的 FSMP 製造系統包含 2 到 5 個工作站，每個工作站包含 2 到 4 部平行機台。Botta-Genoulaz (2000) 也利用數學模式建構 FSMP 之製造系統，以比較四種排序演算法以及兩種排程演算法，並在包含 5 個工作站、每個工作站各 3 部平行機台的 FSMP 製造系統中進行測試。

針對 FSMP 派工問題的相關文獻，大致上可分為兩類：(1) 利用單一屬性來作為派工的依據，稱為單一屬性派工法則，如最長處理時間 (Longest Processing Time, LPT)、先進先出 (First In First Out, FIFO) 等等。(2) 考慮兩種以上的屬性來作為派工的依據，則稱為多重屬性派工法則，例如關鍵比率 (Critical Ratio, CR) 等等。

在派工法則應用於 FSMP 的相關文獻中，Hunsucker and Shan (1994) 利用模擬建構 FSMP 環境，藉此來比較六種派工法則的表現；Uzsoy *et al.* (1992) 以半導體測試廠為研究對象，比較八種派工法則的優劣；Grangeon *et al.* (1999) 利用物件導向建構 FSMP 之模擬模式，並從十三種的生產情境中比較四種派工法則的表現；Petroni and Rizzi (2002) 使用模糊理論為基礎的方法，來評估五種派工法則在 FSMP 生產系統中的表現。在利用多重屬性派工法則的文獻方面，Kim *et al.* (1998) 將半導體晶圓的製程分為光罩區與非光罩區，兩區分別包含許多工作站，並探討兩邊分別使用不同派工法則對於系統績效的表現；Jayamohan and Rajendran (2000) 模擬三階段的 FSMP 生產系統，藉此模式比較十種派工策略的優劣，其中八種策略為所有階段皆使用相同的多屬性派工法則，其餘兩種策略則是使用派工組合的方法；Kuo *et al.* (2007) 利用類神經網路的模擬 Metamodel，用以簡化派工問題的複雜模擬模式。

在啟發式方法應用於 FSMP 的相關文獻中，Kyparism and Koulamas (2005) 同

樣探討兩階段之 FSMP 問題，並利用 Soewandi and Elmaghraby (2003) 所研究之 S-E 演算法，來找出總完工時間最小化之下界。Haouari and M'Hallah (1997)、Gupta *et al.* (2002) 以及 Allaoui and Artiba (2006) 亦探討類似的議題。目前與本研究最相關的文獻為 Bellanger and Oulamara (2009) 所提出的論文，該文獻考慮處理特性且第二階段為成批加工機台的兩階段 FSMP 問題，並提到過去流線式生產的成批加工機台並沒有同時考慮工件處理特性的文獻，因此該問題屬於典型的流線式生產問題之延伸。Oulamara *et al.* (2009) 針對兩階段各單一機台且第二階段為成批加工機台的問題提出啟發式方法求解。Oulamara (2007) 亦針對兩階段各單一機台且第一階段為成批加工機台的問題提出啟發式方法來求解總完工時間最小化。但 Bellanger and Oulamara (2009) 所提出的問題，其所有的工件為同時到達，而本研究是考慮工件到達時間不同的影響，因此本研究所提出的問題更為複雜。

## 2.3 禁忌搜尋法

由於本研究將以 Yang *et al.* (2004) 所提出的 TS 作效能與效率的測試，因此關於 TS 的原理及流程在本章節作詳細的介紹。禁忌搜尋法是由 Glover (1989) 所提出的一種啟發式搜尋法。此方法用於解決組合最佳化問題，與傳統區域搜尋法差別在於可利用本身機制來跳脫區域最佳解 (Local Optimum) 而達到整體最佳解 (Global Optimum)，其主要的過程首先是利用初始化的方法產生一起始解 (Initial Solution)，再藉由移步 (Move) 的方式將搜尋範圍拓展到其鄰域解 (Neighbor Solution)，經由評估 (Evaluation) 之後若發現鄰域解的表現比現目前解更好，則接受此鄰域解並取代成為現在的解；反之，則繼續尋找；如此重複執行直到滿足終止條件即停止，並以搜尋過程中最佳的解作為搜尋的結果。為了避免尋找最佳解的過程中，倒退到前一個解或重複路徑形成迴圈 (cycle)，最近幾次成功的移步都會以短期記憶的方式被記錄在一禁忌名單中 (Tabu List)，並禁止這些移步在禁忌期限 (Tabu Tenure) 再次發生，直到禁忌期限過後才會將該移步自禁忌名單中移除。倘若禁忌名單中的某移步滿足了免禁準

則 (Aspiration Criterion)，亦即該移步所產生的鄰域解比目前解更佳，則該移步會被接受。

較為先進的禁忌搜尋法除了短期的記憶以外，還包含長期的記憶。其主要功能在於當搜尋過程中當所有的鄰域解皆無法改善且包含禁忌名單中的移步也無法滿足免禁準則時，可藉由集中化 (Intensification) 及分散化 (Diversification) 的方式來強化搜尋過的解空間或分散搜尋的空間。以下針對禁忌搜尋法的基本概念提出說明：

### 起始解

在搜尋法開始執行前皆需要先產生一組解，然後再以此起始解作為初始值進而找尋其他的解，通常可用隨機的方式產生；然而有時也可根據特殊的問題型態，利用特別的方法來產生起始解。

### 移步

在禁忌搜尋過程中，依鄰域解的定義方式，每一個解都會存在著相對應的鄰域解，而將目前的解轉換為其中一個鄰近解的步驟，稱之為移步。常見的方法有鄰近交換法 (Adjacent Exchange)、兩兩交換法 (Pairwise Exchange)、插入法 (Insertion) 以及群插入法 (Block Insertion)。鄰近交換法為從排序中隨機選取一個位置  $i$ ，並將  $i$  及  $i+1$  交換位置，其中  $i$  小於排序的長度。兩兩交換法為隨機選取排序中的兩個位置  $i$  及  $j$ ，並將  $i$  及  $j$  交換位置。插入法旨在隨機選取排序中的兩個位置  $i$  及  $j$ ，並將  $j$  插入  $i$  的位置。群插入法為隨機選取排序中的兩個位置  $i$  及  $j$  以及一個隨機長度  $k$ ，再從  $i$  起算後  $k$  個排序插入到  $j$  位置，其中  $k$  小於排序的長度。

### 鄰域解

所有鄰近之可行解稱為鄰域解，亦即透過移步法則將現行解轉移至另一解之所有可能解。

### 禁忌名單

禁忌名單是利用短期記憶的方式來記錄過去一段時間搜尋中所選移步之屬性。於

禁忌名單中之屬性在一定迭代次數內不能被選取，直到該屬性被釋出禁忌名單後才能再被選取，此方式主要是用來避免鄰域解的重複搜尋。一般而言，禁忌名單愈大，則陷入區域最佳解的機率將越低，但是所需的記憶體空間也愈大，且電腦所需的求解時間也愈久。

### 禁忌期限

每當有新的移步發生時，則所有被記錄在禁忌名單中的移步將會被更新。而更新手法則是採先進先出（FIFO）的方式，亦即下一個新的移步進來，且經過一段禁忌期限之後，就把禁忌名單中最舊的移步從名單中移除。由於名單只固定存放最近幾次的記錄，因此禁忌名單中的移步會一直變動。

### 免禁準則

在搜尋的過程中，當所有的移步都無法再找到更佳的鄰域解時，則會開始搜尋禁忌名單中的移步。若從禁忌名單中的移步發現更佳的鄰域解時，將以此鄰域解來取代現有的解而成為新的解，此種解除禁忌名單的機制，即稱為免禁準則。

### 集中化

在禁忌搜尋法進行的過程中，除了成功的移步會被記錄於短期記憶外，所有成功移步所產生的解也會被記錄在長期記憶中。當搜尋經過一段期間以後，搜尋的結果通常會進入收斂的情況，尤其是當所有的鄰域解以及在免禁準則的條件中所搜尋的解均無法有更好的結果時，便可藉由長期記憶來尋找新的解；而集中化即為利用此長期記憶之記錄來產生一組新解，藉由此方式將搜尋帶入新的區域以求得更佳的解，並繼續進行禁忌搜尋法。

### 分散化

分散化執行的條件與集中化相同，雖然同樣利用長期記憶來尋找新的解，但其方法與集中化不同。分散化是盡量避開過去搜尋過的空間來尋找新的解，此原理在於將搜尋的方向分散，以增加搜尋範圍，避免在相同的空間中持續搜尋。



通常禁忌搜尋法會從分散化或集中化的長期記憶處理方式擇一使用，然而文獻中亦有學者（Ben-Daya and Al-Fawzan, 1998）將此兩種方式加以結合而同時使用，本論文第三章所使用之禁忌搜尋法（詳見 3.6 節）亦嘗試作類似的處理。

禁忌搜尋法大致流程圖如圖 2.1 所示。

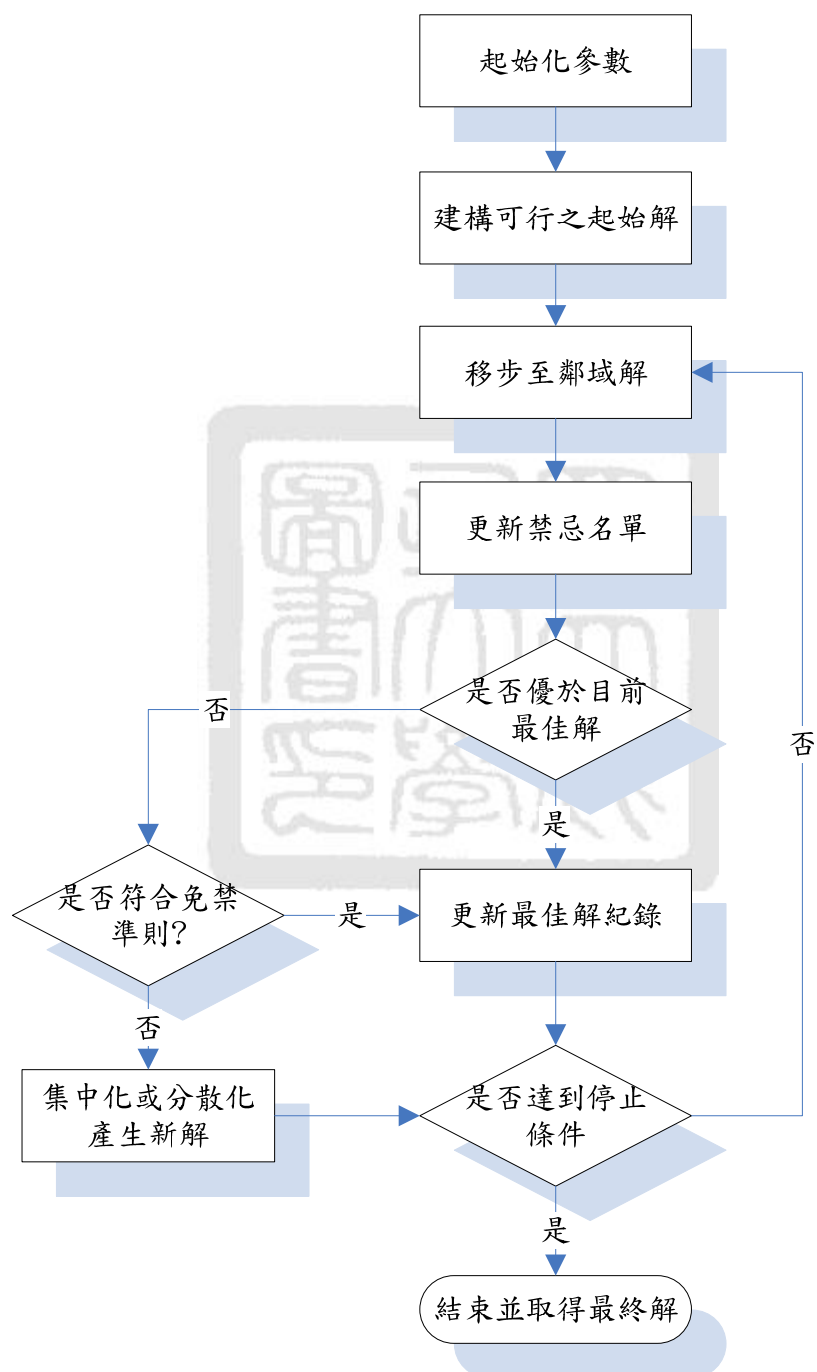


圖2.1 禁忌搜尋法大致流程圖

禁忌搜尋法在許多方面與模擬退火法類似，兩者以一個解為基礎再移至另一個解；然而，模擬退火法在移動到下一個解時，會給予機率來決定是否移動，而禁忌搜尋法是在禁忌名單中檢查此解是否曾經出現過；若有，則不移動且重新尋找新的鄰域解；若無，則移動到此解並移除禁忌名單中最舊的移步，以維持禁忌名單中一定的個數。由於禁忌名單的存在，在搜尋最佳解的過程中，一方面可避免回到上一次搜尋過的解，形成迴圈（cycling）的現象；另一方面，亦可避免求解時受限於區域最佳解。

禁忌搜尋法的應用層面相當廣泛，Glover（1989, 1990）分別提出了兩篇完整介紹禁忌搜尋法的文獻，此兩篇為禁忌搜尋法最重要的文獻，其中說明了禁忌搜尋法的精義及相關研究。很多文獻均提出禁忌搜尋法為基礎的方法來解決排程上的問題，例如 Armentano *et al.*（1999）利用禁忌搜尋法求解流線式生產排程的問題，以四種參數設定來訂定到期日，並對這四種參數作分析，來最小化總完工時間。Wardono and Fathi（2004）亦提出禁忌搜尋法來求解多階段平行機台中考慮有限暫存區下總完工時間最小化之排程問題。而在近期的文獻中，Lin and Ying（2009）針對流線式生產排程問題提出模擬退火法、禁忌搜尋法以及結合模擬退火法與禁忌搜尋法來求解總完工時間最小化。其中，結合模擬退火法與禁忌搜尋法的方式能夠求得較佳的總完工時間。Al-Anzi and Allahverdi（2009）針對兩階段流線式生產排程問題，提出禁忌搜尋法以及兩種啟發式方法來求解最大延遲時間及總完工時間最小化排程，並從實驗結果得知該兩種啟發式方法皆較禁忌搜尋法的績效表現優異。

## 2.4 全域下界法

由於在不少的文獻中均提到 FSMP 的排程問題皆屬於複雜度高的問題(Hoogeveen *et al.* 1996)，當求解的排程問題愈大，要找到最佳解的可能性將更加的困難。在無法保證有效求得最佳解且要符合現實製程之需求的情況下，通常都利用派工法則或智慧搜尋法來求得近似最佳解（Near-Optimal Solutions），因此在這領域的文獻中都只能以求得一組好的解為目標，而績效的比較則必須透過被定義出的全域下界法來衡量（Jin *et al.*

2006)。

Gupta 於 1988 年首先提出在兩階段 FSMP 問題的下界法，之後許多文獻也相繼定義出了不同的下界法 (Lee and Vairaktarakis 1994, Haouari and M'hallah 1997)。其後，Gupta *et al.* (1997) 針對第一階段為平行機台且第二階段只有一部機台的排程問題提出了另一種的下界法；Soewandi and Elmaghraby (2001) 延伸 Haouari and M'hallah (1997) 的研究，針對三階段 FSMP 排程問題發展出了數種求解下界的方法；Santos *et al.* (1995) 針對多階段 FSMP 排程問題提出了全域下界法；Jin *et al.* (2006) 結合了 Santos *et al.* (1995) 與 Haouari and M'hallah (1997) 的全域下界法來作為衡量指標。本研究將以 Jin *et al.* (2006) 的全域下界法為基礎，並針對本研究的問題特性（例如增加搬運時間與機台可開始處理時間）於 4.3 節發展出適合的衡量指標。



### 3. 研究方法

本章節將說明本研究所需使用的方法。第一節將問題結構的績效指標作一說明；第二節說明本研究的基本假設；第三節說明混整數規劃模型的建構以及減少混整數規劃模式變數之方法；第四節說明 BFIFO 派工法則的建構；第五節說明 BFIFO 派工法則結合混整數規劃模型；第六節說明 BFIFO 派工法則結合 TIT 改善機制；第七節說明禁忌搜尋法。

#### 3.1 問題描述

晶圓廠內機台加工的形式，可分為序列加工（Serial）與成批加工（Batch）。序列加工以一個工件 25 片晶圓為例，其加工方式為一次加工一片晶圓，需等 25 片加工完成，方可將整批工件搬運至下一道製程站點。而成批加工則是一次可同時處理 2 到 6 個工件不等（本研究假設第二階段機台最多一次能處理兩個工件），亦即成批的工件是在同一時間完成加工。也因為工件受加工方式的影響，機台的生產排程也就更加地複雜。

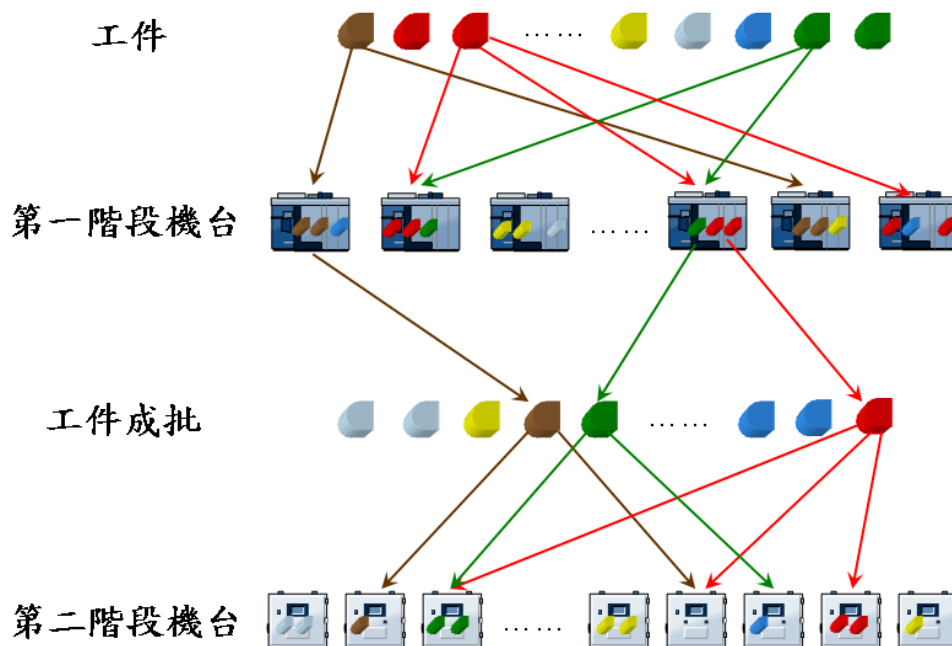


圖3.1 問題概略示意圖

如圖 3.1 所示，假設有  $N$  個工件必須經過兩階段的工作站，第一階段的工作站即屬於序列加工的方式，一次只能加工一個工件；第二階段的工作站可以同時加工兩個工件（亦即，成批加工）。我們將決定各工件應在哪一機台上加工，以使兩階段工作站中各機台之處理量均衡，並達到總完工時間最小化的目標。

本研究探討的基本假設有下列幾項：

- 僅考慮兩階段（亦即，兩道工作站）平行機台，其中第一階段為序列加工，而第二階段為成批加工
- 機台的加工無論是同產品不同步驟，或不同產品相同加工步驟，其處理特性（Recipe）也有不同
- 各工件將會因其處理特性的不同而只能在某些特定的機台上加工
- 當工件進入機台之後，該機台必須在結束其加工之後才能進行下個工件加工
- 每一個工件都需要經過各工作站中之某一機台一次
- 不考慮機台故障（Breakdown）的情形發生
- 工件在工作站之機台上所需的加工時間已知
- 工件搬運到每部機台的搬運時間已知
- 每一工件到達第一階段工作站的時間已知
- 各工作站有固定的加工次序
- 不考慮插單與交期變更

### 3.2 混整數規劃模型建構

本研究將針對此兩階段 FSMP 排程問題，建構一混整數規劃模型，以下先定義模型相關的數學符號，及混整數規劃模型的目標函式與限制式，如公式 3.1 至公式 3.23 所示。

#### 1. 給定之參數：

$i$ ：工件編號

$m$ ：第一階段機台編號

$m'$ ：第二階段機台編號

$p$ ：加工順序編號

$k$ ：批量編號

$j$ ：工件的處理特性編號

$I_m$ ：工件可在機台  $m$  加工的集合

$M_i$ ：第一階段機台可加工工件  $i$  的集合

$M'_j$ ：第二階段機台可加工處理特性  $j$  的集合

$I_j$ ：工件屬於第  $j$  種處理特性的集合

$I^k$ ：工件可包含在批量  $k$  的集合

$B_j$ ：批量屬於第  $j$  種處理特性的集合

$B^i$ ：批量可以放工件  $i$  的集合

$P_{i,m}$ ：工件  $i$  於機台  $m$  的處理時間

$P'_{k,m'}$ ：批量  $k$  於機台  $m'$  的處理時間

$t_{i,m}$ ：工件  $i$  到機台  $m$  的搬運時間

$t'_{i,m'}$ ：工件  $i$  到機台  $m'$  的搬運時間

$r_i$ ：工件  $i$  於第一階段的到達時間

$d_i$ ：工件  $i$  於第一階段的到期時間

$MT_m$ ：機台  $m$  可開始加工的時間

$MT'_{m'}$ ：機台  $m'$  可開始加工的時間

$Q_i$ ：工件  $i$  允許在後工作站等待加工的時間長度

$L$ ：極大的值（類比 $\infty$ ）

## 2. 決策變數：

$X_{i,m,p}=1$  代表工件  $i$  於機台  $m$  的第  $p$  個順序，否則  $X_{i,m,p}=0$

$Y_{i,k}=1$  代表工件  $i$  於批量  $k$  中，否則  $Y_{i,k}=0$

$Y'_{k,m',p}=1$  代表批量  $k$  於機台  $m'$  的第  $p$  個順序，否則  $Y'_{k,m',p}=0$

$Z_{i,k,m',p}=1$  代表工件  $i$  於批量  $k$  且於機台  $m'$  的第  $p$  個順序，否則  $Z_{i,k,m',p}=0$

$S_{m,p}$ ：第一階段機台  $m$  於第  $p$  個順序的開始加工時間

$C_{m,p}$ ：第一階段機台  $m$  於第  $p$  個順序的完成加工時間

$S'_{m',p}$ ：第二階段機台  $m'$  於第  $p$  個順序的開始加工時間

$C'_{m',p}$ ：第二階段機台  $m'$  於第  $p$  個順序的完成加工時間

$r'_i$ ：工件  $i$  於到達第二階段的時間

$d'_i$ ：工件  $i$  於第二階段的到期時間

$C_{\max}$ ：總完工時間

目標函式：

$$\text{Minimize } Z = C_{\max} \quad (3.1)$$

限制式：

$$\sum_{m \in M_i} \sum_p X_{i,m,p} = 1 \quad \forall i \quad (3.2)$$

$$\sum_{i \in I_m} X_{i,m,p} \leq 1 \quad \forall m, p \quad (3.3)$$

$$\sum_{i \in I_m} X_{i,m,p} \geq \sum_{i \in I_m} X_{i,m,p+1} \quad \forall m, p \quad (3.4)$$

$$S_{m,p} + \sum_{i \in I_m} P_{i,m} X_{i,m,p} \leq C_{m,p} \quad \forall m, p \quad (3.5)$$

$$C_{m,p} \leq S_{m,p+1} \quad \forall m, p \quad (3.6)$$

$$MT_m \leq S_{m,1} \quad \forall m \quad (3.7)$$

$$L \cdot X_{i,m,p} + r'_i + t_{i,m} - S_{m,p} \leq L \quad \forall i, m, p \quad (3.8)$$

$$L \cdot X_{i,m,p} - d_i + C_{m,p} \leq L \quad \forall i, m, p \quad (3.9)$$

$$L \cdot X_{i,m,p} - r'_i + C_{m,p} = L \quad \forall m, p \quad (3.10)$$

$$r'_i + Q_i = d'_i \quad \forall i \quad (3.11)$$

$$\sum_{k \in B^i} Y_{i,k} = 1 \quad \forall i \quad (3.12)$$

$$\sum_{i \in I^k} Y_{i,k} \leq 2 \quad \forall k \quad (3.13)$$

$$\sum_{m' \in M'_j} \sum_p Y'_{k,m',p} = 1 \quad \forall k \quad (3.14)$$

$$\sum_{k \in B^i} Y'_{k,m',p} \leq 1 \quad \forall m', p \quad (3.15)$$

$$\sum_{k \in B^i} Y'_{k,m',p} \geq \sum_{k \in B^i} Y'_{k,m',p+1} \quad \forall m', p \quad (3.16)$$

$$Z_{i,k,m',p} \geq Y_{i,k} + Y'_{k,m',p} - 1 \quad \forall k \in B^i, i, m', p \quad (3.17)$$

$$S'_{m',p} + \sum_{k \in B^i} P_{k,m'} Z_{i,k,m',p} \leq C'_{m',p} \quad \forall i, m', p \quad (3.18)$$

$$C'_{m',p} \leq S'_{m',p+1} \quad \forall m', p \quad (3.19)$$

$$MT'_{m'} \leq S'_{m',1} \quad \forall m' \quad (3.20)$$

$$L \cdot Z_{i,k,m',p} + r'_i + t_{i,m'} - S'_{m',p} \leq L \quad \forall i, k, m', p \quad (3.21)$$

$$L \cdot Z_{i,k,m',p} - d'_i + C'_{m',p} \leq L \quad \forall i, k, m', p \quad (3.22)$$

$$C_{\max} \geq C_{m',p} \quad \forall m', p \quad (3.23)$$

其中公式 3.1 為混整數規劃模型的目標函數，旨在最小化總完工時間；公式 3.2 表示每一個工件只能被排一次；公式 3.3 表示每部機台的每個順序（即工件位置）只能排一個工件；公式 3.4 表示若機台  $m$  的順序  $p$  未安排工件，則順序  $p+1$  亦不可安排工件；公式 3.5 表示機台  $m$  中第  $p$  個順序的工件  $i$  之開始加工時間加上其加工時間應早或等於其完成時間；公式 3.6 表示機台  $m$  中順序  $p$  的工件應早或等於該機台後一順序



(即順序  $p+1$ ) 工件的開始加工時間；公式 3.7 表示機台  $m$  最早的可開始加工時間應早或等於該機台第一個工件的開始加工時間；公式 3.8 表示機台  $m$  中順序  $p$  之工件  $i$  的開始加工時間應晚或等於該工件到達本工作站的時間 (亦即其到達前工作站的時間再加上其至本工作站的搬運時間)；公式 3.9 表示機台  $m$  中順序  $p$  的工件  $i$  之完成時間應早或等於其工作站的到期時間；公式 3.10 表示機台  $m$  中順序  $p$  的工件  $i$  之完成時間等於其到達後工作站的時間；公式 3.11 表示工件  $i$  到達後工作站的時間加上其允許在後工作站的等候時間應等於其於後工作站的到期時間；公式 3.12 表示每一個工件只能被排在一個批量中；公式 3.13 表示每個批量最多只能排 2 個工件；公式 3.14 表示每一個批量只能被排一次；公式 3.15 表示每部機台的每個順序只能排一個批量；公式 3.16 表示若機台  $m'$  的順序  $p$  未安排批量，則順序  $p+1$  亦不可安排批量；公式 3.17 表示當工件  $i$  被安排於批量  $k$ ，且批量  $k$  於機台  $m'$  的第  $p'$  個順序時，則  $Z_{i,k,m',p'}$  等於 1；公式 3.18 表示機台  $m'$  中順序  $p$  的開始加工時間加上其加工時間應早或等於機台  $m'$  中順序  $p$  的完成時間；公式 3.19 表示機台  $m'$  中順序  $p$  的完成時間應早或等於機台  $m'$  中順序  $p+1$  的開始加工時間；公式 3.20 表示機台  $m'$  的可開始加工時間應早或等於機台  $m'$  中第一個批量的開始加工時間；公式 3.21 表示機台  $m'$  中順序  $p$  的開始加工時間應晚或等於被安排至該工作站的任一工件  $i$  到達後工作站的時間 (即離開前工作站的時間) 再加上該工件搬運到機台  $m'$  的時間；公式 3.22 表示機台  $m'$  中順序  $p$  的工件  $i$  之完成時間應早或等於該工件於後工作站的到期時間；公式 3.23 表示總完工時間應晚或等於所有工件的完成時間。

以本研究問題為例：假設有  $N$  個工件，兩階段之工作站各有  $M$  部機台，則本研究中所提出的混整數規劃模型中將有  $N^2(NM+2M+1)$  個二元變數， $2N(2M+1)+1$  個實變數以及限制式  $M(3N^3+4N^2+8N+2)+5N$  條；此混整數規劃模型的確能正確地求解原排程問題；然而，在給定的 10 分鐘內，混整數規劃模型僅能處理 24 個工件各 6 部機台的小規模問題。因此以下我們將提出一種技巧來減少混整數規劃變數之規模，以在時限內可求解較大規模問題的最佳排程。

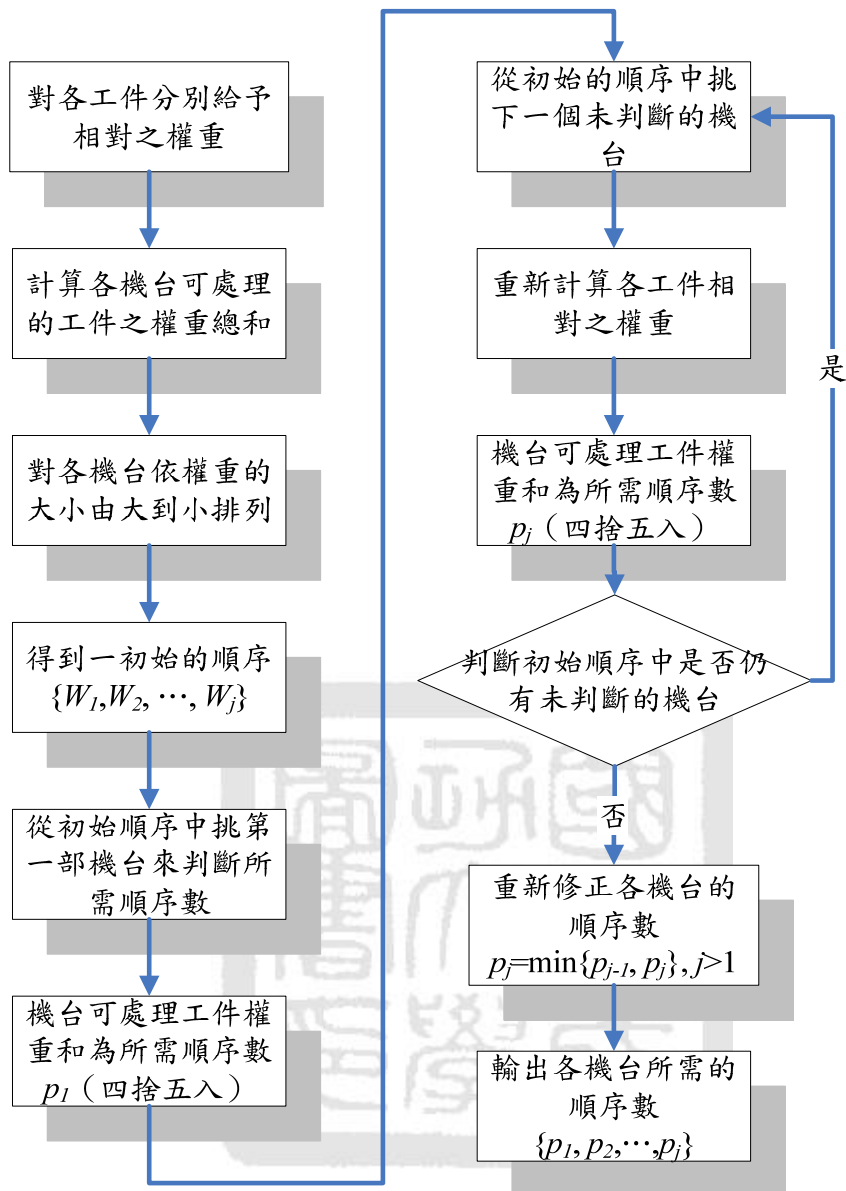


圖3.2 減少混整數規劃變數之技巧流程圖

如圖 3.2 所示，我們主要依據機台所能加工的工件數來判斷各機台所需的順序數，希望藉由減少順序數的作法來簡化原數學模型之規模，進而加快求解速度。以下將以一簡單的例子來說明上述之方法，圖 3.3 代表有 3 部平行機台及 8 個工件，每部機台分別只能處理灰底實線的工件，例如機台 1 只能處理編號 1 至 5 號的工件、機台 2 只能處理編號 5 至 8 號的工件以及機台 3 只能處理編號 1 至 4 號的工件。

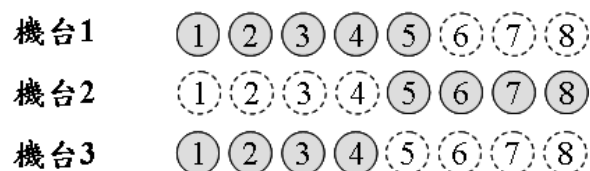


圖3.3 減少混整數規劃變數之技巧 (1)

首先根據每一工件可以處理該工件的機台數來分別給予權重，例如工件 1 能在機台 1 與機台 3 中選擇其中一部來加工，則該工件的權重為  $1/(\text{可加工機台數})=1/2$ ，以此類推，分別求出各工件相對之權重後，再計算各機台可處理的工件之權重總和，如表 3.1 所示。

表3.1 各機台權重之計算

	機台可處理工件的權重加總	各機台之權重
機台 1	$1/2+1/2+1/2+1/2+1/2=2.5$	2.5
機台 2	$1/2+1+1+1=3.5$	3.5
機台 3	$1/2+1/2+1/2+1/2=2$	2

將各機台的權重由大到小排列，並從權重最大的機台開始判斷該機台所需的順序數，例如目前權重最大的機台為機台 2，而機台 2 的權重為 3.5，則根據機台的權重 3.5 四捨五入後為 4，即為機台 2 所需的順序數。

由於機台 2 已經判斷過所需的順序數，因此刪除機台 2 不再判斷，如圖 3.4 所示，接著從初始順序中挑選下一個尚未判斷的機台 1，並分別求出各工件相對之權重後，再重新計算機台 1 可處理的工件之權重總和為  $1/2+1/2+1/2+1/2+1=3$ 。

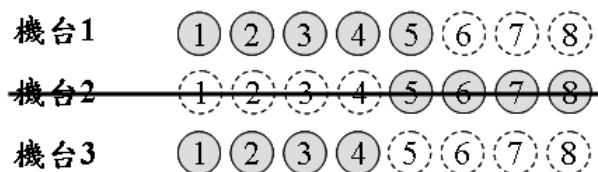


圖3.4 減少混整數規劃變數之技巧 (2)

接著再從初始順序中挑選尚未判斷的機台，由於機台 1 與機台 2 已經判斷過所需的順序數，因此機台 1、2 可被刪除，不再加入判斷，如圖 3.5 所示。分別求出各工件相對之權重後，重新計算機台 3 可處理的工件之權重總和為  $1+1+1+1=4$ 。

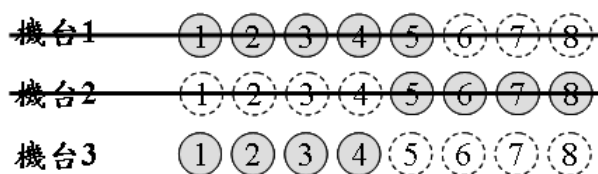


圖3.5 減少混整數規劃變數之技巧 (3)

接下來重新修正各機台所需的順序數，利用  $p_j = \min\{p_{j-1}, p_j\}$  且  $j > 1$  來做判斷，由上面依機台的權重由大到小所求得順序數分別為 4、3、4 ( $p_1$ 、 $p_2$ 、 $p_3$ )，由於  $j$  必須大於 1，因此從  $P_2$  開始判斷。從例子中來看， $p_2 = \min\{p_1, p_2\} = \min\{4, 3\} = 3$ ，則  $p_2$  仍然等於 3； $p_3 = \min\{p_2, p_3\} = \min\{3, 4\} = 3$ ，在此  $p_3$  將從原本的 4 修正為 3。根據以上修正的結果得到，依機台的權重由大到小（機台 2 > 機台 1 > 機台 3）所求得的順序數分別為 4、3、3，即為各機台最終所需的順序數。

以本研究問題為例：假設有  $N$  個工件，兩階段之工作站各有  $M$  部機台，則本研究中所提出的混整數規劃模型，在經過減少混整數規劃變數之技巧後，將有  $N^2(NM + 2M + 2)/2$  個二元變數， $N(2M+1)+1$  個實變數以及  $M(3N^3 + 4N^2 + 8N + 4)/2 + 5N$  條限制式，與原本尚未減少混整數規劃變數的數學模式來比較，其決策變數與限制式大約減少了 1/2。雖然此技巧的確能對給定的時限內解決更大規模問題有所幫忙，然而整體而言，此法仍將耗費不少時間。舉例來說，此法在 10 分鐘內僅能求解 48 個工件的可行（仍非最佳）排程。因此，以下我們先以 FIFO 派工法則所求得之解為基礎，再提出一套啟發式演算法，將效能不斷地改善之。

### 3.3 BFIFO 派工法則

為了因應現實環境中即時排程的需求，我們以能在 10 分鐘內將 160 個工件於兩階段各

40 部機台的製程順序排好為原則，測試並比較數種排程機制之效率及效能，進而找出最符合需求的建議解法。由於數學規劃模式通常會耗時甚久，因此在求解架構上我們擬先執行傳統的派工法則（Dispatching Rule），接著再就其所得之解加以改善。

在派工法則方面，我們測試了傳統的 FIFO（First In First Out）以及新的 BFIFO（Batch First In First Out）兩種派工法則。其中，FIFO 乃依據工件到達工作站的優先順序，由最早到達者優先進行加工；而 BFIFO 在第一階段工作站用 FIFO 排定後，其第二階段工作站的成批排程將由同一處理特性的工件中依其到達時間先後順序兩兩成批；由於 BFIFO 執行速度快速，若欲成批之總工件數  $n'$  為奇數，則我們會利用兩種方法來比較，第一種方法為由前而後的成批法，成批的工件分別為  $(1,2), (3,4), \dots, (n'-2, n'-1), (n')$ ，主要是希望藉由此方法將要成批的工件在到達第二階段工作站的時間較接近，其調整工件順序的步驟如下所示：

**步驟一：**首先將  $n$  個工件依到達第一階段的時間，由小到大排列得到初始加工順序，令  $i=1$ 。

**步驟二：**從第  $i$  個工件往後判斷是否有可成批的工件，若有則將第  $i$  個工件順序移至可成批的工件之前，並記錄此兩個工件；若無則只記錄第  $i$  個工件。

**步驟三：**若第  $i+1$  個工件已被記錄，則  $i=i+1$  並進行下一步驟；若無則  $i=i+1$  並回到步驟二。

**步驟四：**若所有工件都被記錄，則目前的排序為最終的排序；若無則回到步驟三。

第二種方法為由後而前的成批法，成批的工件分別為  $(1), (2,3), \dots, (n'-1, n')$ ，主要是觀察到第一種方法在工件為了要等待另一個成批的工件，一開始將導致機台閒置，因此藉由第二種方法在可成批的工件為奇數時，將第一個工件先行進入加工，避免一開始機台的閒置時間過長，其調整工件順序的步驟如下所示：

**步驟一：**首先將  $n$  個工件依到達第一階段的時間由小到大排列得到初始加工順序，令  $i=n$ 。

步驟二：從第  $i$  個工件往前判斷是否有可成批的工件，若有則將可成批的工件順序移至第  $i$  個工件之前，並記錄此兩個工件；若無則只記錄第  $i$  個工件。

步驟三：若第  $i-1$  個工件已被記錄，則  $i = i-1$  並進行下一步驟；若無則  $i = i-1$  並回到步驟二。

步驟四：若所有工件均被記錄，則目前的排序為最終的排序；若無則回到步驟三。

BFIFO 法則在排定所有工件的成批方式之後，再將各批工件依原 FIFO 法則排程之，並從此兩種方法取其效能較優者。此法則可避免要被成批的兩工件間產生過長等待的情況。BFIFO 派工法則流程圖如圖 3.6 所示。其中  $E_{im}$  的符號表示工件  $i$  於機台  $m$  的完工時間； $E_{km}$  的符號表示批量  $k$  於機台  $m$  的完工時間。

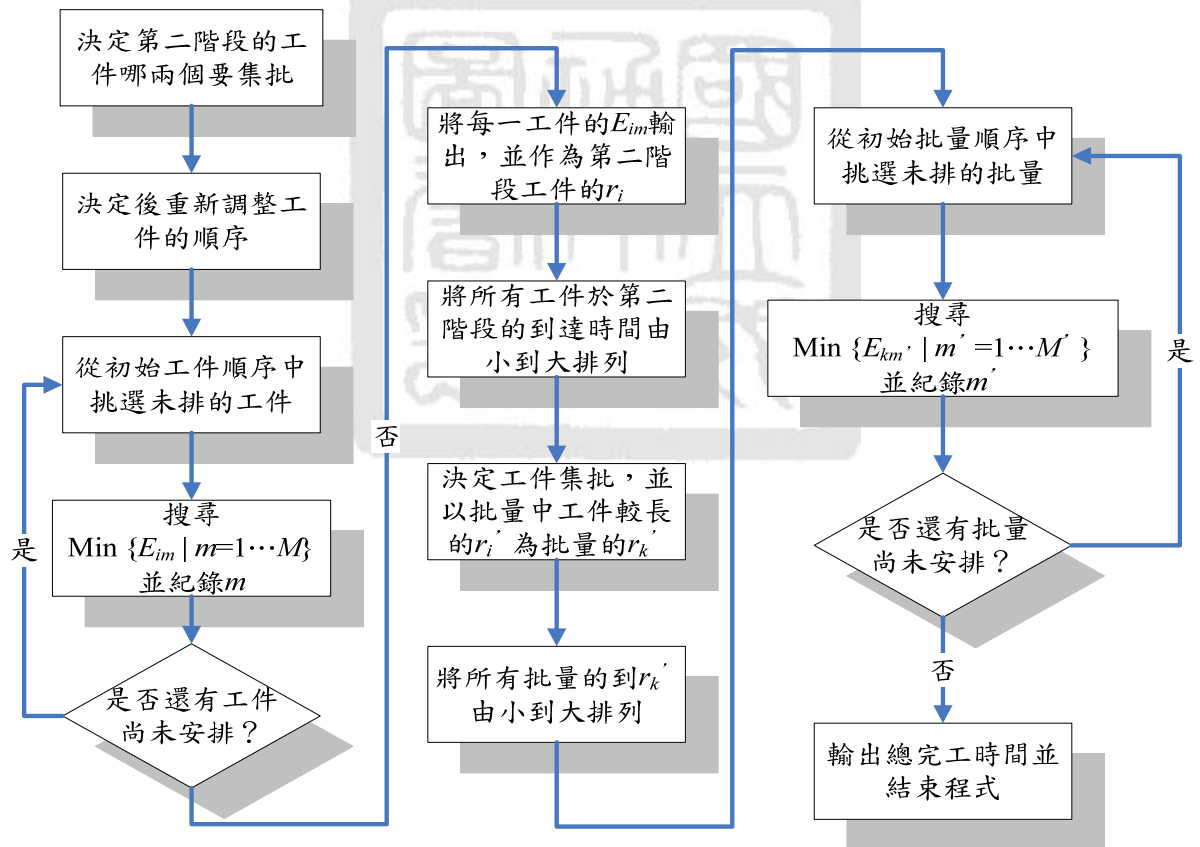


圖3.6 BFIFO 派工法則流程圖

### 3.4 BFIFO 派工法則結合混整數規劃模型

此機制旨在縮小原本混整數規劃模型的求解範圍，首先利用 BFIFO 派工法則求得一可行解，並將該可行解的成批方式固定，作為混整數規劃模型的起始解，再以一混整數規劃模型來重新決定各批量於後工作站中將被安排的機台及其順序。此機制之好處在於該混整數規劃模型之規模已大幅縮小（因為不用再考量如何成批，僅考量批量順序），因此較可在時限內求解較大的問題，其目標函式與限制式如下所示：

目標函式：

$$\text{Minimize } Z = C_{\max} \quad (3.24)$$

限制式：

$$\sum_{m' \in M'_j} \sum_p Y'_{k,m',p} = 1 \quad \forall k \quad (3.25)$$

$$\sum_{k \in B^i} Y'_{k,m',p} \leq 1 \quad \forall m', p \quad (3.26)$$

$$\sum_{k \in B^i} Y'_{k,m',p} \geq \sum_{k \in B^i} Y'_{k,m',p+1} \quad \forall m', p \quad (3.27)$$

$$S'_{m',p} + \sum_{k \in B^i} P_{k,m'} Y'_{k,m',p} \leq C'_{m',p} \quad \forall m', p \quad (3.28)$$

$$C'_{m',p} \leq S'_{m',p+1} \quad \forall m', p \quad (3.29)$$

$$MT'_m \leq S'_{m',1} \quad \forall m' \quad (3.30)$$

$$L \cdot Y'_{k,m',p} + r'_k + t'_{k,m'} - S'_{m',p} \leq L \quad \forall k, m', p \quad (3.31)$$

$$L \cdot Y'_{k,m',p} - d'_k + C'_{m',p} \leq L \quad \forall k, m', p \quad (3.32)$$

$$r'_k + Q_k = d'_k \quad \forall k \quad (3.33)$$

$$C_{\max} \geq C'_{m',p} \quad \forall m', p \quad (3.34)$$

其中公式 3.24 為混整數規劃的目標函數，旨在最小化總完工時間；公式 3.25 表示每一個批量只能被排一次；公式 3.26 表示每部機台的每個順序（即批量位置）只能排一個批量；公式 3.27 表示若機台  $m'$  的順序  $p$  未安排批量，則順序  $p+1$  亦不可安排批

量；公式 3.28 表示機台  $m'$  中順序  $p$  的批量  $k$  之開始加工時間加上其加工時間應早或等於機台  $m'$  中順序  $p$  的完成時間；公式 3.29 表示機台  $m'$  中順序  $p$  的完成時間應早或等於同機台之後一順序  $p+1$  的開始加工時間；公式 3.30 表示機台  $m'$  的可開始加工時間應早或等於該機台第一個批量的開始加工的時間；公式 3.31 表示機台  $m'$  中順序  $p$  的批量  $k$  之開始加工時間應晚或等於其到達後工作站的時間加搬運到機台  $m'$  的時間；公式 3.32 表示機台  $m'$  中順序  $p$  的批量  $k$  之完成時間應早或等於其於後工作站的到期時間；公式 3.33 表示批量  $k$  到達後工作站的時間加上其允許在後工作站的等候時間等於該批量於後工作站的到期時間；公式 3.34 表示總完工時間應晚或等於所有批量的完成時間。

在試驗過後，我們發現 BFIFO 派工法則結合混整數規劃模型在求解的過程中，混整數規劃模型尚能在 10 分鐘內於 16 個工件各 4 部機台的小例子中求得最佳解，但在工件個數超過 88 個且機台數各 22 部之後即無法在時限內對 BFIFO 派工法則於後工作站的起始解進一步地改善；遑論原來本研究必須解決 160 個工件，兩階段各 40 部機台的測試目標。此外，即使再利用先前所提出之減少混整數規劃變數的技巧，我們發現該機制的混整數規劃模型能於 10 分鐘內求得規模為 32 個工件各 8 部機台左右的問題最佳解，且對於規模為 144 個工件各 36 部機台以內的問題所得之可行解其品質平均皆比單純應用 BFIFO 派工法則所得之解更好。

而 BFIFO 派工法則結合混整數規劃模型之流程如圖 3.7 所示，其流程步驟說明如下：在第一階段工作站中，首先利用 BFIFO 派工法則產生工件初始的加工順序，並依照工件的先後順序來搜尋能夠最早完成加工的機台來進行加工，直到所有的工件均被排入為止；所有工件排入後，記錄各工件第一階段的完工時間，作為第二階段的到達時間；接著將工件依照第二階段的到達時間由小到大來排列來決定工件的成批，再依照批量的先後順序搜尋能夠最早完成加工的機台來進行加工，直到所有的批量均被排入為止；最後將 BFIFO 派工法則於機台中批量排定後的順序輸入到混整數規劃模型中作為起始解，並重新透過混整數規劃模型來決定批量的排程派工，之後即結束程式。





### 3.5 BFIFO 派工法則結合 TIT 改善機制

由於 BFIFO 派工法則在工件挑選機台時僅選擇最快完成加工之機台，無法預測機台限制對求解品質造成的影響，因此本研究再提出 TIT（即 Translocation、Interchange、Transposition）等三種改善求解品質機制，以 BFIFO 派工法則所得之解為基礎，持續地改善其求解品質，此三種求解機制分別將於章節 3.5.1 至章節 3.5.3 作更詳細的介紹，其 BFIFO 派工法則結合 TIT 改善機制流程圖如圖 3.8 所示：

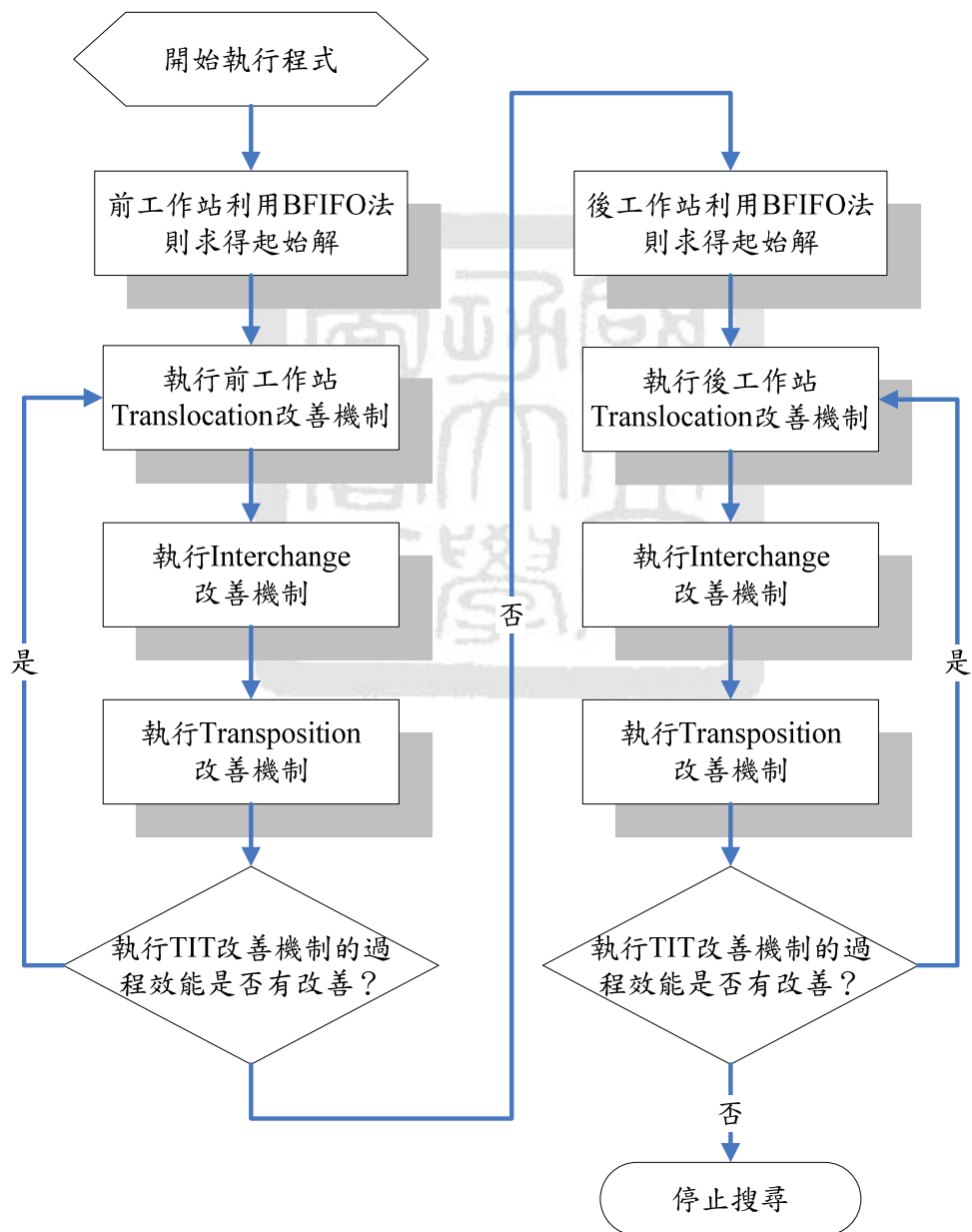


圖3.8 BFIFO 派工法則結合 TIT 改善機制流程圖

根據上圖 3.8 BFIFO 派工法則結合 TIT 改善機制流程圖流程執行步驟如下所示：

**步驟一：**在前工作站先利用 BFIFO 派工法則將工件指派到機台中來產生起始解。

**步驟二：**開始執行前工作站的 Translocation 改善機制。

**步驟三：**開始執行 Interchange 改善機制。

**步驟四：**開始執行 Transposition 改善機制。

**步驟五：**本次執行步驟二至步驟四的過程中，若效能有改善則回到**步驟二**；若無則進行下一步驟。

**步驟六：**工件到達後工作站後，利用 BFIFO 派工法則將工件成批後，將批量指派到後工作站的機台中來產生起始解。

**步驟七：**開始執行後工作站的 Translocation 改善機制。

**步驟八：**開始執行 Interchange 改善機制。

**步驟九：**開始執行 Transposition 改善機制。

**步驟十：**本次執行步驟七至步驟九的過程中，若效能有改善則回到**步驟七**；若無則停止搜尋。

而此三種求解改善機制的順序安排上，我們首先針對不同的排列組合（e.g. TIT, ITT, TTI,...,etc.）來進行效能收斂速度的測試，以提出一個較穩健的改善機制執行順序。由於 Transposition 改善機制是分別針對單一機台的最小完工時間來進行改善，並非該階段所有機台的最小總完工時間，因此不考慮 Transposition 改善機制與其他兩種改善機制之間排序的影響，僅測試 Translocation 改善機制以及 Interchange 改善機制之間排列組合對收斂程度的影響。

我們分別對前後工作站的排列組合測試了問題規模大小為 40-10-10、80-20-20、120-30-30 以及 160-40-40 的測試例，其中 40-10-10 代表 40 個工件，前後工作站各 10 部機台的問題規模。在此我們使用 160-40-40 問題規模為例子作說明，觀察前後工作站 Translocation 改善機制與 Interchange 改善機制排列組合的收斂程度分別如圖 3.9 與

圖 3.10 所示。圖 3.9 中 Stage1\_123 中的 Stage1 代表前工作站，而後面的 123 數字中之 1 代表 Translocation 改善機制、2 代表 Interchange 改善機制以及 3 代表 Transposition 改善機制，亦即此三種求解改善機制的求解順序為先做 Translocation，再做 Interchange，而最後做 Transposition。圖 3.10 中的 Stage2 代表後工作站，而 Stage1\_123+Stage2\_123 代表前工作站與後工作站執行的求解順序皆為 Translocation 改善機制、Interchange 改善機制以及 Transposition 改善機制。

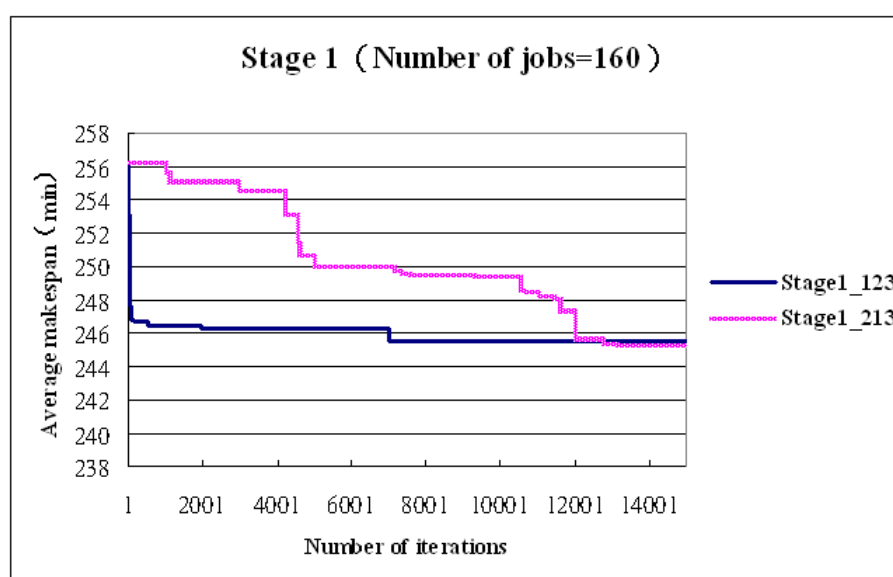


圖3.9 前工作站不同改善機制之先後順序排列的求解收斂情況

從圖 3.9 可得知，先執行 Translocation 改善機制再執行 Interchange 改善機制的收斂速度，明顯比先執行 Interchange 改善機制再執行 Translocation 改善機制來得快速；在比較其它可能的順序測試結果後，我們發現這個 123 的順序皆有不錯的表現，因此前工作站我們決定先執行 Translocation 再執行 Interchange，最後執行 Transposition 改善機制。



圖3.10 後工作站不同改善機制之先後順序排列的求解收斂情況

決定出前工作站三種改善機制的求解先後順序後，再進行後工作站的測試，從圖 3.10 中來觀察後工作站求解品質的收斂速度，如同圖 3.9 的結果，先執行 Translocation 改善機制再執行 Interchange 改善機制的收斂速度，仍然比先執行 Interchange 改善機制再執行 Translocation 改善機制來得快速，因此後工作站我們亦決定先執行 Translocation 改善機制再執行 Interchange 改善機制。然而，其他三種問題規模（40-10-10、80-20-20 以及 120-30-30）所測試的結果與上述的結果大致相同，因此僅以 160-40-40 的問題規模作說明。

以下我們將分別針對 Translocation 改善機制（前後工作站）、Interchange 改善機制以及 Transposition 改善機制來詳細介紹。

### 3.5.1 Translocation 改善機制

Translocation 改善機制將從完工時間最長的機台挑選一個工件「插入」到其它機台的工件順序中，後工作站的 Translocation 改善機制示意圖如圖 3.11 所示。首先，從最晚完工的機台中挑選第一個工件先插入到其他機台的第一個位置中，若沒有改善則再試

插入到該機台的第二個位置，以此類推，若有改善則更新工件位置後重新搜尋；反之，若插入到最後一個位置仍沒有改善，則挑選下一個預備被安插之工件繼續判斷，直到最晚完工之機台的所有工件均被挑選過且無法改善其完工時間時，則停止搜尋。

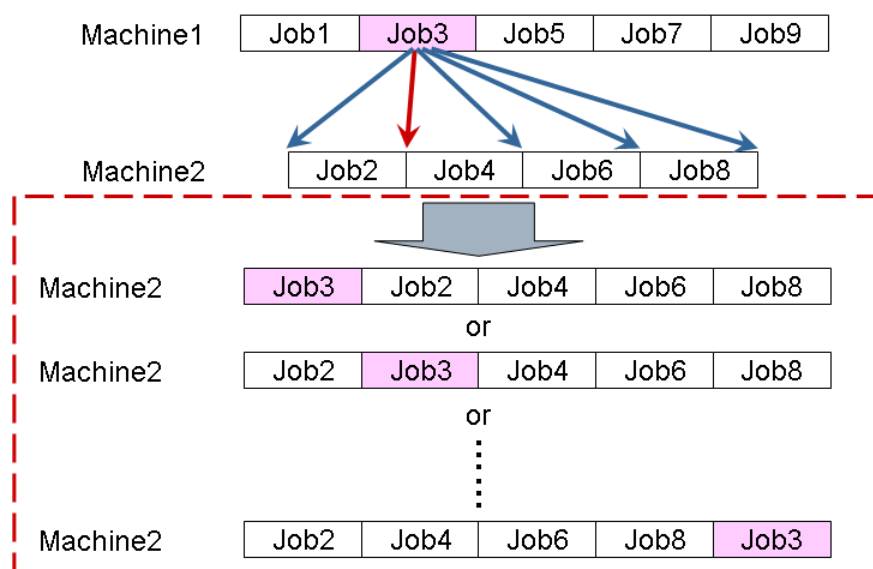


圖3.11 後工作站的 Translocation 改善機制

根據下圖 3.12 後工作站的 Translocation 改善機制流程執行步驟如下所示：

- 步驟一：將所有機台依照完工時間由大到小排列。
- 步驟二：從最晚完工時間的機台中挑選排序第一個加工的工件。
- 步驟三：工件若能在其他機台中加工，則進行下一步驟；若無則跳到步驟六。
- 步驟四：將工件插入到其他機台中的位置，若總完工時間有改善則更新工件的位置，並回到步驟一；若無則進行下一步驟。
- 步驟五：工件若能插入到其他位置，則回到步驟四；若無則進行下一步驟。
- 步驟六：最晚完工時間的機台若還有下一個排序的工件，則回到步驟三；若無則停止搜尋。

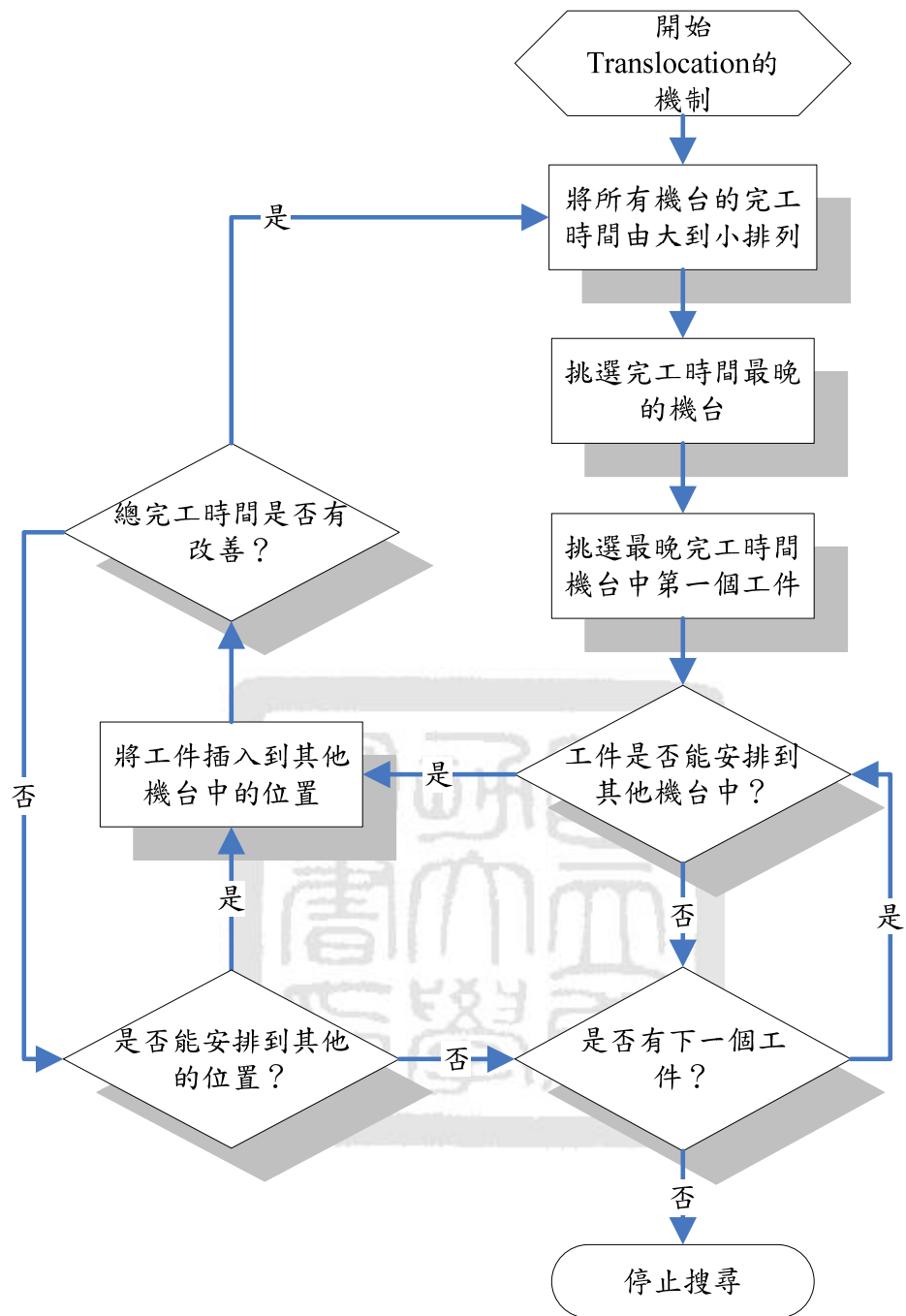


圖3.12 後工作站的 Translocation 改善機制流程圖

而前工作站若利用後工作站的 Translocation 改善機制，將會破壞了 BFIFO 法則的初始排序，其主要造成的影響如圖 3.13 所示，圖中 Job1 與 Job2 為同一種處理特性的工件，而當另一種處理特性的 Job3 插入到機台 2 後，雖然局部的解有所改善，卻導致原本於第二階段要成批的 Job1 在完成加工後，必須花更長的時間等待 Job2 加工完成

才能成批。

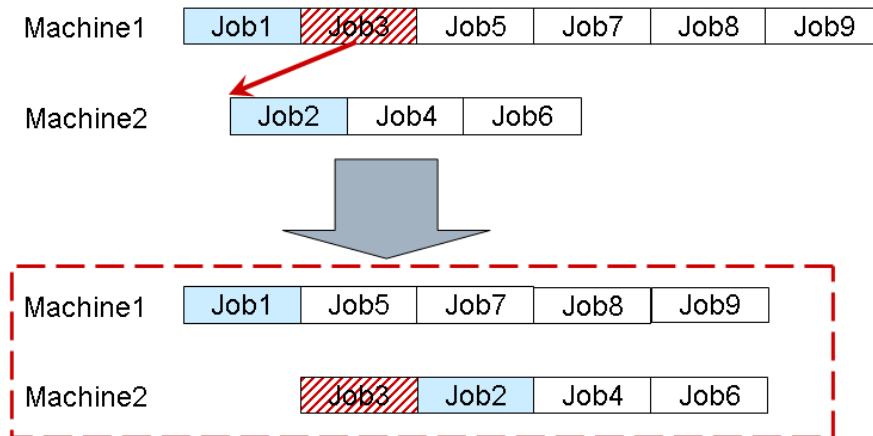


圖3.13 前工作站使用後工作站的 Translocation 對求解品質的影響

根據上述的考量，本研究決定採用另一種 Translocation 改善機制作為前工作站的求解機制，而前工作站的 Translocation 改善機制示意圖如圖 3.14 所示。該機制主要從完工時間最晚的機台中挑選其中一個工件，並將工件插入至其他機台的位置中，且工件插入的位置不能破壞 BFIFO 法則的初始排序。如圖中 Job1 至 Job8 為 BFIFO 法則的初始排序，則 Job3 要插入到機台 2 時，僅能插入至 Job2 與 Job4 之間，主要藉由此方式來避免破壞 BFIFO 法則的初始排序。

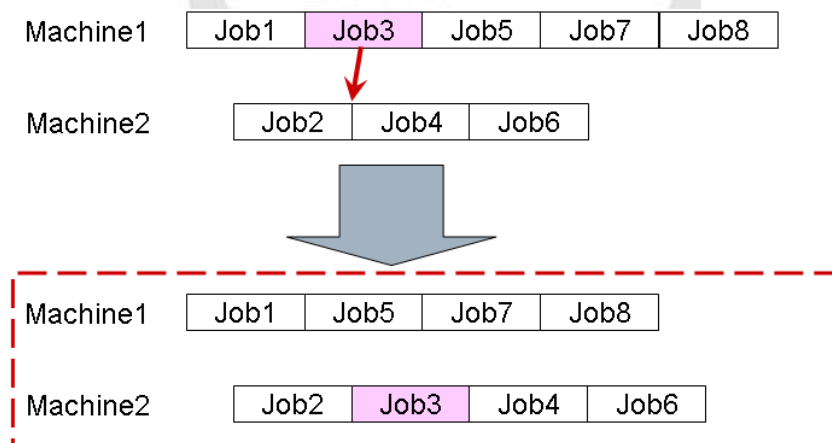


圖3.14 前工作站的 Translocation 改善機制示意圖

根據下圖 3.15 前工作站的 Translocation 改善機制流程的執行步驟如下所示：



步驟一：將所有機台依照完工時間由大到小排列。

步驟二：從最晚完工時間的機台中挑選排序第一個加工的工件。

步驟三：工件若能在其他機台中加工，則進行下一步驟；若無則跳到步驟五。

步驟四：將工件插入到其他機台中的位置，若總完工時間有改善則更新工件的位置，並回到步驟一；若無則進行下一步驟。

步驟五：最晚完工時間的機台若還有下一個排序的工件，則回到步驟三；若無則停止搜尋。

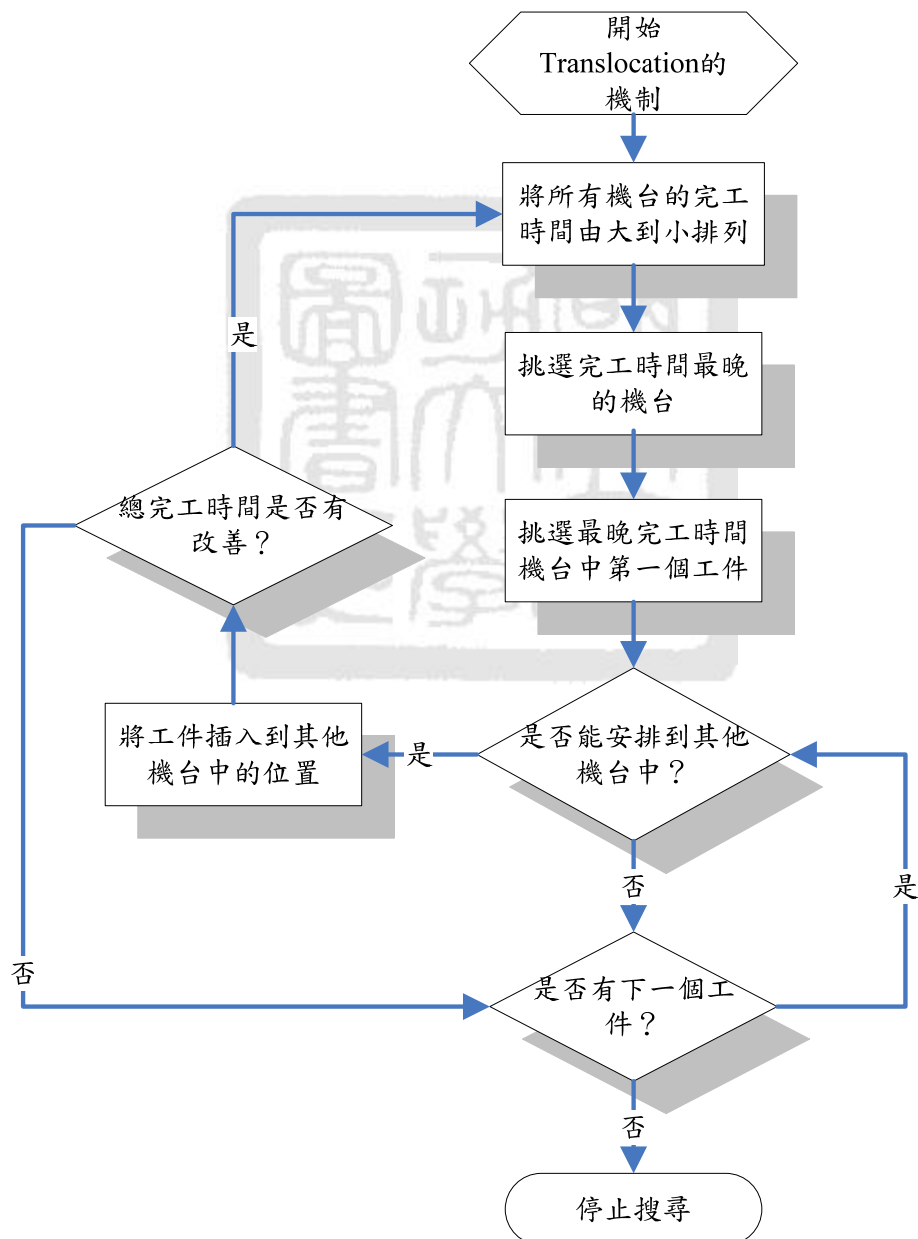


圖3.15 前工作站的 Translocation 改善機制流程圖

在此，本研究將針對前工作站使用後工作站的 Translocation 改善機制作測試，其中 Translocation 1 代表工件插入至其他機台的位置不能破壞 BFIFO 法則的初始排序，如圖 3.14 所示；Translocation 2 代表工件移至其他機台中從第一個位置開始插入，直到有改善或插入到最後一個位置為止，即為前工作站使用後工作站的 Translocation 改善機制，其示意圖如圖 3.11 所示。

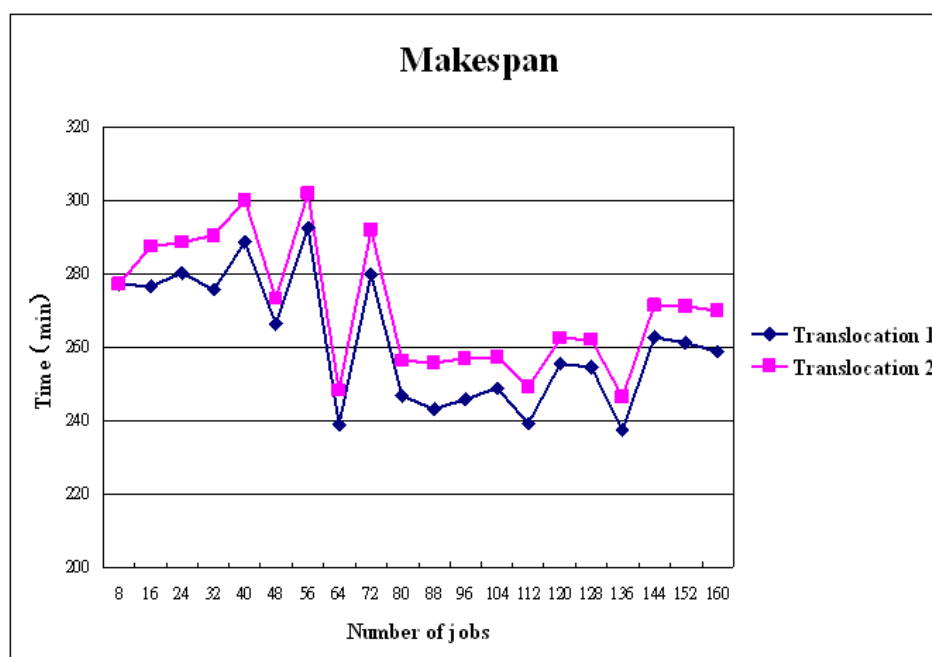


圖3.16 前工作站不同 Translocation 改善機制測試

從圖 3.16 的測試結果可得知，雖然 Translocation 1 僅允許工件可插入至不能破壞其初始排序的位置，但第一階段工作站利用此求解機制所得到的求解品質，反而比 Translocation 2 的求解品質佳。根據上述的結果，前工作站若利用後工作站的 Translocation 改善機制，的確會破壞了 BFIFO 法則的初始排序，因此本研究在前工作站決定採用圖 3.15 的 Translocation 改善機制流程。

### 3.5.2 Interchange 改善機制

Interchange 改善機制旨在將不同機台間之工件位置「互換」，如圖 3.17 所示，當

完工時間最晚的機台中原本安排加工的工件允許在另一部機台上加工，且另一部機台上原本安排的某一工件亦能在最晚完工的機台加工時，則交換彼此的位置。此改善機制主要藉由不同機台間的工件互換位置，來改善各階段的最終完工時間。

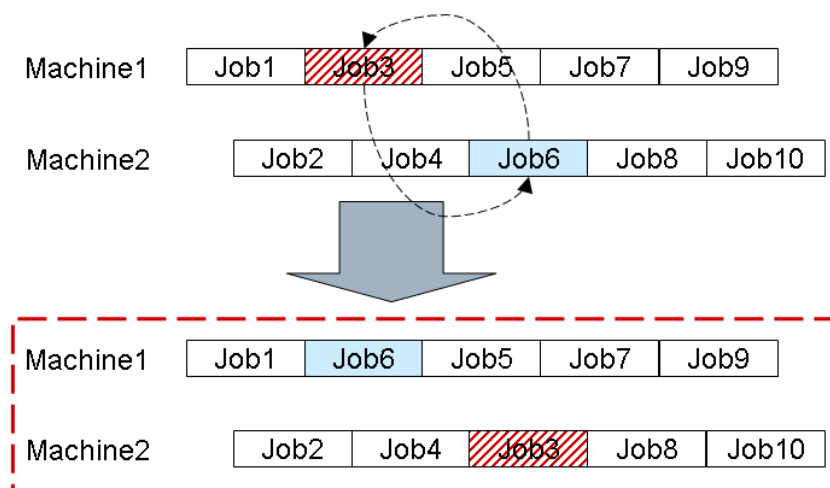


圖3.17 Interchange 改善機制示意圖

圖 3.18 之 Interchange 改善機制流程的執行步驟如下所示：

- 步驟一：將所有機台依照完工時間由大到小排列。
- 步驟二：從最晚完工時間的機台中挑選排序第一個加工的工件。
- 步驟三：工件若能在其他機台中加工，有進行下一步驟；若無則跳到步驟六。
- 步驟四：在機台限制的條件下判斷其他機台是否另有工件能進行交換，若有則進行下一步驟；若無則跳到步驟六。
- 步驟五：將不同機台間之工件交換位置，若總完工時間有改善則更新工件的位置，並回到步驟一；若無則回到步驟四。
- 步驟六：最晚完工時間的機台中是否還有下一個排序的工件，若有則回到步驟三；若無則停止搜尋。



### 3.5.3 Transposition 改善機制

Transposition 改善機制將「同一」機台之工件位置「互換」，如圖 3.19 所示，主要是將機台中第一個等待加工的工件換到第二個位置，若該機台的完工時間沒有改善，則再換到第三個位置，直到有改善或換到最後一個位置為止；若有改善則更新工件位置後重新搜尋，若換到最後一個位置仍沒有改善，則挑選機台中下一個工件繼續判斷，直到所有的工件均判斷過且都沒有改善則停止搜尋。在執行求解改善機制時，我們將先測試該「交換」或「插入」動作是否造成求解品質下降，在不會降低求解品質的情況下才實質進行該調整機制。

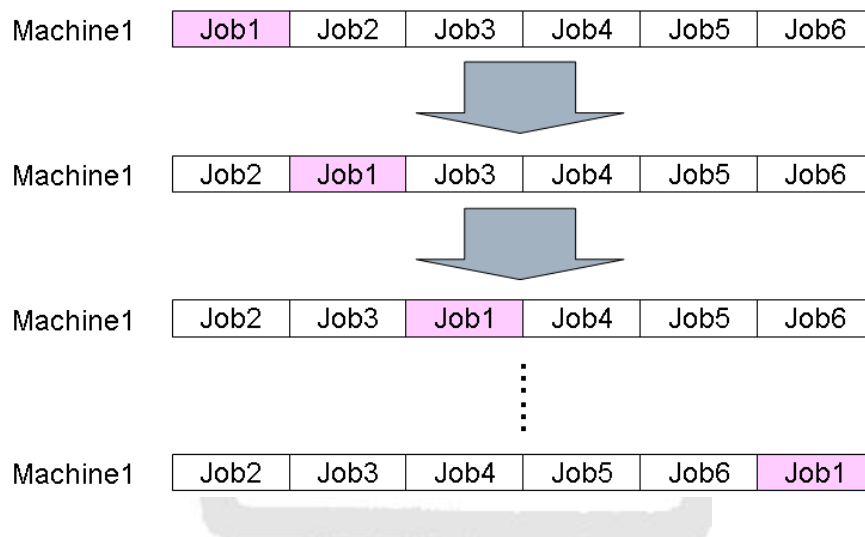


圖3.19 Transposition 改善機制示意圖

根據下圖 3.20 中 Transposition 改善機制流程的執行步驟如下所示：

**步驟一：**令  $k=1$

**步驟二：**挑選第  $k$  部機台中第一個等待加工的工件。

**步驟三：**將工件換到同一機台中下一個工件之後。

**步驟四：**若完工時間有改善則更新位置後回到**步驟一**；若無則進行下一步驟。

**步驟五：**若還能換到下一個工件之後，則互換後回到**步驟三**；若無則進行下一步驟。

步驟六：令  $k = k+1$ ，如果  $k$  大於機台數則停止，若為其他則回到步驟二。

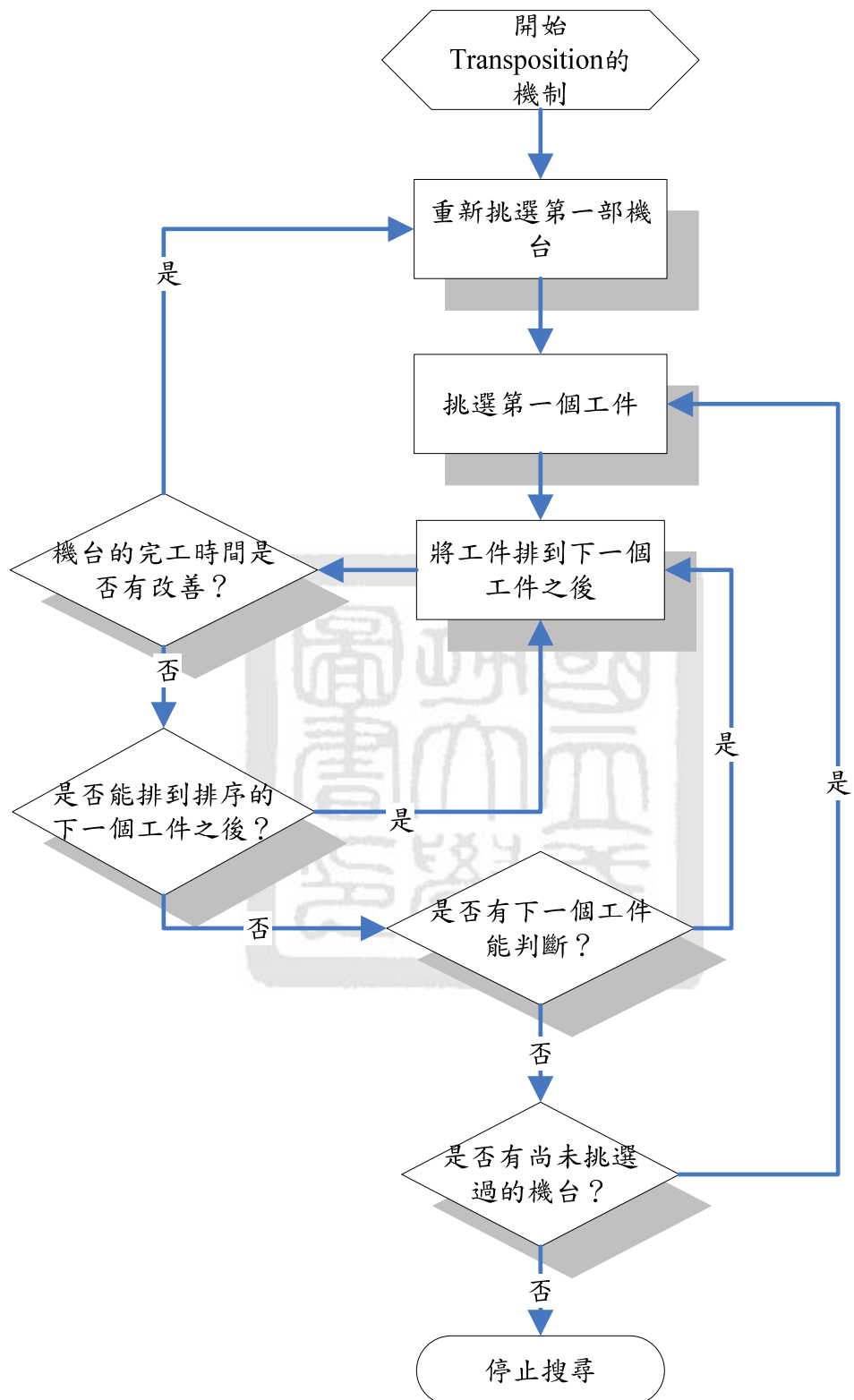


圖3.20 Transposition 改善機制流程圖

### 3.6 禁忌搜尋法

本研究的禁忌搜尋法是採用 Yang *et al.* (2004) 所提的方法，文獻中在針對 FSMP 排程最佳化的研究中，著重在結合模擬與禁忌搜尋法來求解工件排序的排程規劃，而文獻中亦提到禁忌搜尋法在解決排序的問題上有其應用性及穩健性，且都有相當良好的表現，因此本論文採用此篇文獻所提出的禁忌搜尋法來做實驗，其它關於禁忌搜尋法的各步驟流程以及相關參數（禁忌期限、終止條件等）將於本節中詳細描述。

#### 3.6.1 起始解

在解決排程問題前，一開始將根據問題的特性來產生起始解，以提高禁忌搜尋法最終所求得的績效之表現，一般常見的手法是利用隨機的方式產生，而本論文中所提出的 BFIFO 派工法則相較於傳統的 FIFO 派工法則對於降低總完工時間上有不錯的表現，因此利用 BFIFO 派工法則作為產生起始解的法則。建立在此起始解的基礎之上，持續地搜尋最佳的可行解。

#### 3.6.2 移步

在禁忌搜尋過程中，依鄰域解的定義方式，每一個解都會存在著相對應的鄰域解，而將目前的解轉換為其中一個鄰域解的步驟，稱之為「移步」(move)。本論文將採用鄰近交換法與兩兩交換法兩種方式來做比較。選擇使用鄰近交換法的主要原因是考量排程的即時性，因為當有  $n$  個工件順序時，從排序中隨機選取一個位置  $i$ ，並將  $i$  及  $i+1$  交換位置（如圖 3.21 所示），所有鄰近交換的可能只有  $n-1$  種。由於鄰近交換法在隨機選取工件交換的過程中，僅考慮交換  $i$  及  $i+1$  的位置，將導致求解品質無法有效的改善，因此另外又利用兩兩交換法來做比較；即藉由隨機交換任兩筆工件之順序來產生新的解（如圖 3.22 所示），當有  $q$  個工件順序時，所有兩兩交換的可能有  $q(q-1)/2$ ，

來探討利用兩兩交換的方式對於求解時間與求解品質的影響。

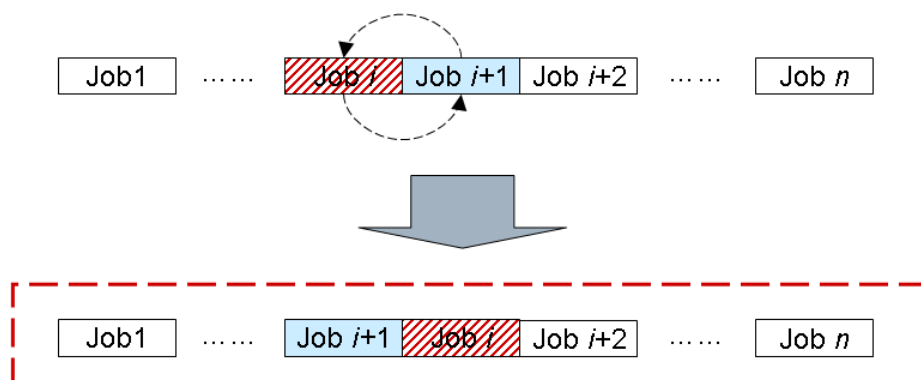


圖3.21 鄰近交換法示意圖

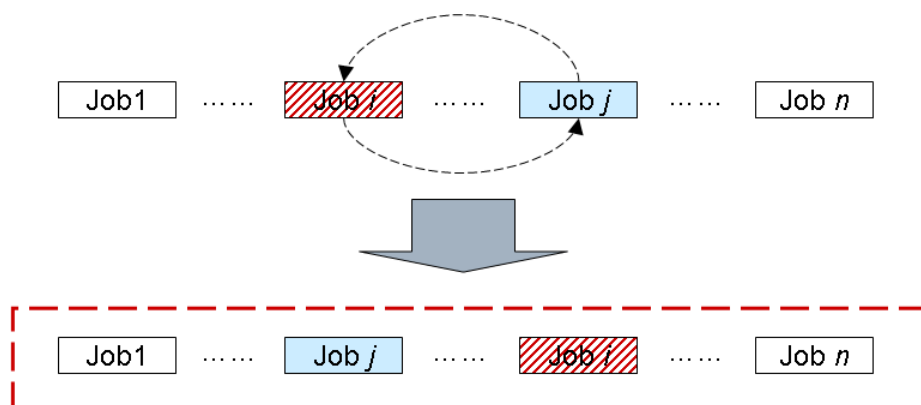


圖3.22 兩兩交換法示意圖

若藉由移步所產生的新解其績效相對於原本的解還佳的話，則將新的解取代原本的解並持續利用移步的方式搜尋下去，然而在選擇新解的方式中有三種策略 (Osman, 1991)，分別為最佳移步選取策略、優先改善選取策略以及樣本最佳移步策略，其說明如下：

#### 最佳移步選取策略

首先利用移步的方式產生所有的新解，並形成一個候選名單 (Candidate List)，藉由評估的方式將候選名單中的所有新解評估完成後，選擇其中最佳的解來取代原本的解。

#### 優先改善選取策略



以移步的方式隨機產生一個新解，經由評估後若此新解的績效優於原本的解，則此新解直接取代原本的解；若無，則繼續以移步的方式來隨機產生新的解，直到出現的新解優於原來的解為止。

#### 樣本最佳移步策略

此策略介於最佳移步選取策略以及優先改善選取策略之間，其方法為利用移步的方式產生部分樣本的候選名單（此名單為最佳移步選取策略中候選名單的子集合），然後利用最佳移步選取策略相同的方式將候選名單中的所有的新解評估完成後，從中選擇最佳的解來取代原本的解。

由於半導體廠中的生產排程必須具備即時性的需求，為了避免搜尋的過程中在評估較差的解上花過長的時間，因此本研究以優先改善選取策略來決定取代原本的解。

### 3.6.3 免禁準則

在禁忌演算法搜尋的過程中，當所有的移步都無法再找到更佳的鄰域解時，則會開始搜尋禁忌名單中的移步；若從禁忌名單中的移步發現更佳的鄰域解時，將以此鄰域解來取代現有的解而成為新的解；由於此種作法忽略了原本禁忌名單的機制，因此稱為免禁準則（Aspiration criterion）。

### 3.6.4 集中化及分散化

一個較為先進的禁忌搜尋法除了短期的記憶之外，還包含長期的記憶如集中化及分散化，典型的集中化作法為利用過去搜尋中最佳的解作為新的起始解來重新搜尋，或是採用 Glover and Kochenberger（2003）改變鄰域結構（Neighborhood Structure）的作法來縮小搜尋的方向，例如將移步的優先改善選取策略改變為樣本最佳移步策略，或是將樣本最佳移步策略的樣本增加，或者是改成以最佳移步選取策略來進行移步。

分散化則主要有兩種方式，第一種為重新分散（Restart Diversion），此方式主要

是以過去搜尋中較少出現的或者是過去搜尋中最佳的解，接著以一個新的解重新搜尋。第二種為繼續分散（Continuous Diversification），則是將分散化整合到搜尋法中，將過去搜尋過解的頻率元素（Frequency Component）考慮在未來求解時新解評估的目標函數上。

而 Yang *et al.*（2004）採用的是 Ben-Daya and Al-Fawzan（1998）所提的方式來進行分散化以及集中化，當免禁準則後依舊無法搜尋到比現存解還好的解時將執行以下的策略來產生另一個新解重新開始。由於以另一個新解繼續搜尋的過程稱為分散化，又此新的解為強化過去解的頻率元素所產生的，因此為一個結合分散化與集中化的作法。以下為一個新解產生的方法：

假設有一個  $q \times q$  矩陣，分別代表  $n$  個工件及  $n$  個順序位置，矩陣中的值以  $f_{ij}$  表示，代表工件  $i$  被排在順序  $j$  的總次數，則集中化及分散化的執行步驟如下所示：

步驟一：令  $k=1$

步驟二：在整個矩陣中找出一最大值之  $f_{ij}$

步驟三：將工件  $i$  分派至第  $j$  個順序

步驟四：刪除矩陣中之第  $i$  列與第  $j$  行

步驟五：令  $k=k+1$

步驟六：如果  $k>n$  則停止，反之則回到步驟二

以下將參照郭宜雍（民 94）研究中所使用的例子，來說明上述的方法，如圖 3.23 為一  $5 \times 5$  的矩陣，其為禁忌搜尋法之長期記憶，記錄在過去的搜尋中每個工件在每一順序出現的次數，而圖 3.23 為禁忌搜尋法成功移步六次之後長期記憶的結果，其中第四列第二行中的 4 代表在成功移步六次的結果中，編號四號的工件排在第二個順序的次數總共有 4 次。

		順序				
		1	2	3	4	5
工 件	1	3	2	0	0	1
	2	0	0	5	0	1
	3	1	0	1	3	1
	4	1	4	0	1	0
	5	0	1	0	2	3

圖3.23 禁忌搜尋法之長期記憶 1 (修改自 郭宜雍，民 94)

首先令  $k=1$ ，並尋找矩陣中數字最大者，為第二列第三行中的 5，因此將第二個工件放在第三個順序，並將第二列以及第三行刪除，其結果如圖 3.24 所示。

		順序				
		1	2	3	4	5
工 件	1	3	2	0	0	1
	2	0	0	5	0	1
	3	1	0	1	3	1
	4	1	4	0	1	0
	5	0	1	0	2	3

順序				
		2		

圖3.24 禁忌搜尋法之長期記憶 2 (修改自 郭宜雍，民 94)

接著令  $k=1+1=2$ ，然後在從矩陣中尋找數字最大者，為第四列第二行中的 4，因此將第四個工件放在第二個順序，並將第四列以及第二行刪除，其結果如圖 3.25 所示。

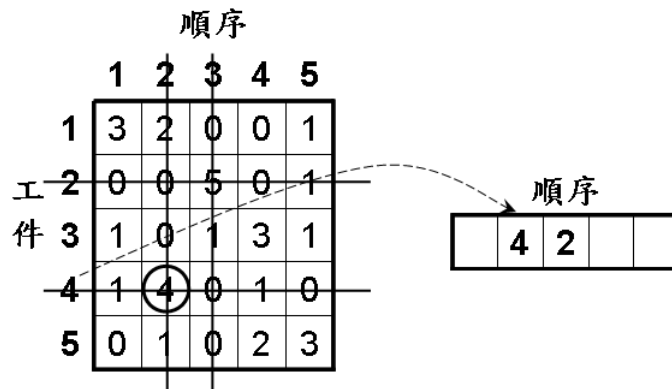


圖3.25 禁忌搜尋法之長期記憶 3 (修改自 郭宜雍，民 94)

再令  $k=2+1=3$ ，再次尋找矩陣中數字最大者為 3，但矩陣中一共有三個，分別為第一列第一行、第三列第四行以及第五列第五行，此時將先從三個中任選一個，假設選擇第一列第一行，則將第一個工件排在第一個順序，並將第一列第一行，其結果如圖 3.26 所示。

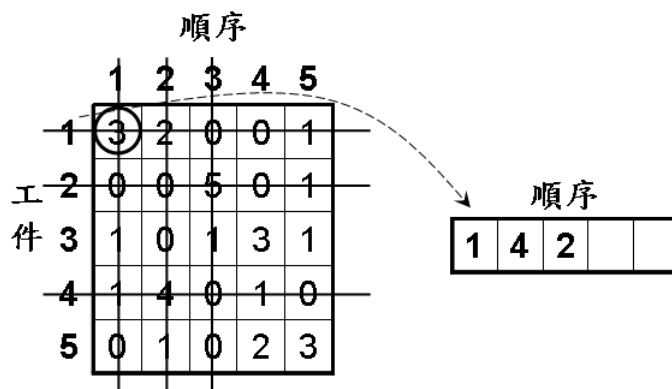


圖3.26 禁忌搜尋法之長期記憶 4 (修改自 郭宜雍，民 94)

同理將第三個工件排在第四個順序以及第五個工件排在第五個順序，執行至  $k=6$  時即停止，則最終的結果為 1-4-2-3-5。

此種方式所產生的新解不但使搜尋法從另一個解空間重新開始，而且可能得到更好的解，因此本方式結合了集中化及分散化的特性。

### 3.6.5 禁忌搜尋法參數設定

在智慧搜尋法中，好的設定通常能得到較佳的搜尋結果，由於半導體廠中的生產排程必須具備即時性，求解時間對於參數設定也是很重要的考量，因此如何在最短的求解時間內得到較佳的結果，將取決於智慧搜尋法中的參數設定。而智慧搜尋法在搜尋的過程中包含了隨機的過程（例如禁忌搜尋法的移步方式），這些過程會降低搜尋的結果對於參數設定或起始解變化的敏感度，故禁忌搜尋法僅期望得到一組局部最佳（Local Optimal）的參數而不期望得到一組全域最佳（Global Optimal）的參數。

禁忌搜尋法中有兩個重要的參數，一為禁忌期限（Tabu Tenure）；另一個為終止條件，接下來將介紹此兩種參數設定。

#### 禁忌期限

禁忌期限代表一個移步在進入禁忌名單後要經過多久才能從名單中消除，主要目的就是利用短期記憶結構記錄前幾次的移動，當一個移步發生之後，避免在禁忌期限內又再次的發生而形成迴圈。Yang *et al.* (1999) 提出假設問題的大小為  $q$ ，則禁忌期限應該介於  $1/3q$  到  $3/2q$  之間的整數。由於問題大小會影響求解的時間因而增加搜尋的成本，因此在此以 40 個工件兩階段各 10 部機台為試驗對象，藉由試驗的方式來決定該參數的設定，並以  $0.6q$ 、 $0.9q$ 、 $1.2q$  及  $1.5q$  來比較不同禁忌期限之表現，結果如圖 3.27 所示，可發現當禁忌期限為  $0.6q$  時搜尋收斂的速度較快，因此我們決定以  $0.6q$  作為禁忌期限的參數設定。

在禁忌期限的設定上，應該針對不同的例子都來進行上述的測試，從每個例子中都找出一個最佳的禁忌期限設定，然而本研究後續將測試二十群不同問題規模的例子，若每一群均個別測試出一個最佳的禁忌期限設定，將耗費不少時間，因此本研究暫時先以 40 個工件的測試結果作為禁忌期限的參數設定。

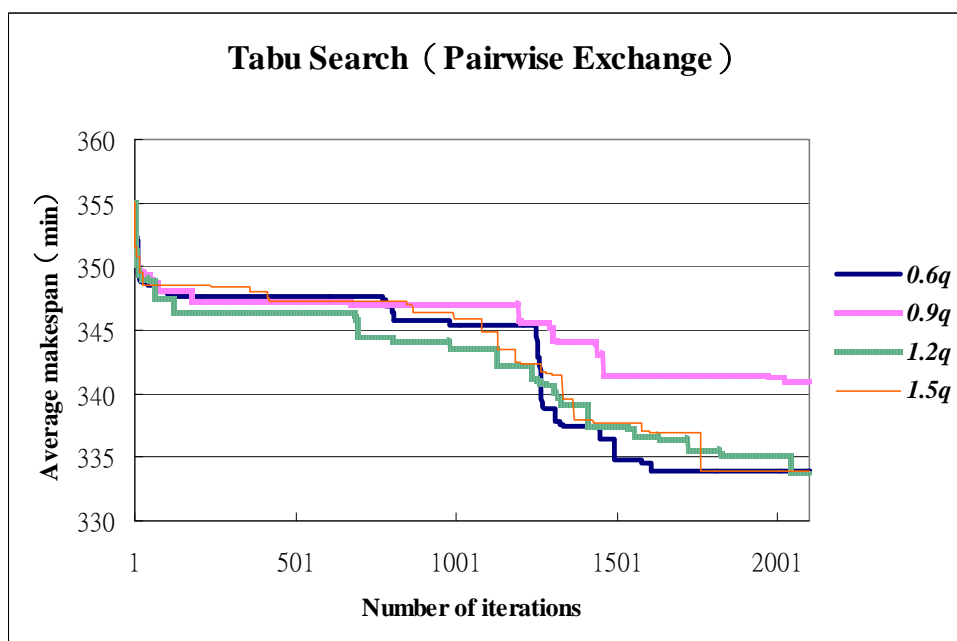


圖3.27 不同禁忌期限之收斂情況

#### 終止條件

禁忌搜尋法用來終止搜尋進行之條件，常用方法如下：

- i. 目標函數在一定迭代次數內未改善則停止搜尋
- ii. 當達到預設之可接受目標即停止搜尋
- iii. 當搜尋達到預設的最大迭代次數則停止搜尋
- iv. 當搜尋達到特定時間時即停止搜尋

由於以第一種作法常會發生搜尋過程太早結束而導致最終結果不佳，或是搜尋過久導致花費相當長的求解時間；第二種作法也常造成花費相當長的求解時間仍無法達到預設之可接受目標；而第三種作法以及第四種作法相似，在此將以最大迭代次數作為終止條件，當給定某個長度為  $q$  的順序時，所有兩兩交換的可能為  $q(q-1)/2$ ，並令  $q(q-1)/2$  為  $Q$ 。根據圖 3.27 的收斂情況來看，當  $q$  為 40 時  $Q$  為 780，當迭代次數超過 1500 之後搜尋的結果會進入收斂。因此在本研究的應用上認為當搜尋的次數超過  $2Q$  時，則停止搜尋，並以搜尋過程中最佳的解作為最終解。

在 Yang *et al.* (2004) 的禁忌搜尋法之流程如圖 3.28 所示，其中各代號之解釋如

下：

$T$ =禁忌期限； $M_{count}$ =终止條件；Count=迭代次數； $Z^*$ =最佳值； $X^*$ =最佳解；

$X_{now}$ =現存解； $X_{next}$ =鄰域解

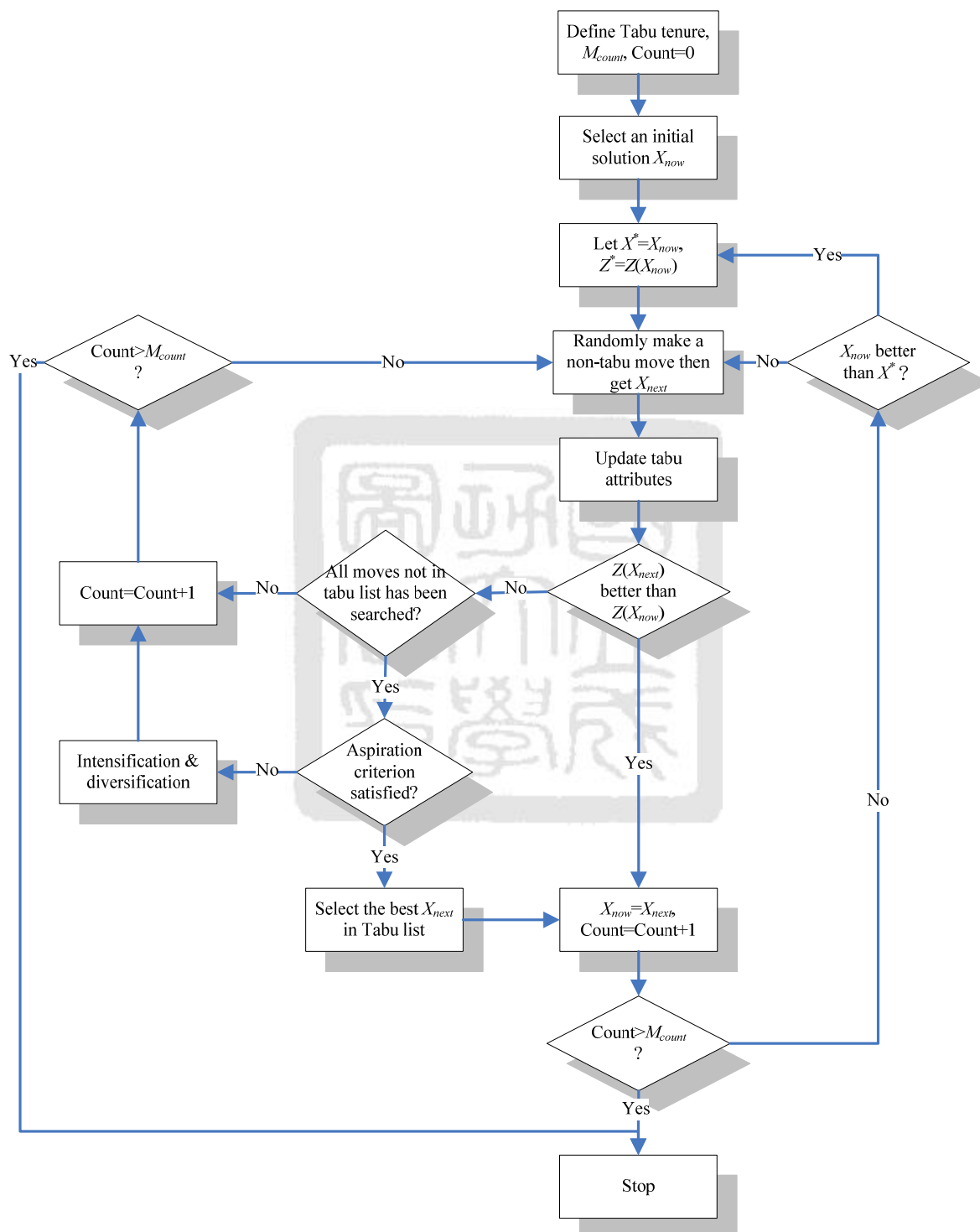


圖3.28 禁忌搜尋法之流程 (Yang et al., 2004)

根據圖 3.28 禁忌搜尋法之流程描述如下：首先設定禁忌搜尋法之相關參數，如禁忌期限 ( $T$ ) 以及終止條件中的最大迭代次數 ( $M_{count}$ )；接著以 BFIFO 派工法則產生一組起始解，並令此解為現存解 ( $X_{now}$ )、局部最佳解 ( $Z^*$ ) 以及最佳解 ( $X^*$ )。之後，再分別以鄰近交換法與兩兩交換法，隨機選取尚未存在於禁忌名單中的工件進行交換，來產生新的鄰域解 ( $X_{next}$ )，並更新禁忌搜尋中長期及短期的記憶。若此鄰域解比現存解還好，則將鄰域解取代現存解成為新的現存解；若新的現存解還比目前為止之最佳解好的話則還會取代最佳解，之後再以新的現存解來搜尋新的鄰域解。

反之，若鄰域解表現較差，則判斷是否所有非禁忌移步搜尋的鄰域解皆已搜尋過；若還未全部搜尋過，則繼續以原有的現存解搜尋新的鄰域解；反之則以禁忌移步來搜尋鄰域解中的最佳解。若鄰域解比現存解好，則與先前方式一樣取代現存解成為新的現存解；若此解還比目前為止之最佳解好的話則還會取代最佳解；之後再以新的現存解搜尋新的鄰域解。若上述之免禁準則未成立，便利用 Ben-Daya and Al-Fawzan (1998) 所提的集中化及分散化的方式來產生新的解，並令此解為現存解 ( $X_{now}$ ) 以及目前為止之最佳值 ( $Z^*$ )。

在整個搜尋的過程中，每利用移步搜尋一次新的鄰域解，不管此移步是否能求得更佳的解，迭代次數皆會累加下去；若當迭代次數大於初始設定的最大迭代次數 ( $M_{count}$ )，則停止禁忌搜尋法的搜尋，並以整個搜尋過程中出現最佳的解 ( $X^*$ ) 作為最終的解。



## 4. 實驗結果分析

### 4.1 作業環境

本研究使用 CPLEX 11.1.1 數學規劃軟體來求解混整數規劃的問題，而在各派工法則及求解改善機制等啟發式演算法則以 C++ 語言設計程式來求解；作業環境為 Windows XP，而硬體設備為 Intel(R) Pentium(R) D CPU 3.00GHz 以及 1.99GB 的記憶體。

### 4.2 模式資料設定

在建構模式前，需要收集現場相關的真實資料，來決定出合理的參數設定範圍。本研究以案例公司提供的機台資訊、搬運時間、加工種類、產品種類、加工流程與機台加工時間等資料為基礎，隨機產生數組測試資料，藉此來觀察各種排程機制的績效。

本研究的參數設定如下所示：

- i. 前工作站機台可開始處理時間： $U(10,20)$
- ii. 工件到達時間： $U(0,62.5)$
- iii. 工件到前工作站的搬運時間： $U(1,10)$
- iv. 前工作站工件加工時間： $U(20,30)$
- v. 後工作站機台可開始處理時間： $U(20,40)$
- vi. 工件到後工作站的搬運時間： $U(1,10)$
- vii. 後工作站批量加工時間： $U(30,40)$
- viii. 工件允許在後工作站等待加工的時間長度： $U(720,960)$
- ix. 工件數／機台數=4

### 4.3 衡量指標

由於 FSMP 排程是屬於 NP-Complete 的問題 (Garey and Johnson, 1979)，為了判斷本實驗的求解品質，本研究以 Jin *et al.* (2006) 的全域下界法為基礎，並針對本研究的問題特性（例如增加搬運時間與機台可開始處理時間）發展出適合的衡量指標。接著在符號說明的部份，我們使用  $LSA(i)$  以及  $RSA(i)$  來代表一個遞增的序列，其詳細說明如下：

**符號說明：**

$w$ ：機台數

$LSA(i)$ ：表示工件  $i$  在前工作站開始被加工之時刻下限，意指在 (1) 機台  $m$  之可開始加工時刻，以及 (2) 工件  $i$  到達前工作站的時刻加上搬運至機台  $m$  的時間，從這兩者之間取較大之時刻，再從所有機台中挑選該時刻最小之值，其值為  $\min_{m \in M_i} \{ \max \{ MT_m, r_i + t_{i,m} \} \}$ 。

$RSA(i)$ ：表示工件  $i$  在前工作站加工時間之後所必須經過的時刻，意指工件  $i$  搬運至機台  $m'$  的時間加上其於該機台的加工時間，再從所有機台中挑選時間最小的值，其值為  $\min_{m' \in M'_k} \{ P'_{i,m'} + t'_{i,m'} \}$ 。

$LSA(1)$ ：為  $LSA$  序列中的第一個值，其值為  $\min_{m \in M_1} \{ \max \{ MT_m, r_1 + t_{1,m} \} \}$ 。

$RSA(1)$ ：為  $RSA$  序列中的第一個值，其值為  $\min_{m' \in M'_k} \{ P'_{1,m'} + t'_{1,m'} \}$ 。

$LSA(i) = \{ LSA(1), LSA(2), \dots, LSA(n) \}$  為一個遞增的序列，其中  $LSA(1) \leq LSA(2) \leq \dots \leq LSA(n)$ 。

$RSA(i) = \{ RSA(1), RSA(2), \dots, RSA(n) \}$  為一個遞增的序列，其中  $RSA(1) \leq RSA(2) \leq \dots \leq RSA(n)$ 。

假設每道工作站都有  $w$  部平行機台的多機台流線式製程，若有  $n$  個工件要加工，

則總完工時間可用兩種下界  $LB_1$  以及  $LB_2$  來表示，其中  $LB_1$  代表以各階段工作站為基礎的下界估計值（其意義大概指將前後工作站各  $w$  部機台僅處理一個工件及批量之平均完成加工時刻），而  $LB_2$  代表以單一工件為基礎的下界估計值（其意義大概指若個別處理各工件時，最晚的工件之完成加工時刻），其定義分別如下所示：

$$C_{\max} \geq LB_1 = \frac{1}{w} \left( \sum_{i=1}^w LSA(i) + \sum_{i=1}^n P_{i,m} + \sum_{k=1}^w RSA(k) \right) \quad (4.1)$$

$$C_{\max} \geq LB_2 = \max_i \left\{ \min_{m \in M_i} \left\{ \max \{ MT_m, r_i + t_{i,m} \} + P_{i,m} + \min_{m' \in M'_i} \{ t'_{i,m'} + P'_{i,m'} \} \right\} \right\} \quad (4.2)$$

根據上面的方程式，則 FSMP 的總完工時間下界可表示成 LBMAX，定義如下：

$$LBMAX = \max \{ LB_1, LB_2 \} \quad (4.3)$$

本論文所使用之全域下界法乃以 Jin *et al.* (2006) 的方法為基礎，並增加搬運時間與機台可開始加工時間發展而得。然而，此全域下界法仍無法克服機台限制的特性，若大部分的工件只能在特定某幾部機台加工時，全域下界法所求得解會比實際所能求得的最佳解還低。接下來利用全域下界法所求得的下界與本研究所提出的 BFIFO 結合 TIT 三種改善求解機制之績效相比較，其全域下界法的值均明顯偏低（如圖 4.1），主要在於此全域下界法不考慮機台限制對求解品質的影響，因此全域下界法若能針對機台限制的影響進一步的探討，將能求得更適合的下界來與本研究所提出的方法做比較。

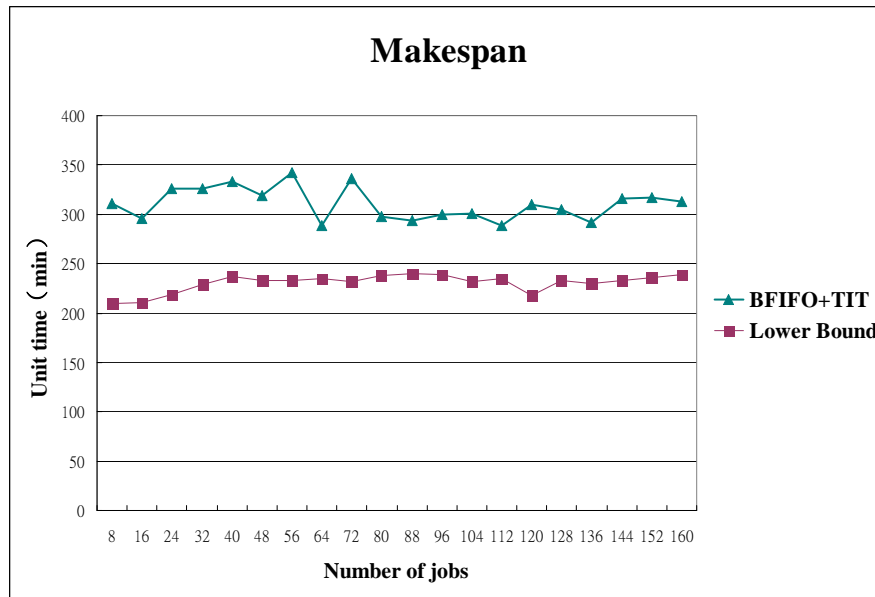


圖4.1 BFIFO 結合 TIT 三種改善求解機制與全域下界法之比較

## 4.4 實驗分析與討論

本研究擬測試九種排程機制，分別為 MIP、MIPH、FIFO、BFIFO、BFIFO+MIP、BFIFO+MIPH、BFIFO+TIT、TABU\_AE 以及 TABU\_PE，其中 MIP 代表原始之混整數規劃模型；MIPH 代表應用減少變數技巧後的混整數規劃模型；BFIFO 代表本研究所提出的 BFIFO 派工法則；BFIFO+MIP 代表 BFIFO 結合原始之混整數規劃模型；BFIFO+MIPH 代表 BFIFO 結合應用減少變數技巧後的混整數規劃模型；BFIFO+TIT 代表 BFIFO 結合 TIT 三種改善求解機制；TABU\_AE 代表使用鄰近交換法的禁忌搜尋法；TABU\_PE 代表使用兩兩交換法的禁忌搜尋法。

### 4.4.1 實驗結果分析

本研究將測試二十群不同問題規模的例子，各測試群皆隨機產生 10 筆參數不同之測試例，使用各種求解機制來測試求解，並將其結果取平均畫出折線圖（圖 4.2 至圖 4.8）來比較。

由於 MIP 於 56 個工件以後即無法在 10 分鐘內求得可行解，因此我們僅用 MIP 與 MIPH 來求解問題規模為 4 到 56 個工件之測試例。由圖 4.2 得知數學規劃模式（MIP、MIPH）耗時最多，而 BFIFO+MIPH 又比 BFIFO+MIP 有效率，至於其它的派工法則、啟發式演算法以及禁忌搜尋法皆可快速求解。

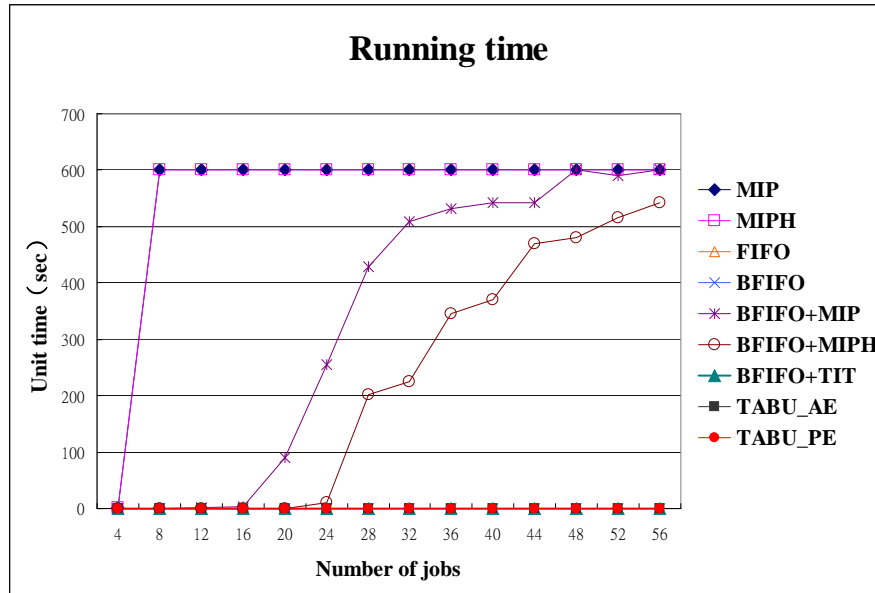


圖4.2 九種求解機制之時間

在求解品質方面，針對 4 到 56 個工件規模的測試例而言，圖 4.3 顯示大部分的啟發式演算法皆可得到不錯且品質穩定的排程（但未必最佳）；雖然 MIP 及 MIPH 旨在求出最佳排程，然而當測試例規模變大之後，其在時限內所能獲得之可行解反而品質變差，且當 MIPH 在 60 個工件規模的測試例以後，即無法於 10 分鐘內求得可行的解。

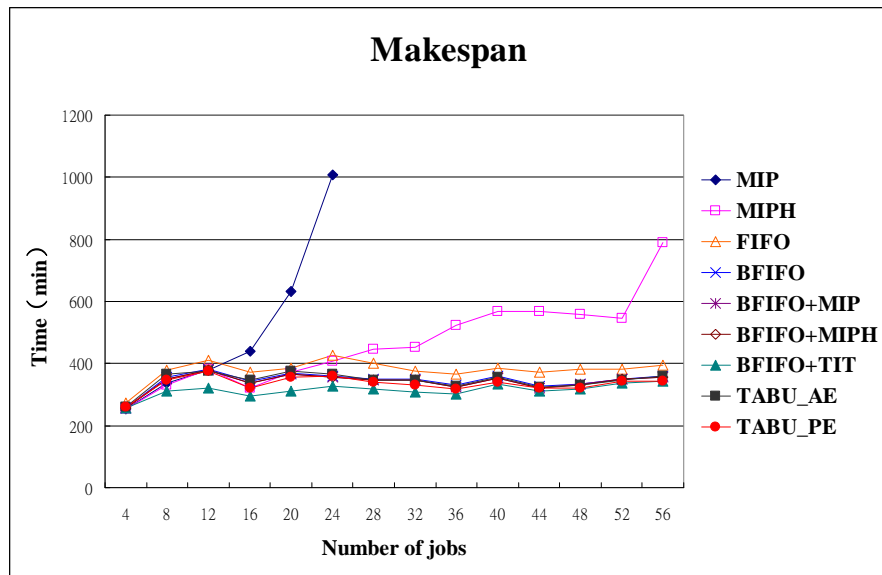


圖4.3 九種求解機制之總完工時間

針對更大規模（譬如工件數大於 56）的問題，我們僅能使用七種啟發式演算法求解之。如圖 4.4 所示，只要用到 MIP 或 MIPH 的解法皆會耗時較久，BFIFO+MIP 演算法在 48 個工件規模之後以及 BFIFO+MIPH 演算法在 64 個工件規模之後，求解時間均需要在設定 10 分鐘的終止條件才會停止，而 MIPH 也的確較 MIP 更具求解較大規模問題的能力。其它五種求解機制即使在最大的工件規模之中，仍然可以快速的求解。

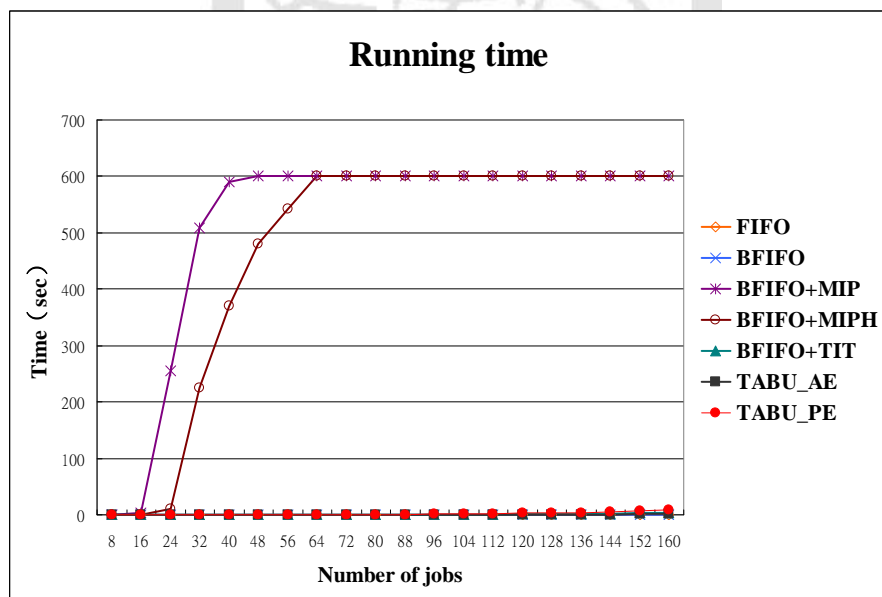


圖4.4 七種啟發式演算法之求解時間

在求解品質方面，由圖 4.5 可得知，除了傳統的 FIFO 派工法則所求得的可行解相對於其他求解機制明顯較差外，其他的求解機制相較於 FIFO 派工法則的求解品質都有顯著地改善。至於本研究所提出的 BFIFO+TIT 解法可得到七種啟發式演算法中品質最好的解。

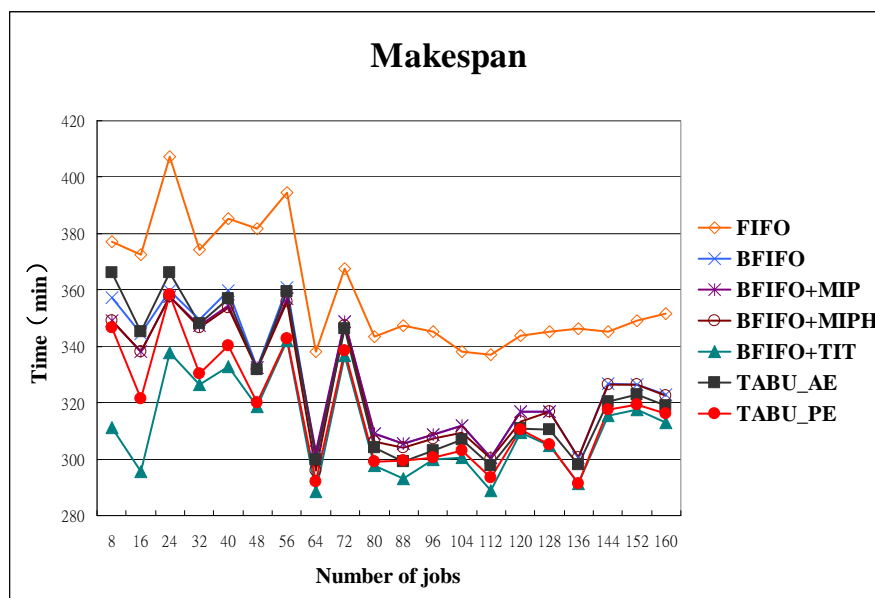


圖4.5 七種啟發式演算法之總完工時間

接著觀察 BFIFO、BFIFO+MIP 以及 BFIFO+MIPH 之總完工時間，如圖 4.6 所示，可看出 BFIFO+MIP 在小規模的問題中能比 BFIFO 求得較佳的可行解，在大於 64 個工件規模之後，即無法在時限內有效改善 BFIFO 的起始解。而 BFIFO+MIPH 在 128 個工件規模之後，亦無法在時限內有效改善 BFIFO 的起始解，因此使用 MIP 與 MIPH 的求解機制，在更大規模的例子中，僅能維持 BFIFO 的起始解，無法有效得到更佳的可行解。

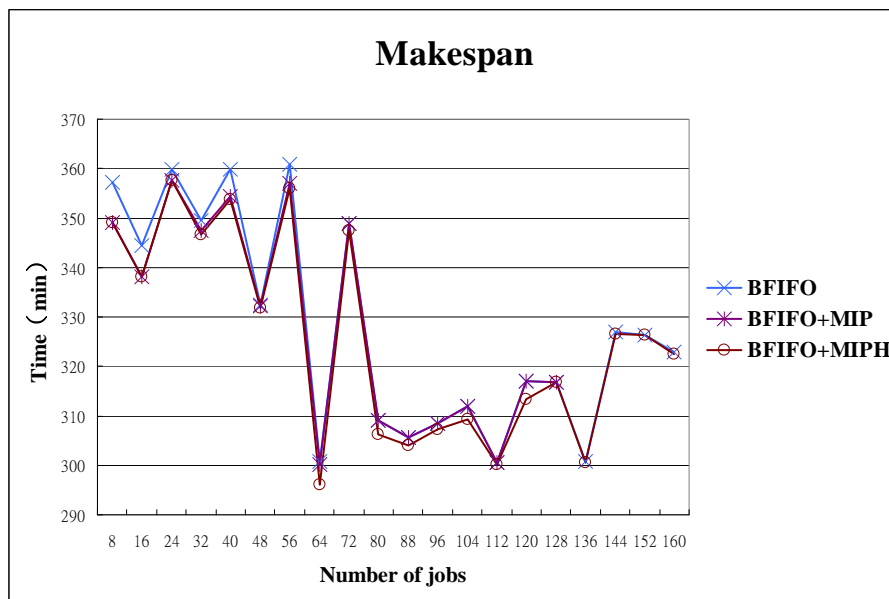


圖4.6 BFIFO、BFIFO+MIP 以及 BFIFO+MIPH 之總完工時間

若再進一步分析 BFIFO+TIT 所改善的效率，我們可由圖 4.7 得知其實 BFIFO+TIT 耗時甚少，它能在 3 秒以內求解如 160 個工件的大規模問題。而禁忌搜尋法的求解時間主要取決於迭代次數的大小，當給定某個長度為  $q$  的順序時，所有兩兩交換的可能為  $q(q-1)/2$ ，並令  $q(q-1)/2$  為  $Q$ ，而本研究設定的最大迭代次數為  $2Q$ ，因此當規模越大時，迭代的次數越多，也使得求解時間相對較長。

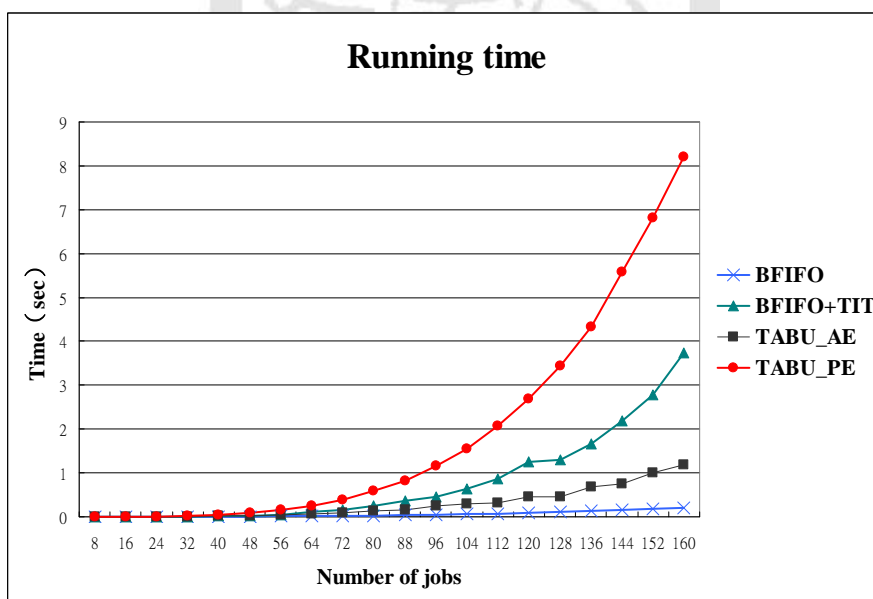


圖4.7 BFIFO、BFIFO+TIT 與禁忌搜尋法之求解時間比較



針對 BFIFO+TIT 所改善的效能，我們可由圖 4.8 得知，BFIFO+TIT 在 BFIFO 的初始解為基礎之下，再進一步的利用 TIT 三種改善機制，所求得解均有明顯的改善。至於 BFIFO+TIT 與兩種交換方式的禁忌搜尋法比較，由於鄰近交換法的方式，僅考慮鄰近的工件交換，因此無法有效得到較佳的求解品質；而兩兩交換的方式，則是任選兩個工件進行交換，此方式所求得的求解品質僅次於 BFIFO+TIT 的求解品質，然而本研究僅針對 40 個工件的問題規模來尋找禁忌搜尋法最佳的參數設定，若能在其它的問題規模大小中尋找最佳參數設定，或許能求得更加的可行解。

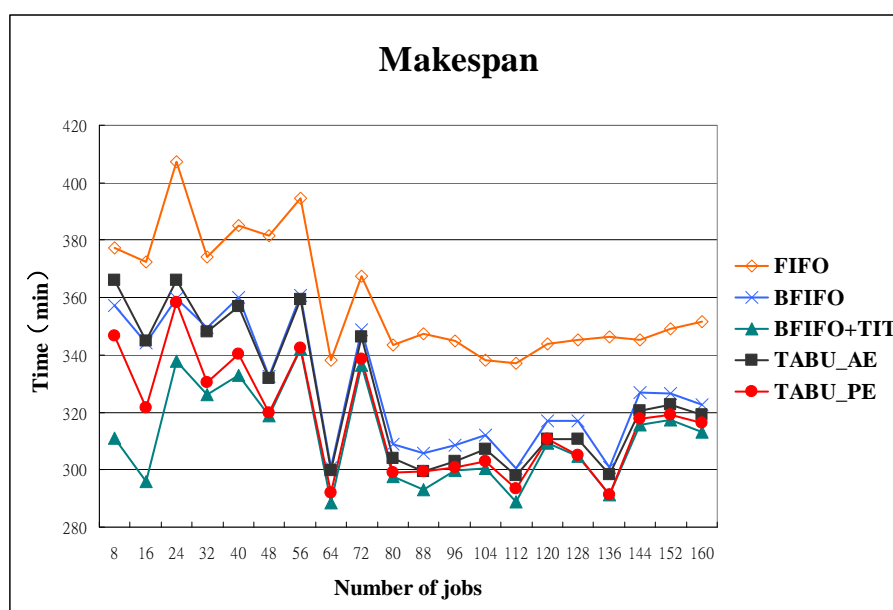


圖4.8 BFIFO、BFIFO+TIT 與禁忌搜尋法之總完工時間比較

接著挑選出五種不同問題規模大小，來觀察此九種改善機制之效能的表現，如表 4.1 所示，表中的 32-8-8 代表有 32 個工件且前後工作站各有 8 部機台的問題規模，從表中的數據可看出，MIP 與 MIPH 在小規模的例子中皆會耗時較久，且在 64 個工件之後，即無法於 10 分鐘內求得可行解。以混整數規劃為基礎的求解機制 (BFIFO+MIP 與 BFIFO+MIPH)，雖然能夠求解更大規模的問題，但仍然需要耗費很長的求解時間。其餘五種的改善機制，由於沒有利用到數學規劃的方法，因此都能快速的求解。

表4.1 求解時間之數據比較

求解機制	Running Time (sec)					Average
	32-8-8	64-16-16	96-24-24	128-32-32	160-40-40	
<b>MIP</b>	600.0312	-	-	-	-	600.0312
<b>MIPH</b>	600.0201	-	-	-	-	600.0201
<b>FIFO</b>	0.0016	0.0047	0.0125	0.0266	0.0469	0.01846
<b>BFIFO</b>	0.0016	0.0016	0.0032	0.0032	0.0063	0.00318
<b>BFIFO1+MIP</b>	507.9137	600.0703	600.2107	600.4213	600.1024	581.7437
<b>BFIFO1+MIPH</b>	225.2372	600.0201	600.0316	600.0402	600.0405	525.0739
<b>BFIFO+TIT</b>	0.0095	0.1063	0.4656	1.3016	3.7438	1.1254
<b>TABU_AE</b>	0.0079	0.0672	0.2484	0.4641	1.1844	0.3944
<b>TABU_PE</b>	0.0188	0.2547	1.1531	3.4422	8.1969	2.6131

除了 MIP 與 MIPH 外，針對另外七種求解機制在五種不同問題規模大小的總完工時間之比較如表 4.2 所示，BFIFO+MIP 與 BFIFO+MIPH 在求解較小規模的排程問題時，均能得到比 BFIFO 派工法則更好的求解品質，而 BFIFO+MIPH 在所有的問題規模中都能夠得到比 BFIFO+MIP 更好的求解品質。BFIFO 法則所求得之解皆較傳統的 FIFO 法則所求得之解有更佳之品質。TABU\_AE 能求得比 BFIFO 較佳的效能，但受限於交換的方式，因此求解品質的改善仍然有限。TABU\_PE 的效能相較於 FIFO 大約改善了 11.97%。BFIFO+TIT 相較於其他六種求解機制，所求得的效能表現最佳，且相較於 FIFO 大約改善了 12.66%。

表4.2 總完工時間之數據比較

求解機制	Makespan (min)					Improvement
	32-8-8	64-16-16	96-24-24	128-32-32	160-40-40	vs. FIFO
<b>FIFO</b>	374.24	338.09	345.06	345.34	351.72	0.00 %
<b>BFIFO</b>	349.41	300.78	308.54	316.90	322.81	8.89 %
<b>BFIFO+MIP</b>	347.36	300.11	308.54	316.90	322.81	8.99 %
<b>BFIFO+MIPH</b>	346.69	296.01	307.22	316.76	322.48	9.42 %
<b>BFIFO+TIT</b>	326.27	288.54	299.77	304.79	312.99	12.66 %
<b>TABU_AE</b>	347.95	299.86	302.90	310.55	319.10	9.92 %
<b>TABU_PE</b>	330.43	291.89	300.70	305.11	316.26	11.97 %

#### 4.4.2 敏感度分析

由於先前的測試是以工件數大約為機台數 4 倍的情況為基礎，為理解 TIT 改善機制與禁忌搜尋法對不同的工件/機台比例之排程表現，本研究在此節利用敏感度分析來測試工件數與機台數的比值分別為 8、16、24、32 以及 40 等 5 種情境（如表 4.3）下，觀察 FIFO、BFIFO、BFIFO+TIT 與兩種交換方式的禁忌搜尋法對於效能與效率的影響。

表4.3 敏感度分析表

工件數÷機台數	原始設定	敏感度分析				
	4	8	16	24	32	40
工件數	160	160	160	160	160	160
機台數	40	20	10	7	5	4

此五種機制求解時間的表現如圖 4.9 所示，其中 FIFO 與 BFIFO 的效率仍然是最好；在禁忌搜尋法方面，使用 TABU\_PE 仍然是五種求解機制中耗時較長的，而使用 TABU\_AE 依舊能夠快速求解；而在 BFIFO+TIT 的表現上，其效率仍然介於兩種交換方式的禁忌搜尋法間，並不會因為工件數與機台數的比值增加，而導致求解時間過長的情形。整體而言，工件數與機台數的比值大小對此五種求解機制的效率並沒有造成太大的影響，此結果顯示我們的求解機制還算有不錯的強健性（Robustness）。

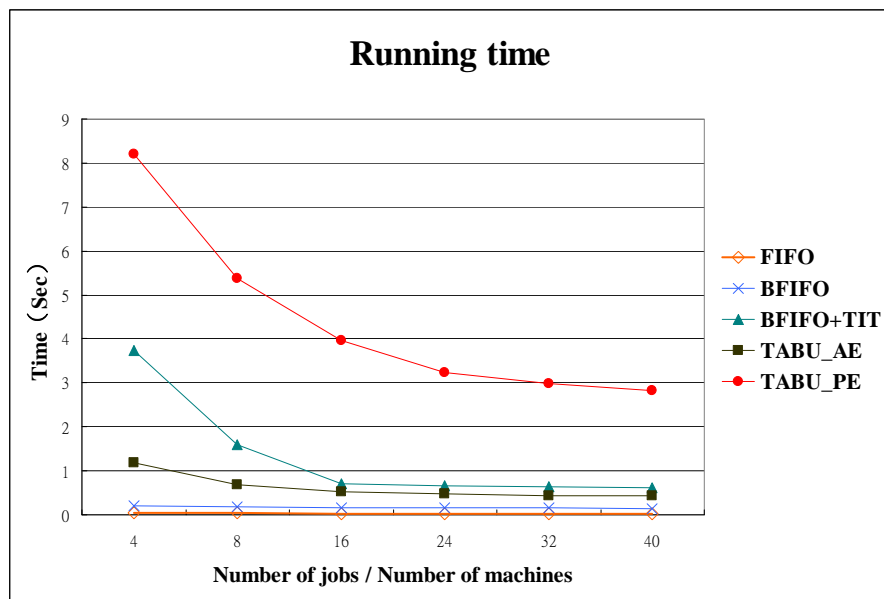


圖4.9 FIFO、BFIFO、BFIFO+TIT 與禁忌搜尋法之求解時間比較

在求解品質方面，由圖 4.10 可看出在工件數與機台數比值大於或等於 24 時，BFIFO+TIT 所得之可行解的品質明顯較佳；傳統的 FIFO 派工法則所求得解仍然最差，而其他三種機制的效能差異並不明顯。由圖可看出本研究所提出的 BFIFO 派工法則結合 TIT 求解機制，在工件數與機台數的比值更大時，仍然可以即時得到四種求解機制中品質最好的解，且隨著工件數與機台數的比值愈大，所改善的效能明顯愈好。

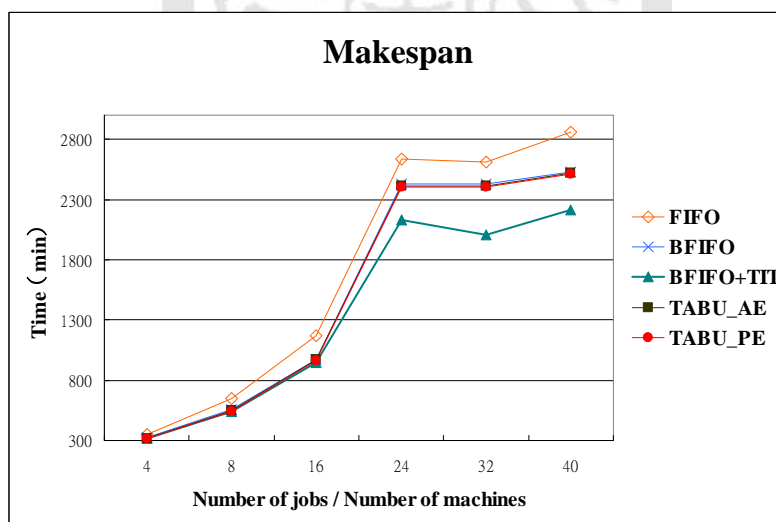


圖4.10 FIFO、BFIFO、BFIFO+TIT 與禁忌搜尋法之總完工時間比較

針對此五種求解機制對於求解時間數據之比較，由表 4.4 的數據整理來看，在工

件數與機台數比值不同的情況下，除了 TABU\_PE 耗費的時間稍微較長外，FIFO、BFIFO、BFIFO+TIT 與 TABU\_AE 平均都能在 2 秒以內求解。

表4.4 求解時間之數據比較（敏感度分析）

工件數÷機台數	Running time (sec)						Average
	4	8	16	24	32	40	
<b>FIFO</b>	0.0469	0.0328	0.0316	0.0263	0.0249	0.0198	0.0304
<b>BFIFO</b>	0.2125	0.1735	0.1563	0.1484	0.1500	0.1406	0.1636
<b>BFIFO+TIT</b>	3.7438	1.6047	0.7101	0.6719	0.6328	0.6194	1.3305
<b>TABU_AE</b>	1.1844	0.6922	0.5282	0.4688	0.4407	0.4234	0.6230
<b>TABU_PE</b>	8.1969	5.3719	3.9563	3.2422	2.9750	2.8203	4.4271

針對此五種求解機制在敏感度分析的總完工時間之比較，如表 4.5 所示，BFIFO 法則所求得之解仍比傳統的 FIFO 法則所求得之解有更佳之品質。TABU\_AE 亦能求得比 BFIFO 較佳的效能，但受限於交換的方式，求解品質的改善仍然有限。TABU\_PE 在工件數與機台數比值愈大的情況下，雖然都比 TABU\_AE 求得較佳的效能，但並無明顯的改善。BFIFO+TIT 相較於其他四種求解機制，在工件數與機台數比值愈大的情況下，此機制的效能表現最佳且改善幅度更為明顯，平均而言，相較於 FIFO 的效能大約改善了 20.73%，且相較於 TABU\_PE 的效能亦大約改善了 9.69%。

表4.5 總完工時間之數據比較（敏感度分析）

工件數÷機台數	Makespan (min)						Improvement
	4	8	16	24	32	40	vs. FIFO
<b>FIFO</b>	351.72	649.31	1172.87	2632.52	2609.00	2863.34	0.00 %
<b>BFIFO</b>	322.81	553.92	974.63	2426.19	2425.24	2526.16	10.21 %
<b>BFIFO+TIT</b>	312.99	541.01	947.21	2126.67	2004.86	2215.59	20.73 %
<b>TABU_AE</b>	319.10	546.27	968.53	2407.84	2413.14	2523.33	10.71 %
<b>TABU_PE</b>	316.26	544.13	965.81	2401.85	2407.43	2508.95	11.04 %

## 4.5 小結

由測試結果我們可得到以下觀察：

- (1) 在混整數規劃模型方面，由於 MIPH 主要是依照機台所能處理的工件數來判斷機台所需的順序數，因此當加工時間差異很大時，其求解品質在處理小規模（16 個工件以內）的排程問題時可能會比 MIP 差；在處理 16 至 56 個工件的排程問題時，MIPH 之求解品質又優於 MIP。此外，在 10 分鐘以內，MIPH 最多可求出 56 個工件的排程問題可行解，而 MIP 最多僅可求得 24 個工件的可行排程。
- (2) 在派工法則方面，BFIFO 法則所求得之解皆較傳統的 FIFO 法則所求得之解有更佳之品質，由於派工法則主要的方式是依照工件的排列順序依序將工件安排至可最快完工的機台上加工，此方式雖然能夠快速得到好的解，但無法預測機台限制對績效所造成的影響，因此求解品質仍有很大的改善空間。
- (3) 在 BFIFO 派工法則結合混整數規劃模型方面，BFIFO+MIP 在求解 64 個工件以內的排程問題與 BFIFO+MIPH 在求解 128 個工件以內的排程問題時，均能得到比 BFIFO 派工法則更好的求解品質；然而，BFIFO+MIP 在求解 64 個工件以上之大規模的排程問題時，即無法在時限內有效改善 BFIFO 的起始解；至於 BFIFO+MIPH 通常可求得比 BFIFO+MIP 更好的排程，但在更大規模的排程問題（128 個工件以上），亦無法在時限內有效改善 BFIFO 的起始解。
- (4) 在禁忌搜尋法方面，鄰近交換方式能求得比 BFIFO 派工法則較佳的求解品質，但由於只考量鄰近的工件進行交換，因此求解品質的改善仍然有限；而兩兩交換方式亦能得到較佳的求解品質，若在最佳的參數設定之下，或許能得到更好的效能，但針對不同的問題規模大小來測試最佳的參數設定將耗費許多時間成本且求解時間比 BFIFO+TIT 長，因此在時間成本的考量之下，BFIFO+TIT 相較於禁忌搜尋法均能夠在更短的時間內得到較佳的效能。

(5) 在 BFIFO 派工法則結合三種改善求解機制方面，BFIFO+TIT 的求解機制在所有的問題規模大小中，相較於能夠求得鄰近最佳解的禁忌搜尋法，均能夠在更短時間內（4 秒內）求得最佳的可行解。在敏感度分析中，工件數與機台數比值愈大的情況下，BFIFO+TIT 的效能表現最佳且改善幅度更為明顯。

總結本研究所提的排程機制，以混整數規劃模型為基礎的機制，均會有遇到求解時間過長的困擾；相對之下，FIFO 派工法則與 BFIFO 派工法則均能夠快速得到可行的解，但由於受到機台限制的影響，因此其求解品質仍有很大的改善空間；在鄰近交換方式的禁忌搜尋法方面，受限於挑選機制的影響，因此改善的空間有限；而兩兩交換方式的禁忌搜尋法，必須耗費時間找到一組最佳的參數設定，才有機會得到較佳的求解品質，而禁忌搜尋法亦無法比 BFIFO 派工法則結合 TIT 的改善求解機制於更短的時間內求得較佳的求解品質；本研究所提出的 BFIFO 派工法則結合 TIT 改善求解機制演算法，不但求解時間極短，且其求解品質亦極佳。

## 5. 結論與建議

### 5.1 結論

在市場競爭激烈的環境，為了滿足客戶的需求，除了有先進的晶圓製造技術外，發展即時性生產排程系統，提昇客戶的達交率更是維持國際化經營的競爭力。在此觀念下，本研究之發展目標在於：發展出以即時性生產排程為核心的系統，建立一合理且有效的排程模式，以提供晶圓代工廠能進行即時性生產規劃活動。

本研究以 12 吋晶圓廠中現場實際所面臨的兩階段排程問題為主要的研究對象，提出了數種的求解方法及其改善求解機制。首先針對此排程問題建構一混整數規劃模型，然而混整數規劃模型在給定的時限內，僅能處理小規模（不超過 24 個工件）的問題，因此我們又提出一種技巧來減少混整數規劃變數之規模。雖然此技巧的確能在時限內解決更大規模的問題，然而此法仍將耗費不少時間。

為了因應現實環境中即時排程的需求，我們以能在 10 分鐘內將 160 個工件於兩階段各 40 部機台的製程順序排好為目標，因此先捨棄混整數規劃的解法，改以使用考量成批加工的 BFIFO 派工法則先快速規劃一個排程，再進一步提出數種改善求解機制以調整改進所得排程之品質。其中，數值測試結果可證實 BFIFO 派工法則的確比傳統的 FIFO 派工法則在處理批量及不同機台限制的排程問題時，更能夠求得具較佳品質之排程；然而，受機台限制的影響，因此 BFIFO 派工法則之效能仍有很大的改善空間。

在改善求解機制方面，我們先提出 BFIFO+MIP 以及 BFIFO+MIPH 兩種結合，旨在縮小原本混整數規劃模型的求解範圍，BFIFO+MIP 求解較小規模（64 個工件以內）的排程問題時，均能得到比 BFIFO 法則更好的求解品質；BFIFO+MIPH 通常可求得比 BFIFO+MIP 更好的排程，且相較於 BFIFO+MIP 能在較大規模（128 個工件以內）的排程問題，得到比 BFIFO 法則更好的求解品質。以混整數規劃模型為基礎的方法，於大規模問題中不僅有求解時間過長的困擾，亦無法有效改善以 BFIFO 為起始解的求解



品質，且派工法則為基礎的方式受機台限制的影響無法有效的改善效能，因此本研究又提出了 BFIFO 派工法則結合 TIT 改善機制，此機制在所有的問題規模大小中，均能夠在短時間內求得最佳的可行解。

後續我們將本研究所提出之混整數規劃法、BFIFO 派工法則、BFIFO 派工法則結合混整數規劃模型以及 BFIFO 派工法則結合 TIT 改善機制與 Yang *et al.* (2004) 所提出的禁忌搜尋法來進行不同問題規模大小之效能與效率的比較以及敏感度分析，以驗證本研究所提出之求解品質與效率。實驗結果顯示，本研究所提的 BFIFO 派工法則結合 TIT 改善機制相較於 Yang *et al.* (2004) 所提出之禁忌搜尋法，大都能在較短的時間得到較佳的排程結果，且在敏感度分析的實驗結果，BFIFO 派工法則結合 TIT 改善機制相較於 Yang *et al.* (2004) 之禁忌搜尋法的求解速度較快，且效能平均改善了 9.69%。因此本研究建議相關產業所遇到的排程問題或可嘗試使用 BFIFO 派工法則結合 TIT 改善機制來解決其即時排程的需求。

## 5.2 未來研究與建議

本研究在發展的過程中仍有未盡完善之處，最後提出以下建議，以供日後相關研究發展之參考：

- (1) 在實驗的結果中，雖然 BFIFO 派工法則結合混整數規劃模型的求解機制在減少變數後亦能夠解決本研究的問題規模，但求解品質於問題規模較大的例子中已不如預期，未來或許能夠推導出更多有效的不等式 (Valid Inequality) 以刪去求解混整數規劃模型時的線性解，來提升求解的效率。
- (2) 本研究所提出的 BFIFO 派工法則結合 TIT 的求解機制在此排程問題中，具有即時性且能有效改善求解品質，若將此機制應用於不同的排程問題中，例如考慮換模時間等等，或許也能得到不錯的績效，並與相關文獻所提出的方法來做比較，可以在未來的研究中進一步去探討。

- (3) 本研究後續可使用模擬最佳化的方法來進行效能的測試，例如使用模擬軟體 Arena 中內建的模擬最佳化軟體 (OptQuest)，主要執行方式為將求解資訊輸入至模擬模式中，再利用 OptQuest 在模擬模式中自動地搜尋最佳解。可藉由此模擬最佳化的方法來求得鄰近最佳解並與本研究所提出的方法做比較。
- (4) 本研究在文獻的搜尋過程中，部分文獻提到基因演算法在參數組合的問題上能有不錯的表現，因此後續亦可嘗試建立基因演算法來決定工件將安排至哪部機台或利用基因演算法來進行機台中工件的位置互換，利用上述的方式或許也能得到不錯的績效。
- (5) 本研究目前僅提供案例公司排程方法之方法論，與實際排程系統進行整合的部份，仍需交由專業的人士來發展出製造執行系統或者與先進排程系統進行整合，以期未來在實際的應用上，提供給決策者整合分析之用。



## 參考文獻

- 郭宜雍，民 94，以模擬為基礎之啟發式演算法求解平行機台排程問題，國立成功大學製造工程研究所，博士論文。
- Al-Anzi, F.S. and Allahverdi, A., 2009, Heuristics for a two-stage assembly flowshop with bicriteria of maximum lateness and makespan, *Computers and Operations Research*, **36**, 2682-2689.
- Allaoui, H. and Artiba, A., 2006, Scheduling two-stage hybrid flow shop with availability constraints, *Computers and Operations Research*, **33**, 1399-1419.
- Armentano, V.A. and Ronconi, D.P., 1999, Tabu search for total tardiness minimization in flowshop scheduling problems, *Computers and Operations Research*, **26**, 219-235.
- Bellanger, A. and Oulamara, A., 2009, Scheduling hybrid flowshop with parallel batching machines and compatibilities, *Computers and Operations Research*, **36**, 1982-1992.
- Ben-Daya, M. and Al-Fawzan, M., 1998, A tabu search approach for the flowshop scheduling problem, *European Journal of Production Research*, **35**, 2857-2870.
- Botta-Genoulaz, V., 2000, Hybrid flowshop scheduling with precedence constraints and time lags to minimize maximum lateness, *International Journal of Production Economics*, **64**, 101-111.
- Centeno, G. and Armacost, R.L., 1997, Parallel machine scheduling with release time and machine eligibility restrictions, *21<sup>st</sup> International Conference on Computers and Industrial Engineering*, **33**, 273-276.
- Centeno, G. and Armacost, R.L., 2004, Minimizing makespan on parallel machines with release time and machine eligibility restrictions, *International Journal of Production Research*, **42**, 1243-1256.
- Chen, B. and Vestjens, A.P.A., 1997, Scheduling on identical machines: How good is LPT

- in an on-line setting?, *Operation Research Letters*, **21**, 165-169.
- Dobson, G. and Nambimadom, R.S., 2001, The batch loading and scheduling problem, *Operations Research*, **49**, 52-65.
- Garey, M.R. and Johnson, D.S., 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York.
- Glover, F., 1989, Tabu Search – Part I, *ORSA Journal on Computing*, **1**, 190-206.
- Glover, F., 1990, Tabu Search – Part II, *ORSA Journal on Computing*, **2**, 4-32.
- Glover, F. and Kochenberger, G.A., 2003, *Handbook of Metaheuristics*, Kluwer Academic Publishers, London.
- Grangeon, N., Tanguy, A., and Tchernev, N., 1999, Generic simulation model for hybrid flow-shop, *Computers and Industrial Engineering*, **37**, 207-210.
- Gupta, J.N.D., 1988, Two stage, hybrid flowshop scheduling problem, *Journal of Operations Research Society*, **39**, 359-364.
- Gupta, J.N.D., Hariri, A.M.A., and Potts, C.N., 1997, Scheduling a two-stage hybrid flow shop with parallel machines at the first stage, *Annals of Operations Research*, **69**, 171-191.
- Gupta, J.N.D., Kruger, K., Lauff, V., Werner, F., and Sotskov, Y.N., 2002, Heuristics for hybrid flow shops with controllable processing times and assignable due dates, *Computers and Operations Research*, **29**, 1417-1439.
- Gupta, J.N.D. and Ruiz-Torres, A.J., 2005, Generating efficient schedules for identical parallel machines involving flow-time and tardy jobs, *European Journal of Operation Research*, **167**, 679-695.
- Haouari, M. and M'Hallah, R., 1997, Heuristic algorithms for the two-stage hybrid flow shop problem. *Operations Research*. **69**, 171-191.
- Hoogeveen, J.A., Lenstra, J.K., and Veltman, B., 1996, Preemption scheduling in a

- two-stage multiprocessor flowshop is NP-hard, *European Journal of Operational Research*, **89**, 172-175.
- Horowitz, E. and Sahni, S., 1976, Exact and approximate algorithm for scheduling nonidentical processors, *Journal of the Association for Computing Machinery*, **23**, 317-327.
- Hung, Y.F., 1998, Scheduling of mask shop E-beam writers, *IEEE Transactions on Semiconductor Manufacturing*, **11**, 165-172.
- Hunsucker, J.L. and Shah, J.R., 1994, Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment, *European Journal of Operational Research*, **72**, 102-114.
- Jayamohan, M.S. and Rajendram, C., 2000, A comparative analysis of two different approaches to scheduling in flexible flow shop, *Production Planning and Control*, **11**, 572-580.
- Jin, Z., Yang, Z., and Ito, T., 2006, Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem, *International Journal of Production Economics*, **100**, 322-334.
- Johnson, S.M., 1954, Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, **1**, 61-68.
- Kashan, A.H., Karimi, B., and Jenabi, M., 2008, A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes, *Computers and Operations Research*, **35**, 1084-1098.
- Kelton, W.D., Sadowski, R.P., and Sturrock, D.T., 2007, *Simulation with Arena*, 4<sup>th</sup> edition, McGraw-Hill, New York.
- Kim, T.D., Kim, J.U., Lim, S.K., and Jun, H.B., 1998, Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility, *IEEE*

- Transactions on Semiconductor Manufacturing*, **11**, 155-164.
- Kuo, Y., Yang, T., Peters, B.A., and Chang, I., 2007, Simulation metamodel development using uniform design and neural networks for automated material handling systems in semiconductor wafer fabrication, *Simulation Modelling Practice and Theory*, **15**, 1002-1015.
- Lee, C.-Y. and Vairaktarakis, G.L., 1994, Minimizing makespan in hybrid flowshops, *Operations Research Letters*, **16**, 149-158.
- Lin, S.W. and Ying, K.C., 2009, Applying a hybrid simulated annealing and tabu search approach to non-permutation flowshop scheduling problems, *International Journal of Production Research*, **5**, 1411-1424.
- Malve, S. and Uzsoy R., 2007, A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families, *Computer and Operations Research*, **34**, 3016-3028.
- Mathirajan, M., Sivakumar, A.I., and Chandru, V., 2002, Scheduling algorithms for heterogeneous batch processors with incompatible job-families, *Proceeding of the 2<sup>nd</sup> International Conference on Responsive Manufacturing, Gaziantep, Turkey*, 769-774.
- Melouk, S., Damodaran, P., and Chang, P.Y., 2004, Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing, *International Journal of Production Economics*, **87**, 141-147.
- Min, H.S. and Yih, Y., 2003, Development of a real-time multi-objective scheduler for a semiconductor fabrication system, *International Journal of Production Research*, **41**, 2345-2364.
- Monch, L., Balasubramanian, H., Fowler, J.W., and Pfund, M.E., 2005, Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times, *Computers and Operations Research*, **32**, 2731-2750.

- Mosheiov, G. and Oron, D., 2004, A note on the SPT heuristic for solving scheduling problem with generalized due dates, *Computers and Operations Research*, **31**, 645-655.
- Osman, I.H., *Metastratesy Simulated Annealing and Tabu Search Algorithms for Combinatorial Optimization Problems*, United Kingdom : Philosophy of the University of London.
- Oulamara, A., 2007, Makespan minimization in a no-wait flow shop problem with two batching machines, *Computers and Operations Research*, **34**, 1033-1050.
- Oulamara, A., Finke, G., and, Kamgaing Kuiten, A., 2009, Flowshop scheduling problem with batching machine and task compatibilities, *Computers and Operations Research*, **36**, 391-401.
- Petroni, A. and Rizzi, A., 2002, A fuzzy logic based methodology to rank shop floor dispatch rules, *International Journal of Production Economics*, **76**, 99-108.
- Pinedo, M., 1995, Scheduling — Theory, algorithms, and systems, Prentice-Hall, *Englewood Cliffs*, New Jersey.
- Santos, D.L., Hunsucker, J.L., and Deal, D.E., 1995, Global lower bounds for flow shop with multiple processors, *European Journal of Operational Research*, **80**, 112-120.
- Santos, D.L., Hunsucker, J.L., and Deal, D.E., 1996, An evaluation of sequencing heuristics in flow shops with multiple processors, *Computers and Industrial Engineering*, **30**, 681-692.
- Soewandi, H., and Elmaghraby, S.E., 2001, Sequencing three-stage flexible flowshops with identical machines to minimize makespan, *IIE Transactions*, **33**, 985-993.
- Uzsoy, R., 1995, Scheduling batch processing machine with incompatible job-families, *International Journal of Production Research*, **33**, 2685-2708.
- Uzsoy, R., Church, L.K., and Ovacik, I.M., 1992, Dispatching rules for semiconductor testing operations: a computational study, *IEEE/CHMT International Electronic*

*Manufacturing Technology Symposium*, 272-276.

Van der Velde, S.L., 1993, Duality based algorithm for scheduling unrelated parallel machines, *ORSA Journal on Computing*, **5**, 192-205.

Wardono, B. and Fathi, Y., 2004, A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities, *European Journal of Operational Research*, **155**, 380-401.

Yang, T., Kuo, Y., and Chang, I., 2004, Tabu-search simulation optimization approach for flow-shop scheduling with multiple processors — a case study, *International Journal of Production Research*, **42**, 4015-4030.

Yang, T., Rajasekharan, M. and Peters, B.A., 1999, Semiconductor fabrication facility design using a hybrid search methodology, *Computers and Industrial Engineering*, **36**, 565-583.

