

國立交通大學
工業工程與管理學系

碩士論文

應用啟發式演算法與家族式派工於多部傳
輸整合步進機在小批量情境下之排程問題



Family-based meta-heuristic algorithms for
scheduling multiple in-line steppers in
small-lot scenarios

研究生：謝佩芸

指導教授：巫木誠教授

中華民國九十九年六月

應用啟發式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

**Family-based meta-heuristic algorithms for
scheduling multiple in-line steppers in small-lot scenarios**

研究生：謝佩芸

Student：Pei-Yun Hsieh

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



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 2010

Hsin-chu, Taiwan, Republic of China

中華民國九十九年六月

應用啟發式演算法與家族式派工於多部傳輸整合步進 機在小批量情境下之排程問題

研究生：謝佩芸

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

半導體工廠中，傳輸整合步進機為瓶頸機台，其在小批量的情況下，會因為機台埠區的限制造成產能損失之問題，而過去文獻中已對此問題作過研究，但和現實狀況比較，其假設過於簡化，因此本研究有三項假設：第一，多部傳輸整合步進機。第二，不同的電路佈局之工件，進入傳輸整合步進機加工，當更換工件時，會產生更換光罩的設置時間。第三，使用相同光罩之工件會將之排序在一起，即家族式派工法(Family-based)，以減少設置時間之發生，故在求解多機之工件排程問題上，工件會面臨三項重要之決策，分別為工件族需要分配給哪一個機台(Family Assignment)，以及工件族在各別機台內的排序問題(Among Family Sequence)，工件族內各工件的順序 (Within Family Sequencing)，而本研究提出七種演算法，經由多次的實驗後，顯示GA-Tabu方法優於其他的演算法，此外，比較家族式派工法與單獨派工法於此排程規劃之績效，更證明家族式派工法則之必要性。

關鍵詞：排程、啟發式演算法、家族式派工排程、傳輸整合步進機

Family-based meta-heuristic algorithms for scheduling multiple in-line steppers in small-lot scenarios

Student : Pei-Yun Hsieh

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

An in-line stepper, a bottleneck machine in a semiconductor fab, may have capacity loss in a small-lot scenario. Prior studies have examined scheduling methods to reduce such a capacity loss. Yet, their contexts of study are relatively simplified, in contrast to that of real-world cases. This research aims to examine scheduling methods in a more general context—with three distinct features. First, there are multiple in-line steppers. Second, jobs that require the same setup are grouped into a family in scheduling. Thirdly, a setup is required while switching to process different families and not required while processing jobs within the same family. Such a context leads to three scheduling decisions: *job-family assignment* (how to assign jobs families to in-line steppers), *among-family sequencing* (how to determine the sequence among families), and *within-family sequencing* (how to determine the job sequence within each family). Seven meta-heuristic algorithms are developed. Numerical experiments indicate that the GA-Tabu algorithm outperforms the other six ones in most test cases. Moreover, experiments also indicate that such a family-based scheduling approach in most test cases outperforms the individual-based scheduling approach, in which jobs are not necessarily grouped into families.

Keywords: scheduling, meta-heuristic algorithms, family-based, in-line steppers

誌謝

本論文得以順利完成，首要感謝的是巫木誠教授的悉心指導與教誨，這二年接受老師的指導，不只是學習到鑽研學問的精神、做研究嚴謹的態度，更是獲得許多寶貴的人生知識和態度，如：Work Hard、Efficiently、Honest。同時也感謝許錫美教授和彭德保教授在論文口試時，對於論文給予寶貴的意見與指導，並撥冗參與口試過程，提供了許多指正與建議，讓本論文更臻完備。

在研究所的兩年中，要感謝同實驗室的金傳、邦豪、曉嚴、浩翰、秀文，陪我度過修課與撰寫論文的的日子，時常相互鼓勵與討論，與我分享日常生活的大小事，使我在交大的生活變的愉快而充實，另外，非常感謝子逸和昌穎的友情幫助，使我能多次突破研究的瓶頸，而在此特別感謝邱志文學長的幫助，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。

最後，特別感謝我最親愛的家人和憲耀，在論文撰寫期間給我無數的關愛與包容。特別是我的父母親，感謝您們多年來辛勞與關懷，您們的體諒讓我得以專心在論文的研究上。在此，謹以此論文獻給我最敬愛的家人、師長與朋友。

謝佩芸 于 新竹交大

2010'7'1

目 錄

中文摘要.....	i
Abstract.....	ii
誌 謝.....	iii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究議題.....	5
1.3 研究方法.....	6
1.4 研究流程.....	7
第二章 文獻探討.....	8
2.1 傳輸整合步進機之結構及製程介紹.....	8
2.2 傳輸整合步進機之排程研究.....	9
2.3 整備時間之分類.....	10
2.4 啟發式演算法 (Meta-heuristic Algorithm).....	11
2.4.1 基因演算法 (Genetic Algorithm).....	11
2.4.2 模擬退火法 (Simulated Annealing).....	13
2.4.3 禁忌搜尋法 (Tabu Search).....	14
2.4.4 螞蟻演算法 (Ant Colony Algorithm).....	15
2.4.5 粒子群最佳化 (Particle Swarm Optimization).....	18
2.4.6 GA-Tabu	21
2.5 文獻探討結語.....	24
第三章 求解多部傳輸整合步進機排序.....	25
3.1 研究問題描述.....	25
3.2 問題複雜度.....	26
3.3 以演算法求解多部傳輸整合步進機.....	27
3.3.1 染色體設計.....	27
3.3.2 染色體解讀.....	28

3.3.3 適合度函數.....	31
3.3.4 基因演算法求解過程.....	33
3.3.5 模擬退火法求解過程.....	39
3.5.6 禁忌搜尋法求解過程.....	40
3.3.7 螞蟻演算法求解過程.....	41
3.3.8 粒子群最佳化求解過程.....	42
3.3.9 GA-Tabu 求解過程.....	43
第四章 實例驗證.....	44
4.1 實驗目的.....	44
4.2 測試情境的設計.....	45
4.3 演算法的參數設定.....	47
4.4 實驗結果與分析.....	48
第五章 結論與未來研究方向.....	56
5.1 研究的結論.....	56
5.2 未來研究方向.....	57
參考文獻.....	58



表目錄

表 2.1 文獻整理.....	24
表 3.2 適合度函數值之範例.....	31
表 4.1 傳輸整合步進機反應室的加工時間（資料來源：Wu & Chiou (2009)） .45	
表 4.2 本主題各演算法相關參數之設定值（資料來源：Wu & Chiou (2009)） .47	
表 4.3 $(M,N) = (2,20)$ ，演算法與不同派工法則之差異	48
表 4.4 $(M,N) = (2,40)$ ，演算法與不同派工法則之差異	49
表 4.5 $(M,N) = (3,20)$ ，演算法與不同派工法則之差異	50
表 4.6 $(M,N) = (3,40)$ ，演算法與不同派工法則之差異	51
表 4.7 $(M,N) = (2,20)$ ，各個演算法間之解品質差異	52
表 4.8 $(M,N) = (2,40)$ ，各個演算法間之解品質差異	53
表 4.9 $(M,N) = (3,20)$ ，各個演算法間之解品質差異	54
表 4.10 $(M,N) = (3,40)$ ，各個演算法間之解品質差異	55



圖目錄

圖 1.1 傳輸整合步進機之組態.....	2
圖 1.2 單傳輸整合步進機在滿批量情境無產能損失之範例.....	3
圖 1.3 單部傳輸整合步進機在小批量情境產能閒置之範例.....	4
圖 1.4 單部傳輸整合步進機在工件排序策略下無產能損失範例.....	4
圖 2.1 傳輸整合步進機的生產系統.....	9
圖 2.2 Procedure Tabu (π_{in}, π_{out}) 之求解過程 (Wu & Chiou (2009)).....	23
圖 3.2 染色體設計範例.....	28
圖 3.3 Among Family Sequence 的解讀範例.....	29
圖 3.4 Within Family Sequence 的解讀範例.....	29
圖 3.5 Family Assignment 的解讀範例.....	30
圖 3.6 求解染色體的適應式值.....	32
圖 3.7 基因演算之求解流程.....	33
圖 3.8 染色體交配：C1 運算.....	34
圖 3.9 染色體交配：LOX 運算.....	35
圖 3.10 染色體交配：PMX 運算.....	36
圖 3.11 染色體突變：SWAP 運算.....	37
圖 3.12 染色體突變：Inverse 運算.....	37
圖 3.13 模擬退火法之求解流程.....	39
圖 3.14 禁忌搜尋法之求解流程.....	40
圖 3.15 螞蟻演算法之求解流程.....	41
圖 3.16 粒子群最佳化之求解流程.....	42
圖 3.17 GA-Tabu 之求解流程.....	43

第一章 緒論

1.1 研究背景與動機

半導體產業技術近年來不斷的成長，因此市場與技術的需求也逐漸增加，故必須以興建廠房的方式允以應付。然而當投資興建工廠時，機台的購買數量會是一個重要的決策，對半導體晶圓廠來說，當建造一座十二吋晶圓廠時，大約有 75%的投資金額投入於機台的購買，故可知必須充分利用機台之產能，才可減少購買過多的機台。然而在生產現場中，瓶頸機台為決定產出之關鍵，因此瓶頸機台的產能利用相當重要，唯有充分利用瓶頸機台之產能，方能增進工廠實際的產出。而在半導體晶圓廠設備投資金額中，又以傳輸整合步進機 (In-line Stepper)所佔比例最高，故在機台購買與生產配置的策略上，會將傳輸整合步進機成視為瓶頸機台，故提高傳輸整合步進機的利用率，以降低產能損失，可在不增購機台的情況下，提高晶圓廠的產出，減少不必要的投資。

在半導體晶圓廠中，生產的加工單位為晶圓片 (wafer)，基於加工經濟規模考量，通常會將相同產品群組成一個生產批 (lot)。每一個晶圓批又可稱為工件 (Job)，且一個晶圓批最多容納 25 片晶圓片。而傳輸整合步進機主要由暫存區 (Buffer)、埠區 (Port) 以及反應室 (Chamber)三個部分所構成，如圖 1.1 所示。暫存區 (Buffer)位於機台的外部，主要功能為暫時儲放等待分配到機台的工件；埠區 (Port)和反應室 (Chamber)位於機台的內部，埠區 (Port)用以暫時存放等待到反應室加工的工件，一部機台最多設計 4 個埠區，每一個埠區僅放置一個晶圓批 (Lot)，而反應室 (Chamber)主要為加工晶片之工作站，一部機台約由 20 多個反應室所構成，每一個反應室一次加工一片晶圓 (Wafer)，其生產方式會依照固定的製程順序一個步驟緊接著下一個步驟加工，故又可將之視為一流程式生產 (Flow Shop)。

此外，在傳輸整合步進機中，曝光反應室 (alignment)為機台之瓶頸製程，其為設置時間發生之所在。當生產之晶圓片於曝光反應室 (alignment)內，會依據曝光製程之

不同，所需的光罩 (mask)亦不同，於每次光罩的更換，均會產生一整備時間 (Setup time)，其大約需要 1.5 至 6 分鐘，因此不同的晶圓批所需的曝光製程不同時，必須更換光罩；反之，不同的晶圓批所需的曝光製程相同時，則不必更換光罩，然而若將相同製程之晶圓批視為同一家族，允以集中進行加工，則可以降低設適時間之發生。

傳輸整合步進機的傳輸單位可分為兩種，分別為晶圓批 (Lot)和晶圓 (Wafer)，如圖 1.1 所示，從暫存區至埠區的傳輸單位為晶圓批 (Lot)；從埠區至反應室的傳輸單位為晶圓 (Wafer)；從反應室回至埠區的傳輸單位為晶圓 (Wafer)；從埠區回至暫存區的傳輸單位為晶圓批 (Lot)，形成傳輸單位不一致的問題。由於機台具有傳輸單位不一致的問題，故使得機台必須加工完一個晶圓批方可離開埠區，好讓下一個晶圓批進入埠區準備加工，因此可知傳輸整合步進機為傳輸單位不一致的流程式生產模式。

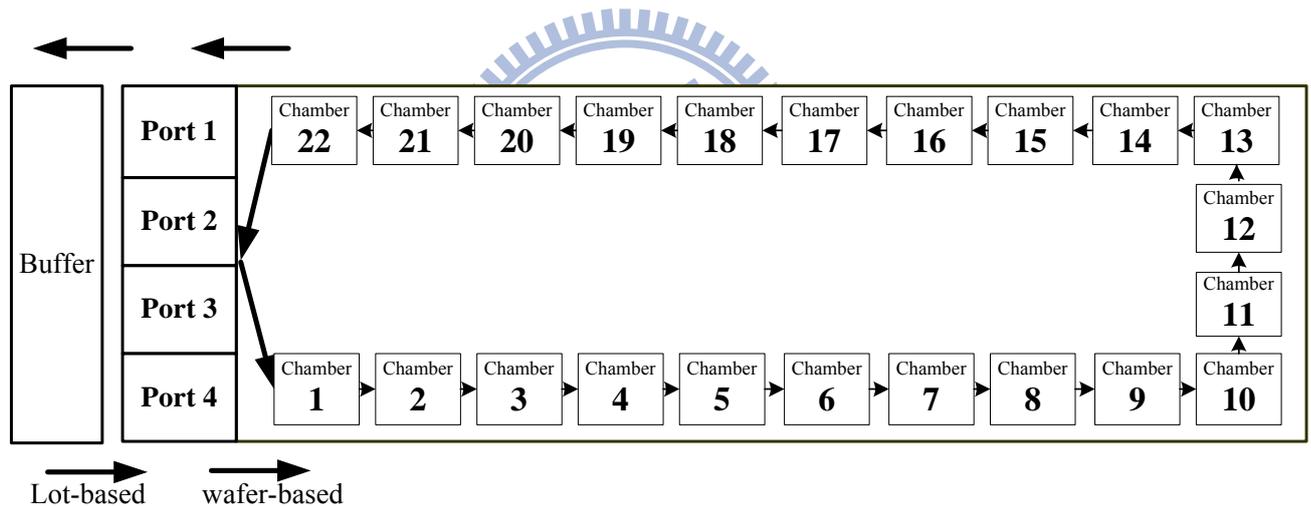


圖 1.1 傳輸整合步進機之組態

在過去傳輸整合步進機的排程文獻研究中，幾乎都是以滿批量 (Full-lot)的情境作為假設，當晶圓批內的晶圓數達到 25 片晶圓，就稱作滿批量。在此情境下，傳輸整合步進機不會有產能損失的問題產生，如圖 1.2 所示，傳輸整合步進機有 22 個反應室，Lot 1、Lot 2、Lot 3 及 Lot 4 共 4 個晶圓批在埠區依序準備加工，每一個晶圓批皆呈現滿批的狀態。當 Lot 1 之第一片晶圓加工完成後回至埠區時，由圖示可知，並且沒有任何一個反應室閒置，因此過去在滿批量情境假設下的文獻，皆將傳輸整合步進機視為單一部機台做為研究。

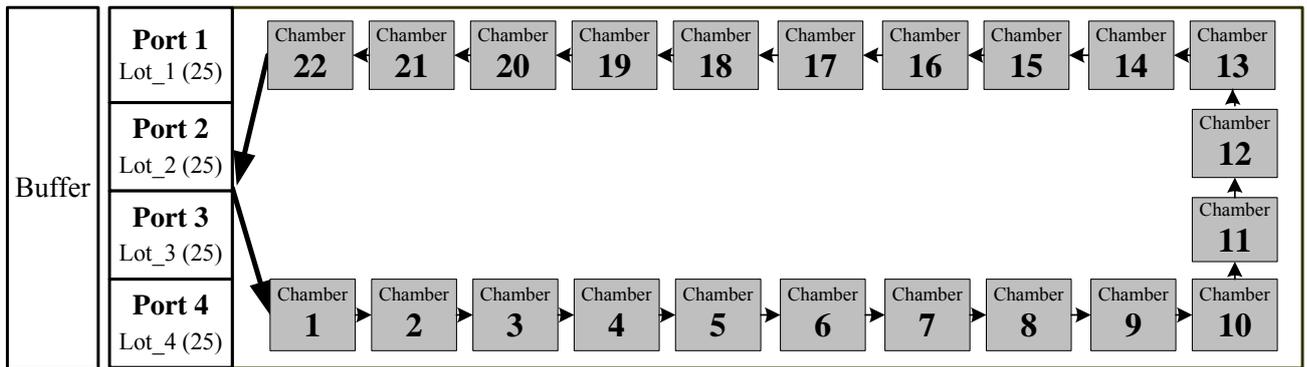


圖 1.2 單傳輸整合步進機在滿批量情境無產能損失之範例

隨著科技不斷進步，消費者需求的多變化，使得產品的生命週期逐漸縮短，故必須不斷開發新產品，方能保有競爭力以賺取更高的利潤，然而在新產品和新製程不斷的導入，晶圓批內的晶圓產生了不滿批（晶圓數量未達 25 片）的情況，稱作小批量 (Small-lot)。

小批量的發生原因可歸納為二：第一，在進行新產品或新製程的驗證時，會因為成本的考量，在投料時會將原本滿批量刻意改成小批量，如：一批晶圓批僅放置 6 片晶圓片。第二，產品在試產階段時，常因低良率的狀況，導致滿批的晶圓批在進入傳輸整合步進機前，就因製程或設備出問題造成部分晶圓報廢，因而形成小批量的情形。假使一間晶圓廠同時俱備新產品和新製程的導入，則小批量發生之情形將顯而易見。

在小批量的情境下，傳輸整合步進機會因為機台埠區的限制和傳出單位不一致造成產能之間置，如圖 1.3 所示，傳輸整合步進機有 22 個反應室，Lot 1、Lot 2、Lot 3、Lot 4 和 Lot 5 其中，Lot 1、Lot 2、Lot 3 和 Lot 4 在埠區依序準備加工，每個晶圓批中的晶圓數量分別為 3、7、4、5 片；由於機台埠區的限制 Lot 5 則無法進入機台中加工，僅能在暫存區等待被分配至機台上，此晶圓批中的晶圓數量為 7 片。當 Lot 1 之第一片晶圓加工完成後回至埠區時，由圖示可知，機台會有 5 個反應室閒置，但若將工件 Lot 5 和 Lot 1 的加工順序交換，加工順序從原本 Lot 1→2→3→4→5 調整成 Lot5→2→3→4→1 後，則就不會有任何一個反應室閒置，如圖 1.4 所示，故由上述可

知小批量生產方式以及機台埠區的限制，會造成產能閒置，而好的排序可以充分運用機台的產能，使產能閒置的狀況獲得改善，然而半導體廠普遍存在多部傳輸整合部進機的現實情況，因此如何改善多部傳輸整合步近機的產能閒置，實為半導體晶圓廠之重要議題，亦為本研究之研究動機。

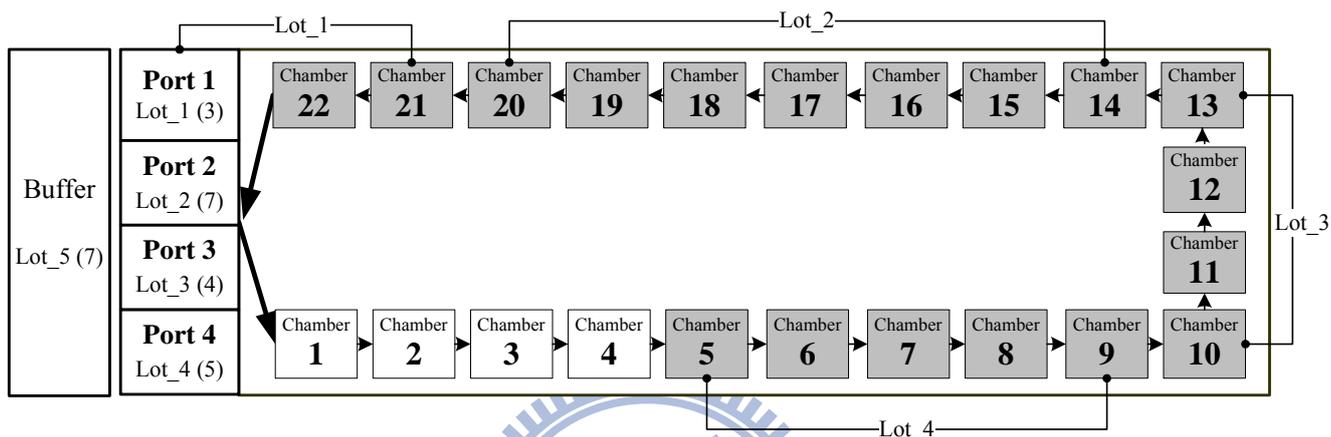


圖 1.3 單部傳輸整合步進機在小批量情境產能閒置之範例

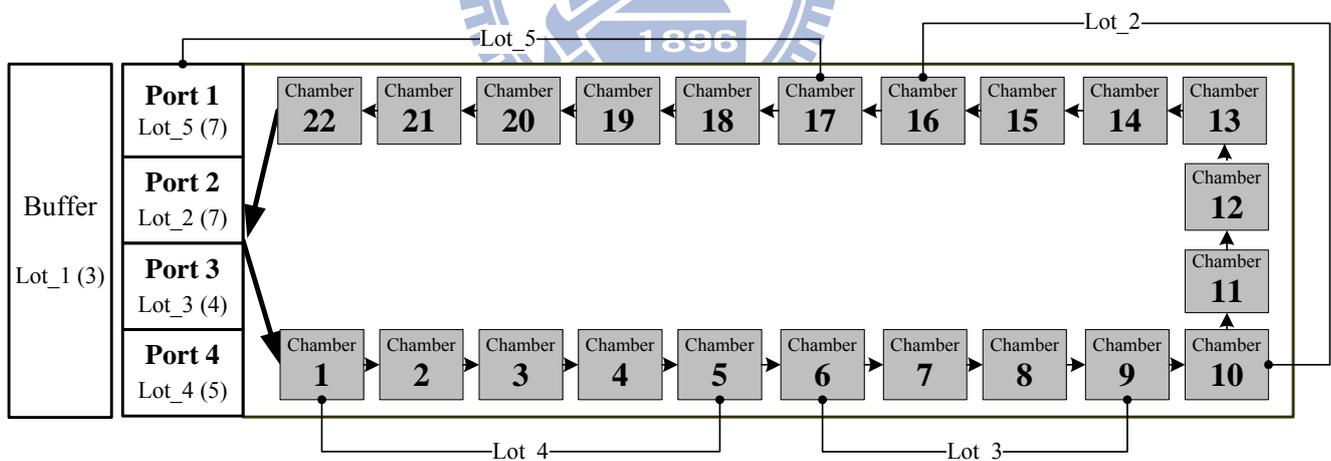


圖 1.4 單部傳輸整合步進機在工作件排序策略下無產能損失範例

1.2 研究議題

由於過去半導體產業處於量產階段，其製程良率高且穩定，故以往的文獻研究中，幾乎都是針對滿批量 (Full-lot)的情境作為假設，在此情境假設下，傳輸整合步進機不會因為傳輸單位不一致與埠區限制導致產能的損失。然而近年來新產品和新製程不斷的導入，由於新製程的不穩定以及新產品驗證的成本考量，使得投料至傳輸整合步進機的晶圓批往往呈現小批量的情形。

在小批量的情境中，傳輸整合步進機就會因為傳輸單位不一致與埠區限制造成產能的損失。除此之外，在半導體廠的實際狀況中，每一個準備加工的晶圓批會面臨到 2 至 3 部傳輸整合步進機的決策選擇，如圖 1.5，故如何透過工件適當的派工和排序，增進多部傳輸整合步進機的平均產出（減少傳輸整合步進機的平均產能閒置）為半導體廠所重視的議題。

因此本論文主要探討小批量在多部傳輸整合步進機的排程問題，藉由求取最佳的工件排序，改善機台產能的閒置發生，並且納入光罩設置的情境，以更符合傳輸整合步進機在工廠裡實際運作的情況，讓研究更有價值。



圖 1.8 實際的多部步進機排程

1.3 研究方法

過去 Wu & Chiou (2009) 已針對小批量的情境作傳輸整合步進機的排程，目的在於增加工廠之產能，但此論文並未考慮產品族(Job Family)的情況，因此假設每一個晶圓批使用的光罩皆不同，故在加工每一個晶圓批之前，都需要更換光罩，即加工每一個晶圓批均具有換光罩的整備時間。

然而在半導體廠不同晶圓批使用相同的光罩情境是顯而易見的，故 Wu, Lu, & Chiou (2009)將產品族 (Job Family) 的情境納入討論，並且探討小批量在單部傳輸整合步進機的排程問題，此論文使用家族式派工法則 (Family-based)，盡量將相同加工條件 (相同光罩) 的晶圓批排序一起加工，以減少在加工時重覆更換光罩的次數，使其總完工時間縮短，增進機台產出率，除此之外，其提出基因演算法之染色體新的解讀法則，此染色體主要為結合基因演算法和家族式派工法的概念下設計而成，而新的染色體解設計和讀所獲得的最佳近似解，已證實比過去的設計和解讀法好 (Wilson et al., 2004; Franca et al., 2005; Lin et al., 2009)。

但在半導體廠中並非僅有單部傳輸整合步進機，在實務上的情境為多個晶元批在暫存區 (buffer) 中，每一個準備加工的晶圓批會面臨到 2 至 3 部傳輸整合步進機的決策選擇。因此本研究主要探討小批量在多部傳輸整合步進機的排程問題，並且納入光罩設置的情境，使其更符合半導體廠的真實運作，此外更應用 Wu, Lu, & Chiou (2009) 提出的染色體之設計與解讀法，以求得更好的近似解。由於傳輸整合步進機為瓶頸站，故如何增進多部傳輸整合步進機的平均產出率 (減少傳輸整合步進機的平均產能閒置率) 為半導體廠所重視的，因此本研究目的在於透過工件適當的派工和排序，使傳輸整合步進機的平均產出率最大，然而由於此問題之解空間龐大，故將使用不同的啟發式演算法於多機上之績效探討。

1.4 研究流程

本論文計畫書文之內容架構分為五個章節。第一章緒論，說明研究背景與動機、研究目的及研究議題等。第二章文獻探討，針對傳輸整合步進機的機台結構與製程、步進機和流程式生產製程以及多機台排程等之相關文獻，加以整理、歸納，作為與本研究問題特色之參考及比較，最後針對求解工具啟發式演算法之相關研究進行探討。第三章研究方法，詳述傳輸整合步進機生產情境、相關參數假設，並說明啟發式演算法如何求解傳輸整合步進機排序問題。第四章為實例驗證。第五章則敘述結論與後續研究方向。



第二章 文獻探討

此章節將介紹與本研究相關之文獻。本研究主要內容為探討小批量在多部傳輸整合步進機之排程問題。因此，首先將針對傳輸整合步進機之機台結構及製程作介紹，接著探討傳輸整合步進機之排程研究之相關議題，由於本研究有光罩設置之限制，故進一步探討家族式派工法則和設置時間之相關文獻，更藉由文獻之回顧，確立本研究之問題獨特性。最後，本研究使用啟發式演算法進行問題求解。故於本章節亦會進行啟發式算法之相關文獻回顧。

2.1 傳輸整合步進機之結構及製程介紹

傳輸整合步進機的生產系統主要由在製品暫存區 (WIP buffers)、埠區 (Port)、以及步進機之反應室 (Chamber)所組成 (Quirk 2001; Xiao 2001)。在製品暫存區主要用來暫存待加工之生產晶圓批。埠區為暫存欲進入傳輸整合步進加工之生產晶圓批，一部機台最多設計 4 個埠區，每一個埠區僅可容納一個晶舟盒，即一個晶圓批 (Lot)。傳輸整合步進機約由 20 多個反應室所構成，每一個反應室一次只能容納一片晶圓片進行加工。由於暫存區、埠區和反應室的單位不同，故在傳輸的過程中會產生傳輸單位不一致的現象 (Transportation incompatibility)。

傳輸整合步進機內部的佈置圖，包括了自動化光阻塗佈及顯影系統與步進機，如圖 2.1 所示。自動化光阻塗佈及顯影系統包括 7 種站點 (Stage)，分別為氣相塗底 (Vapor prime)、光阻塗佈 (Photo-resist coating)、軟烤 (Soft-bake)、對準與曝光 (Alignment)、曝光後烘烤 (Post-exposure bake)、顯影 (Develop)以及硬烤 (Hard-bake)，其中，每一種站點有一個以上的反應室可以加工。步進機包括有 3 種站點，分別為預對準 (Pre-alignment)，曝光(Exposure)，以及暫存區 (Buffer)，而步進機之每一種站點僅有一個反應室可以加工，此外於曝光前會發生更換光罩之情況。

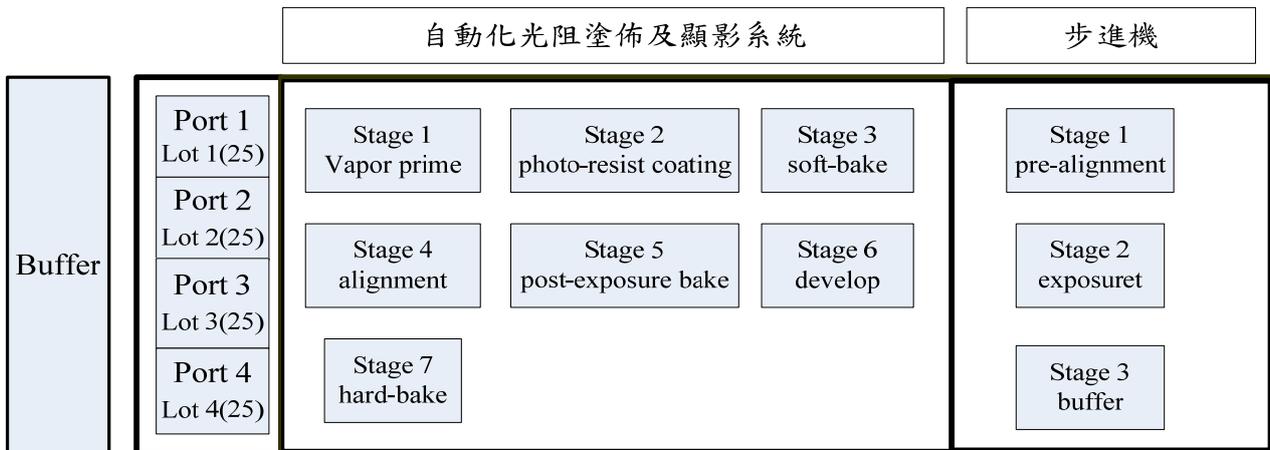


圖 2.1 傳輸整合步進機的生產系統

2.2 傳輸整合步進機之排程研究

傳輸整合步進機為半導體廠之瓶頸機台，因此傳輸整合步進機之排程問題為各個半導體廠所重視的，但過去在傳輸整合步進機的排程問題之研究中，所探討的主要是於滿批量之情境之排程問題，在此情境下，傳輸整合步進機不會因為傳輸不一致與埠區限制導致產能損失，因此以往在滿批量情境假設下之文獻，皆將埠區和反應室視為單一部機台作研究。

於傳輸整合步進機的排程問題中，可從過去的文獻中，將問題歸納成兩類，如下整理：

1. 單機排序：視傳輸整合步進機為單一步機台作工件之排序。
2. 生產系統排序：將傳輸整合步進機和其他非瓶頸機台所視為一生產系統作生產排序 (Ying & Lin, 2009; Wu et al., 2008; Wu et al., 2008; Wu et al., 2006; Chern & Liu, 2003)。

而近期的文獻中，Wu & Chiou (2009)所討論的為小批量在傳輸整合步進機的排程議題，其中，作者將傳輸整合步進機之排程問題，視為一傳輸單位不一致之流程式生產模式之排程問題。其主要著重在小批量的情境下，如何充分運用傳輸整合步進機的產能，且假設每一個晶圓批使用的光罩皆不同，故在加工每一個晶圓批之前，都需要

更換光罩。Wu, Lu, & Chiou (2009)亦是探討小批量在單部傳輸整合步進機的排程問題，但其納入了光罩設置的情境，並且運用家族式派工法則 (Family-based)，盡量將相同加工條件之晶圓批排序一起加工，以減少在加工時重覆更換光罩的整備時間，使其總完工時間縮短，增進機台產出率。

2.3 整備時間之分類

在過去考量排程問題的研究中，整備時間大多忽略不計或是視為加工時間的一部分，但隨著產品生產型態之改變，從原本的少樣多量成為現今之多樣少量模式，因此整配時間對於排程準確性之影響越來越大。

Potts & Kovalyov (2000)將研究設置時間的問題區分為許多不同的種類，其中家族設置時間之問題被區分成順序獨立家族設置時間問題 (Sequence-Independent Family Setup Time)和順序相依家族設置時間問題 (Sequence-Dependent Family Setup Time)。然而本研究之設置時間是發生在換光罩的時候，當加工不同的電路佈局之工件，進入傳輸整合步進機加工，更換工件時，會產生更換光罩的設置時間之浪費，反之加工相同的產品族會使用相同的光罩，即具有相同的設置。因此可知本研究為家族設置時間 (Family Setup Time)之問題。由於每次更換光罩所花費的設置時間皆相同，又可將問題可歸納為順序獨立之家族設置時間之問題。

2.4 啟發式演算法 (Meta-heuristic Algorithm)

啟發式演算法為針對問題特性所發展出來的解法，這一類的方法可以在短時間內求得近似最佳解，故當求解大規模問題時，求解之時間會因為解空間之龐大過長，因此一般廣泛的運用啟發式演算法，如基因演算法、模擬退火法、禁忌搜尋法、螞蟻演算法以及微塵粒法等，(Cavalieri *et al.* 1999; Wang and Uzsoy 2002)。

2.4.1 基因演算法 (Genetic Algorithm)

基因演算法最早由學者 Holland (1975)所提出，其主要是根據達爾文進化論中『物競天擇，適者生存』的原理，發展出的群體搜尋演算法，且 Goldberg (1989)指出，基因演算法在求解最佳化的問題上，比傳統最佳化方法具強健性，因此基於基因演算法之特性，使其廣泛應用於工程或是科學之最佳化求解。而基因演算法的求解運作需要以下四個要件，分別為：

染色體編碼

使用基因演算法進行問題求解前，首先需進行染色體之編碼，其主要目的在於能藉由染色體之解碼，將問題之決策變數轉換成字串，使染色體可代表所要探討問題之可行解。因此可知字串即為染色體，染色體字串中的每個字元稱為基因，本研究設計概念將於本文第三章加以說明。

適合度函數

在基因演算法求解過程中，可同時存在多條染色體，然而這些染色體所形成的群體稱之為族群，不同染色體代表不同解，於本研究中，染色體所代表的為不同之排程方式。染色體會依據研究問題之定義，計算其適合度函數值以評估各條染色體之好壞，即評估各種解之優劣。適合度函數值佳，其代表染色體的適應力較強，存活機率也較高，即解較好。反之，適合度函數值，其代表染色體適應力差，存活機率低，即解較差。故適合度函數值之計算，是為了能在基因演算法進行問題求解之過程，評定各染色體之適應力高低，即各種解之好壞。

染色體之交配與突變

交配主要為隨機選擇成對之染色體，進行部分基因交換，以增產更好的染色體。換句話說，交配為透過母代 (Parent) 染色體間之訊息交換，產生二條具有親代特性的子代 (Child) 染色體，以達到基因演算法搜尋之目的。突變 (Mutation) 的功能在於增加群體的變異度，以提供新的搜尋方向，使基因演算法避免陷入局部最佳解，因此突變通常會與交配一同使用。

交配的方法有單點交配 (One-point crossover)、線性順序交配 (LOX: Linear Order Crossover Operator)、部分相應交配 (PMX: Partially Matched Crossover Operator) 與 NABEL Operator 等方式。而突變方式有交換 (Swap)、反轉 (Inverse) 及插入 (Insert) 等方法 (Wang & Zheng, 2003)。

選擇策略

選擇運算元在基因演算法中為扮演天擇之角色，其依據每一個染色體之適應值高低來決定該染色體保留至下一代之機率，亦代表對於環境適應力高的染色體有較高的機率會繼續演化，而常用的選擇策略有 Goldberg (1989) 提出之俄羅斯輪盤法 (Roulette wheel selection 與競賽法 (Tournament selection))。

輪盤法的概念為分割輪盤，即依照染色體的適應函數值將輪盤分割為數等份，故適應函數值越好所佔輪盤面積之比例則越大，若隨機選取輪盤上之一點，面積越大之染色體區域越易被選取，亦即越可能存活至下一代。而競賽法的作法為從同一族群中隨機挑選兩條或兩條以上的染色體，比較其適合度函數值，選取適合度較佳者存活至下一代，此方法能避免過早陷入局部最佳解，卻可能較難收斂產生近似最佳解。

基因演算法的求解運作，整理如下：

步驟一. 編碼

設計問題解的表示方式進行編碼。

步驟二. 產生初始染色體群

以隨機排序基因格的方式產生不重複的染色體群，稱為初始母體 (Population)。

步驟三. 評估適合度函數

評估每一個解依據其優劣程度給予適應度函數，並判斷是否滿足終止條件時，若滿足，則其代表所獲取之解為最佳近似解；反之，進行步驟四。

步驟四. 染色體演化

設定交配率和突變率，並應用交配運算子(Crossover)產生二個新的子代染色體，或者應用突變運算子(Mutation)產生一個的新子代染色體。

步驟五. 選擇策略

從母代與子代的染色體池中 (pool)篩選較佳的染色體，以保留至下一代。

步驟六. 終止條件

當滿足終止條件時，停止演化法，其代表所獲取之解為最佳近似解。

若未滿足終止條件，則重複步驟四至步驟五，直至符合終止條件。

2.4.2 模擬退火法 (Simulated Annealing)

模擬退火法最早由 Kirkpatrick *et al.*(1983)用於求解組合最佳化的問題，也因而受到重視。模擬退火法基本的概念來自於當固體加熱至結構瓦解成為液體結構，再對其降溫過程加以控制，使得分子在變回固體結構時，重新排列成預期的穩定狀態，故模擬退火法主要為模擬自然界的物理現象，提供尋求最佳解之一種搜尋法則。

而模擬退火法的特色在於求解的過程中，於高溫時，隨機尋找的鄰近解之機率較高，易被接受，此時近似隨機搜尋；而低溫時，隨機搜尋的鄰近解之機率較低，不易

被接受，此時近似區域搜尋，因此其為利用機率關係提供跳脫局部最佳解之機會。

模擬退火法之求解運作過程如下：

步驟一. 設定參數：初始溫度 $T_0=500$ 及 $\alpha=0.9998$ 。

隨機任選一組解作為初始解 $S(0)$ 。

步驟二. 以 Metropolis 演算法判斷是否接受鄰近解當現行解：

- 根據現行染色體 $S(t)$ ，以交換(SWAP)方式產生一組新的鄰近解 $S'(t)$
- 設定 $\Delta E''=C(S'')-C(S)$ ， $C(S)$ 代表現行解的總完工時間 (makespan)， $C(S'')$ 代表鄰近解的總完工時間 (makespan)。若 $\Delta E'<0$ ，表示鄰近解較佳，則接受鄰近解取代現行解；若 $\Delta E'<0>0$ ，表示鄰近解未比現行解好，則計算一個機率 $\exp(-\Delta E'<0/\alpha T)$ ，作為判斷是否接受 S' 當做現行解的依據。此時產生一個介於 $[0,1]$ 的亂數 u ，若 $u < \exp(-\Delta E'/\alpha T_0)$ ，則接受鄰近解取代成為現行解，反之則否。
- $T(t+1)=\alpha T(t)$ 。

步驟三. $t=t+1$ ，當 $T(t+1)<1$ 則終止；反之，則重複步驟二。

2.4.3 禁忌搜尋法 (Tabu Search)

禁忌搜尋法是由 Glover (1989) 提出，適於處理各種最佳化及排列組合之問題，其主要為利用可行解的名單來搜尋下一個解的方向，並以記憶方式來避免不必要之搜尋。禁忌搜尋法主要分為兩個階段來考量，分別為短期記憶及長期記憶。短期記憶階段為記錄搜尋過的點，以減少重複與反覆的搜尋，其目的在於加速達到區域最佳化；而長期記憶階段為透過短期記憶，尋找適合的方向找尋最佳解，其與傳統區域搜尋法之差別在於此方法為利用本身機制來跳脫區域最佳解 (local optimum)，以達到整體最佳解(global optimum)。

而在禁忌搜尋法中，移步即為產生鄰近解之方法，常用的方法有交換 (SWAP)、插入 (Insert)、加入 (Add)和消去 (drop)等，鄰域代表所有鄰近之可行解，此為透過移

步法則將現形解轉移至另一解之所有可能解之集合。禁忌名單又稱為短期記憶之記憶體，用以記錄過去一段時間搜尋中所選移步之屬性，然而禁忌名單大小有各種研究，當名單愈大，所需的記憶體空間越大，搜尋過程亦較為耗時，但可使陷入區域最佳解之機率降低，一般採用 Glover (1989)所提出的魔術數字 7 作為禁忌名單之大小，當禁忌名單內成員數達到上限時，會依據破禁原則更新名單之成員，破禁原則即為解除禁忌名單之條件，禁忌搜尋法求解運作過程如下所示：

步驟一. 建構可行之起始解

步驟二. 採用兩兩交換最陡下降法 (steepest descent pairwise interchange)，即使用現行解，以交換 (SWAP)產生所有可能的鄰近解，然後從這些解中的找出最佳解。

步驟三. 找尋禁忌名單 (tabu list)中是否有與此鄰近最佳解相同的值，若無則採用此鄰近最佳解，並將此組解放入禁忌名單並更新禁忌任期（保留次數）；反之，則僅更新禁忌任期（保留次數）且重複步驟二。

步驟四. 檢查此鄰近解最佳解比現行最佳解好，則取代成為最佳解。

步驟五. 當滿足終止條件時，停止演化法，其代表所獲取之解為最佳近似解；若未滿足終止條件，則重複步驟二至步驟四，直至符合終止條件。

2.4.4 螞蟻演算法 (Ant Colony Algorithm)

螞蟻演算法由 Marco Dorigo (1992)提出，其為仿照自然界螞蟻行為所發展出來的演算法，主要是透過許多人工螞蟻(artificial ant)，模擬真實螞蟻找尋食物與螞蟻巢穴之間最短路徑的行為。螞蟻們一開始會隨機地選擇巢穴與食物間的路徑，並且於移動同時散發費洛蒙 (pheromone)，不同的螞蟻會有不同的路徑選擇，隨著螞蟻的經過次數增加時，費洛蒙濃度亦會隨之增加；反之，濃度則降低，並隨著時間而漸漸蒸發掉，而螞蟻在移動過程中，會以路徑上的費洛蒙濃度作為下一條行走路徑之選擇依據，當時間越長，導引隨後的螞蟻便能循著其他螞蟻的旅行經驗找到食物，即愈來愈多的螞蟻會選擇行走同一路徑（最短路徑）往返蟻穴到食物目的地，因此可以利用螞蟻演算

法解決最佳化問題。而從螞蟻系統 (AS) 發展至今，根據不同問題的特性不斷改進，衍生出求解各種問題之演算法，如 ACS、ASe、M-MMAX、PACO 等，其中 M-MMAX 由 Rajendran (2004) 提出，求解運作過程如下所示：

步驟一. 起始解

以 NEH 產生一組起始解，並將產生的起始解之適應值設為 Z_{best} ， Z_{best} 代表目前找出的最佳解。而費洛蒙路徑的上、下限則設定為 τ_{max} 及 τ_{min} ，其所代表之數學式分別為： $\tau_{max}=1/((1-\rho)/Z_{best})$ ， $\tau_{min}=\tau_{max}/5$ 。其中 $(1-\rho)$ 代表蒸發率(evaporation rate)；起始值對所有的 i 到 j 之 $\tau_{ij}=\tau_{max}$ 。

步驟二. 螞蟻順序的產生：

$T_{ik}=\sum_{q=1}^k \tau_{iq}$ ， T_{ik} 代表工件 i 到位置 k 的累計機率， n 設為總工件數， τ_{ij} 為

節點強度 (trail intensity)。

由隨機產生一個介於 $[0,1]$ 之間的數字 u ，當 $u \leq (n-4)/n$ ，則在最佳排序中未被選擇的前五個工件中，選擇最大 T_{ik} 的工件，或者以 p_{ik} 隨機選取，

$p_{ik}=(T_{ik}/\sum_l T_{lk})$ ， l 代表前五個未排序的工件。

步驟三. 步驟 3：局部搜尋

For $i=1$ to n

For $k=1$ to n

若 $[k] \neq i$ 代表在現有的排序中，將工件 i 插到 k 的位置；而其餘位置之排序不變，並計算其排序後的適應值，從中找出最佳解。

步驟四. 更新費洛蒙強度：

當局部搜尋已更新三次後，則將現有的適應值設為 $Z_{current}$ ，如工件 i 放在已產生的排序的 k 位置上，則 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old} + (1/Z_{current})$ ；或者 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old}$ 。

若 $\tau_{ik}^{new} > \tau_{max}$ ，則 $\tau_{ik}^{new} = \tau_{max}$ ；若 $\tau_{ik}^{new} < \tau_{min}$ ，則 $\tau_{ik}^{new} = \tau_{min}$ 。不斷重複步驟二至四直到符合終止條件。

PACO 由 Rajendran (2004) 提出，求解運作過程如下所示：

步驟一. 起始解

以 NEH 產生一組起始解，並將產生的起始解之適應值設為 Z_{best} ， Z_{best} 代表目前找出的最佳解。若 $(|工件 i 在 PACO 排序中的位置-k|+1) \leq n/4$ ；則 $\tau_{ik} = (1/Z_{best})$ ；若 $n/4 < (|工件 i 在 PACO 排序中的位置-k|+1) \leq n/2$ ；則 $\tau_{ik} = (1/(2 \times Z_{best}))$ ；或者 $\tau_{ik} = (1/(4 \times Z_{best}))$

步驟二. 螞蟻順序的產生

$T_{ik} = \sum_{q=1}^k \tau_{iq}$ ， T_{ik} 代表工件 i 到位置 k 的累計機率， n 設為總工件數， τ_{ij} 為節點強度 (trail intensity)。由隨機產生一個介於 $[0,1]$ 之間的數字 u ，則選擇最佳排序中未被選擇的第一個工件；否則若 $u \leq 0.8$ ，則在最佳排序中未被選擇的前五個工件中，選擇最大 T_{ik} 的工件；或者則隨機以 p_{ik} 選取， $p_{ik} = (T_{ik} / \sum_l T_{ik})$ ， l 代表前五個未排序的工件。

步驟三. 局部搜尋

For $i=1$ to n

For $k=1$ to n

若 $[k] \neq i$ 代表在現有的排序中，將工件 i 插到 k 的位置；而其餘位置之排序不變，並計算其排序後的適應值，從中找出最佳解。

步驟四. 更新費洛蒙強度

當局部搜尋已更新三次後，則將現有的適應值設為 $Z_{current}$ ，當工件數 $n \leq 40$ ，設 h 為工件 i 在現有排序中的位置，以及設 $|h-k| \leq 1$ 則 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old} + (1 / (diff \times Z_{current}))$ ；或者 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old}$ 。當工件數 $n > 40$ ，設 $|h-k| \leq 2$ 則 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old} + (1 / (diff \times Z_{current}))$ ；或者 $\tau_{ik}^{new} = \rho \times \tau_{ik}^{old}$ 。其中 $diff = (|工件 i 在 PACO 的最佳排序中的位置-k|+1)^{1/2}$ 。不斷重複步驟二至四直到符合終止條件。

2.4.5 粒子群最佳化 (Particle Swarm Optimization)

粒子群最佳化由 Eberhart and Kennedy (1995) 所提出，為一種仿生物演算法，透過迭代演化求得近似最佳解。其主要為模擬自然界鳥群飛行的行為所發展出來的演算法，如：鳥群覓食時，除了根據自己本身的認知，飛向曾走過之較佳覓食地點，亦會藉由群體間的合作溝通，得知其他鳥所發現之更佳覓食地點，當時間越長，所有的鳥群都會飛往最佳之覓食地點。

在粒子群最佳化中，搜索空間內的每一個粒子（鳥）都代表可能的最佳解，而各粒子速度會依據三個要素作更新，分別為各粒子本身速度、與個體最佳解的距離以及與群體最佳解的距離，再依據更新後的速度調整目前之位置，以更新每個粒子的搜尋距離與方向。

粒子群最佳化的運作過程如下所示(Wu & Chiou (2009))：

步驟一. 定義粒子群

將粒子群定義為”工件對應分配的位置” (“job-to-position”)，粒子 i 在世代 t 令為 $X_i^t = (x_{i11}^t, x_{i12}^t, \dots, x_{imn}^t)$, $x_{ijk} \in \{0,1\}$ 。因此當粒子 i 的工件 j 放在 k 個順序位置則 x_{ijk} 為 1，否則等於 0。

步驟二. 粒子飛行速度

粒子 i 在世代 t 的飛行速度定義為 $V_i^t = (v_{i11}^t, v_{i12}^t, \dots, v_{imn}^t)$, $v_{ijk}^t \in R$ 。

$$v_{ijk}^t = \omega v_{ijk}^{t-1} + c1r1(p_{ijk}^t - x_{ijk}^t) + c2r2(p_{gjk}^t - x_{ijk}^t)$$

每一個粒子都有一個最佳的適應值以 p_{ijk}^t 代表， $P_i^t = (p_{i11}^t, p_{i12}^t, \dots, p_{imn}^t)$ ，

$p_{ijk}^t \in \{0,1\}$ 。 $P_g^t = (p_{g11}^t, p_{g12}^t, \dots, p_{gmn}^t)$ ， $p_{gjk}^t \in \{0,1\}$ 代表整體的最佳解。

$c1, c2$ 是學習因子， $r1$ 及 $r2$ 為 0 到 1 的隨機亂數， ω 為慣性權重值。

步驟三. 改變的機率

$$\text{改變的機率 } s(v_{ijk}^t) = \frac{1}{1 - \exp(-v_{ijk}^t)}$$

步驟四. 建構一個粒子順序

$$\text{順序機率 } q_i^t(j,k) = \frac{s(v_{ijk}^t)}{\sum_{j \in F} s(v_{ijk}^t)}$$

F 代表未排的基因集合

步驟五. 解題邏輯的虛擬碼

初始參數: $N_p, c1, c2, w, V_{\max}, f, Vi = 0$

產生啟始粒子群 $X_i^1, i = 1, \dots, N_p$

Set $P_i^1 = X_i^1, B = X_1^1$ // B is best sequence in algorithm

Loop

// find P_i^t

For $i = 1$ to N_p

If $Z(X_i^t) \leq Z(P_i^t)$ then // $Z()$ evaluates objective function

$P_i^t = X_i^t$ // P_i^t is best so far

End if

Next i

// find P_g^t

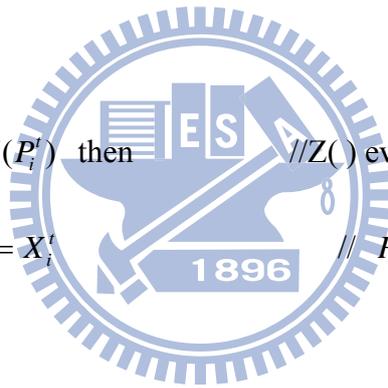
$g = 1$

For $i = 1$ to N_p

If $Z(X_i^t) \leq Z(P_g^t)$ then $g = I$ // g is index of best performer in

population

Next i



If $Z(P'_g) \leq Z(B)$ then $B = P'_g$

// update velocity trail

For $i = 1$ to N_p

$$v'_{ijk} = \omega v_{ijk}^{t-1} + c1r1(p'_{ijk} - x'_{ijk}) + c2r2(p'_{gjk} - x'_{ijk})$$

$$v'_{ijk} \in [-V_{\max}, +V_{\max}]$$

Next i

// construct a new feasible solution

For $i = 1$ to N_p

O_i is a null sequence for particle i

For $k = 1$ to n position

// changes of probabilities

For $j = \text{indices of } F$ // If $\#F < f$ then all unscheduled jobs
are considered

$$s(v'_{ijk}) = \frac{1}{1 - \exp(-v'_{ijk})}$$

Next j

// construction

For $j = \text{indices of } F$

$$q'_i(j, k) = \frac{s(v'_{ijk})}{\sum_{j \in F} s(v'_{ijk})}$$

If random number $\leq q'_i(j, k)$

then job j is placed in the k th position in O_i ; go to Next_Position;

Next j

Next_Position

Next k

$$X'_i = O_i$$

Next i

Until maximum number of iterations.

2.4.6 GA-Tabu

GA-Tabu 由 Mun et. al (2005) 應用於 PC 叢聚系統下，以求解分配系統負荷的最佳化的問題，其主要以GA為主軸求解，如：當 GA 的最佳解保持10代，則在 GA 母體中，取 10% 較佳解進行 Tabu 的搜尋方法，若所獲得之解有改善則加入 GA 母體中。Sundarambal et. al (2008) 以 GA-Tabu 求解 ATM (Asynchronous transfer mode)的網路問題，其亦以 GA為主軸求解，如：GA每次的突變求解後，接著進行 Tabu 的搜尋方法，再加入 GA 母體中。Wu & Chiou (2009) 則使 GA-Tabu 求解傳輸整合步進機之排程問題，其主要同時結合 GA 與 Tabu 之搜尋技術。由於GA可以在整體隨機搜尋可行解，而 Tabu 不但可以在區域 (local)鄰近解上找到更好的解，也可以在整體 (global)隨機找尋可行解，故其求解概念為：當 GA 找到更好的解時，藉由 Tabu 的區域鄰近法來調整，以找尋更好的解；反之，GA 一直無法找到更好的解時，藉由 Tabu的整體 (global)方法來求解，以跳脫目前的解。

工件加工順序的染色體以 $\pi = [\pi(1), \dots, \pi(n)]$ 表示，此處的 $\pi(j)$ 稱為基因，代表工件在排序 π 的 j 的位置， $C_{\max}(\pi)$ 代表其適應值，Wu & Chiou (2009) 的 GA-Tabu 其解題步驟如下整理：

- 步驟一. 產生一個初始解之染色體，稱為 $P(0)$ ，並設定初始狀態有 π^o 及 π^* ， π^o 為目前最好的染色體，其中 π^* 代表 tabu_seed 的染色體。
- 步驟二. 運用交配、突變以及選擇策略更新 $P(t)$ 。再藉由選擇策略從交配池中的新舊染色體以篩選出下一代存活的染色體 $P(t+1)$ 。
- 步驟三. 依據 π_{best} 去更新 π^o ， π_{best} 為 $P(t+1)$ 的最佳染色體，當 π_{best} 優於 π^o ，代表 π_{best} 是好的染色體，因此其鄰近解可以被進一步開發，故由 π_{best} 的鄰近解呼叫程序 Tabu 去創造新的染色體 π_{out} 。在這兩個染色體 π_{best} 及 π_{out} ，選擇最佳的解去更新 π^o 。若 π^o 在世代 t 未被更新則將其 tabu 任期 k 加 1。tabu 任期代表當世代在演進時， π^o 超過 π_{best} 的次數。

步驟四. 在世代 $P(t+1)$ 中，好的解(如：第二、三、四的次佳解)可以被用以更新 π^o 。

即使用第二、三、四的次佳解，以 Tabu 的程序，找出其鄰近解，用以產生更好的染色體去取代 π^o 。在本質上，步驟 3 及 4 為試圖當 GA 無法找到更好解的時候，以 Tabu 求解。

步驟五. 以 $tabu_seed \pi^*$ 去呼叫 Tabu 程序，分開求出其鄰近解，以產生較好的染色體去取代 π^o 。其演進的世代與上述的 GA 個別分開求解不同，如此可以使 π^* 完全跳脫 $P(t+1)$ 世代中的其他染色體。此特性可以幫助演算法跳脫 GA 陷入局部最佳法的困境。

Tabu(π_{in}, π_{out}) 程序是以 π_{in} 的鄰近解來創造新的染色體 π_{out} ，其用以更新 π^o 。輸入值 π_{in} 有兩個可能的來源: $P(t+1)$ 或 $tabu_seed \pi^*$ ， π^* 主要功能為協助染色體之演進，即當 $\pi_{in} = \pi^*$ ，使用 π_{out} 去更新 π^* 。綜合上述，可知 Tabu 的程序為更新三個參數，分別為 π^o 、 π^* 和 $P(t+1)$ 世代，如圖 2.2。



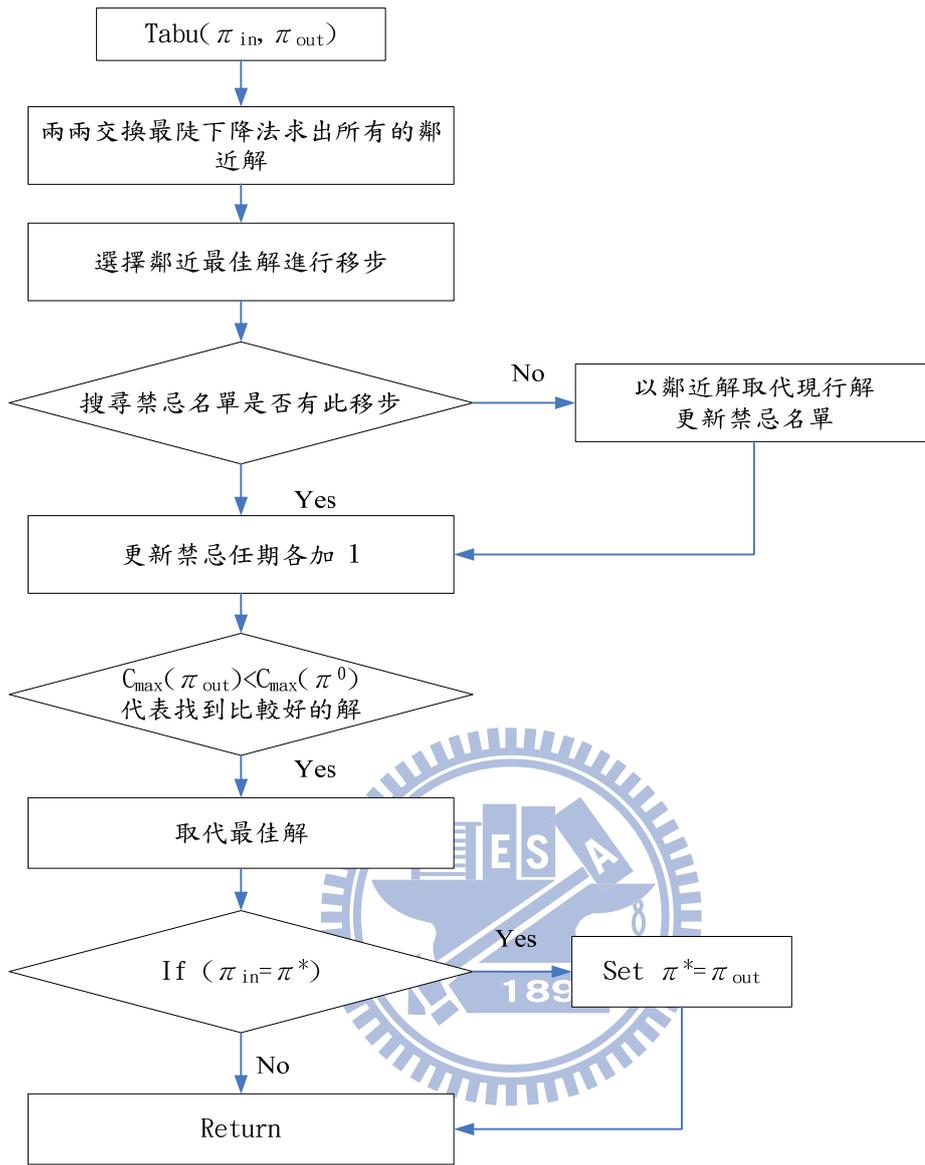


圖 2.2 Procedure $\text{Tabu}(\pi_{in}, \pi_{out})$ 之求解過程 (Wu & Chiou (2009))

2.5 文獻探討結語

在過去文獻研究中，均探討滿批量之情境假設，於此情境下傳輸整合步進機並無產能之閒置，因此皆將傳輸整合步進機視為生產系統中的一部瓶頸機台，但隨著新製程和新產品不斷的導入，小批量的發生明顯趨增，而傳輸整合步進機在小批量的情境下會發生產能之閒置，故在小批量情境下對傳輸整合步進機作規劃顯得相當重要。然而有關小批量情境下求解傳輸整合步進機之排程問題，目前只有 Wu & Chiou (2009) 以及 Wu, Lu, & Chiou (2009) 作探討。

小批量情境下求解傳輸整合步進機之排程問題可分為單機、多機、考量 Job-family (不同產品間可能使用相同光罩) 以及無考量 Job-family 四種情境。其中 Wu & Chiou (2009) 所探討的情境為單機未考慮 Job-family 以及多機和未考慮 Job-family，其主要透過七種演算法搭配單獨式派工法則進行求解，目標為使總完工時間最短，而 Wu, Lu, & Chiou (2009) 為研究單機且考慮 Job-family 之情境，運用基因演算法搭配家族式派工法則進行求解，目標亦為使總完工時間最短。本研究則探討多機且考慮 Job-family 之情境，求解方法為七種演算法搭配家族式派工法則，目標為平均產出最大，因此可知本研究在目標函數和研究方法上之獨特性，上述整理如表 2.1。

表 2.1 文獻整理

分類 情境	作者(年份)	Setup Time	目標函數	決策變數	方法
單 機	Wu & Chiou (2009)	No Job-family	Makespan	1.Job sequence	七種演算法+Individual ● GA-I、SA-I 等
	Wu, Lu, & Chiou (2009)	Job-family	Makespan	1.Among family sequence 2.Within family sequence	GA+Family ● GA-F
多 機	Wu & Chiou (2009)	No Job-family	Makespan	1.Job sequence 2.Job assignment	七種演算法+Individual ● mGA-I、mSA-I 等
	本研究 (2010)	Job-family	Throughput	1.Among family sequence 2.Within family sequence 3.family assignment	七種演算法+Family ● mGA-F、mSA-F 等

第三章 求解多部傳輸整合步進機排序

於實際的半導體晶圓廠中，現場生管人員必須在多部機台上作出良好的工件派工，通常現場派工的決策機台設定為 2~3 台，因此研究多部機台的派工實為半導體廠之重要議題，而本章要探討處理多部傳輸整合步進機台的派工問題，首先問題分析，其次介紹染色體的求解設計，接著再比較各個演算法之差異。

3.1 研究問題描述

在小批量的情境中，傳輸整合步進機就會因為傳輸單位不一致與埠區限制造成產能的損失，然而過去已有人針對單台傳輸整合部進機進行研究，現實半導體廠的情況中，傳輸整合部進台並非僅有一台，因此過去針對單步進機之研究並不適用於真實情況。

在半導體廠的實際狀況中，工廠約有 40 部傳輸整合步進機，每一個準備加工的晶圓批皆需指派至機台上加工，然而每一個準備加工的晶圓批會面臨到 2 至 3 部傳輸整合步進機的決策選擇，因此對半導體來說，研究多部傳輸整合步進機的派工實為重要。

而本章問題主要假設有 n 個晶圓批需要 m 個光罩在暫並於暫存區 (Buffer)，以等待被指派工至多部傳輸整和步進機上，然而本研究將使用相同光罩之晶圓批排序在一起，批視為同家族進行加工，因此在此情境下，工件會面臨三項重要決策，分別為 Family 需要分配給哪一個機台 (Family Assignment)、Family 在各個機台內的排序 (Among Family Sequence) 以及各 Family 內各工件的順序 (Within Family Sequencing)，故本研究提出一條染色體，能同時代表此三項決策，使解題空間變小，以求得更好的工件排序組合。

因此可知本研究擬針對半導體廠普遍存在多部傳輸整合部進機的現實情況，且將傳輸整合部進機視為流程式生產之模式，在小批量與傳輸單位不一致的假設情境下，整合多部傳輸整合部進機之績效產能，基於啟發式演算法提出新的排序組合策略，比較且探討多部傳輸整合部進機的排程規劃問題，目的在於最大化機台平均產出。

3.2 問題複雜度

本研究主要探討多部傳輸整合部進機的排程規劃問題，在有多部具有流程式生產問題特性的傳輸步進機下，找出最佳的工件排序模式。在此模型中，目標函數為最大化多部機台的平均產出，其限制式為機台內埠區的數目、反應室的加工順序以及各機台的產能分配，利用模擬傳輸整合步進機在半導體廠的真實運作情況，求出最佳化的排程組合，以最大化機台產能利用率。

在 n 個晶圓批需要 m 個光罩，將使用相同光罩之晶圓批排序在一起，視為同家族，等待配至多部機台上加工的情境中，將面臨三項決策變數，分別為 Family 需要分配給哪一個機台 (Family Assignment)、Family 在各個機台內的排序 (Among Family Sequence) 以及各 Family 內各工件的順序 (Within Family Sequencing)，要透過整數規劃求解上述三項決策的複雜度相當高，如： n 個晶圓批就有 $n!$ 個排序組合，假設現有 20 個晶圓批，所有解之排序即有 $20!$ 種組合。

除此之外，流程式生產之排程問題中，當工作站或機台數大於或等於三站（部）時，其屬於 NP hard 問題 (Gonzalez & Sahni, 1978; Cheng et al., 2000)。而本研究為排程規劃多部具有流程式生產問題特性的傳輸整合步進機，所探討的機台數為二至三部，由此可知，本研究問題亦屬 NP hard 問題。

NP hard 問題在小規模情境下，可利用線性代數的方法求得最佳解；當問題規模增大時，便無法於短時間內求得最佳解，由於短期生產排序工具必需能快速求解，並且有一個近似解滿足目標即可，因此本研究將利用七種啟發式演算法進行排程問題之求解比較。

3.3 以演算法求解多部傳輸整合步進機

此節將說明如何應用七種啟發式演算法，考量在小批量與光罩設置的情境下，求解多部傳輸整合步進機之排程問題。啟發式演算法之求解重點，主要在於如何透過染色體的編碼設計以及染色體的解讀，表達出問題之決策變數，並計算染色體的適合度函數，以作為排程績效評估之依據。因此本節會介紹染色體的設計與解碼、適應性函數的定義和七種啟發式演算法的求解過程，並配合適當的例子進行說明，本研究演算法的規劃步驟與演算流程如圖 3.1 所示。

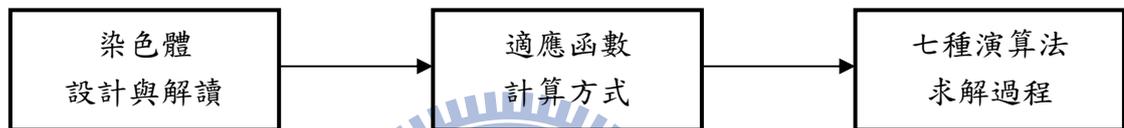


圖 3.1 研究方法流程

3.3.1 染色體設計

本研究使用一條染色體 (Chromosome) 來代表工件加工的順序，染色體中的基因 (Gene) 格數等同於工件個數。每格基因格存放各工件之編碼，且不重覆，此外每一個工件皆具有一個所屬光罩類別的屬性，亦可視為家族類別，當使用相同光罩之工件就稱為同一家族 (Job Family)，而此屬性並不為染色體的一部分，為工件所屬之屬性。

本研究染色體之設計概念，例來說明：若有七個工件欲進行排程，其編碼依序為 J_1 、 J_2 、 J_3 、 J_4 、 J_5 、 J_6 、 J_7 ，而每格基因格中，存放以上七個工件之編碼，且假設加工這七個工件，所屬的家族類別有四種，其編碼依序為 F_1 、 F_2 、 F_3 、 F_4 。此情境之染色體設計範例如圖 3.2 所示，可知工件的加工順序為 $J_1 \rightarrow J_2 \rightarrow J_4 \rightarrow J_6 \rightarrow J_3 \rightarrow J_5 \rightarrow J_7$ ，工件家族的情形分別為 J_1 所屬家族為 F_2 ， J_2 所使用的 F_3 光罩，而 J_4 所使用 F_1 光罩，以此類。

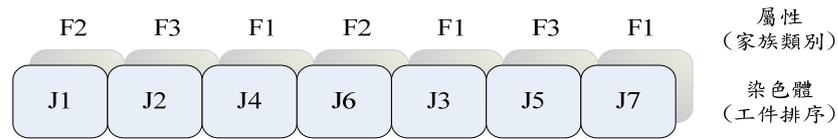


圖 3.2 染色體設計範例

3.3.2 染色體解讀

本研究為求解考量 Job family 之情境下，多部傳輸整合步進機台之最佳工件指派問題，其會面臨三項重要決策，分別為 Family 需要分配給哪一個機台 (Family Assignment)、Family 在各個機台內的排序 (Among Family Sequence) 以及各 Family 內各工件的順序 (Within Family Sequencing)，由於染色體設計的基因格數越少以及越簡單，會使基因演算法求解效率越高，因此本研究提出一條染色體，透過特定解讀法，將染色體之基因格之內容與排序等資訊，解讀成此三項決策，而在解讀染色體時，原染色體之基因格內容、排序皆不會允以更動。

在過去文獻中，Wu, Lu, & Chiou (2009) 在考量 Job family 之情境下，求解傳輸整合步進機排程問題時，提出了新的染色體解讀方法，其將 Family-based 的概念納入染色體的設計解讀中，使用相同光罩之工件排序在一起（同一家族），目的在於減少更換光罩的次數，降低設置時間之發生，而此解讀法之績效已證實比傳統之解讀法更佳。因此本研究亦應用 Family-based 解讀的概念於決定 Family 在各個機台內的排序 (Among Family Sequence) 以及 Family 內各工件的順序 (Within Family Sequencing)，再依負荷準則決定 Family 分配至哪一個機台上加工 (Family Assignment)。在染色體解讀中，本研究分為三部份進行說明：

Among Family Sequence

Family 在各個機台內的排序 (Among Family Sequence) 是決定於各個工件之屬性（家族類別）順序，例來說明：若有七個工件欲進行排程，所屬的家族類別有四種，隨機產生一組染色體（工件加工順序），如圖 3.3 所示，可知工件所屬之家族類別的

順序為 $F_2 \rightarrow F_3 \rightarrow F_1$ ，代表家族類別為 F_2 的工件會優先加工，接著加工家族為 F_3 的工件，最後是加工之家族為 F_1 的工件。

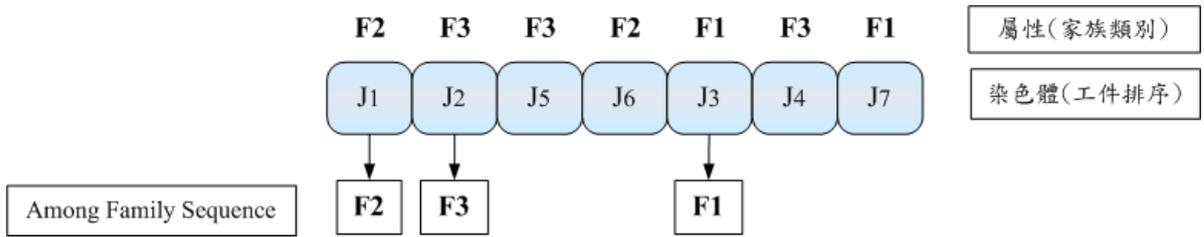


圖 3.3 Among Family Sequence 的解讀範例

Within Family Sequencing

確定家族（光罩）的加工順序後，接著排序出在每個家族內各工件的加工順序 (Within Family Sequencing)。而決定 Family 內 Job 的排序為依據基因之順序，如圖 3.4 所示，在同屬於 F_2 的基因順序為 1、6，代表在同屬於家族 2 之工件加工順序為 Job 1、Job 6；同為 F_3 的基因順序為 2、5、4，代表在家族 3 之工件加工順序為 Job 2、Job 5、Job 4；下同為 F_1 的基因順序為 3、7，代表在家族 1 之工件加工順序為 Job 3、Job 7。排序完各家族內的工件順序後，即獲得一組新的染色體，其加工順序為 $J_1 \rightarrow J_6 \rightarrow J_2 \rightarrow J_5 \rightarrow J_4 \rightarrow J_3 \rightarrow J_7$ 。

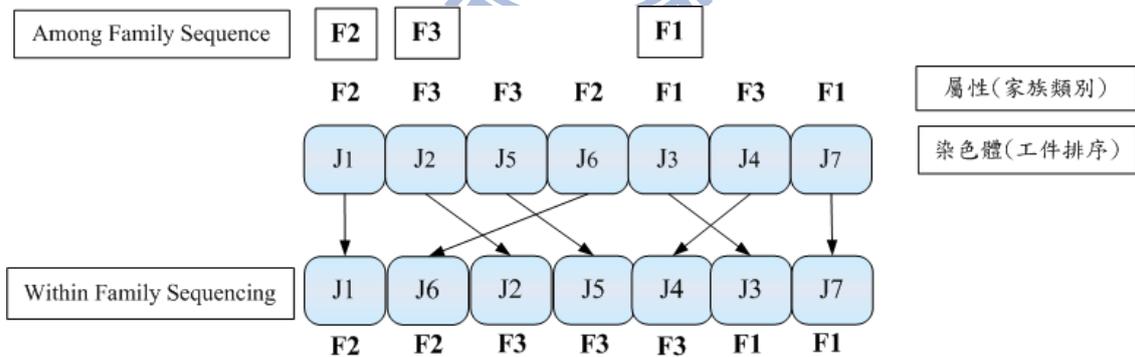


圖 3.4 Within Family Sequence 的解讀範例

Family Assignment

Family 分配至哪一個機台上加工 (Family Assignment) 主要依曝光製程之負荷決定，原因在於曝光製程為傳輸整合步進機之瓶頸製程，故其為該機台產出之關鍵，且以負荷準則作分配，可均分機台之產能，避免部分機台產能分配過大，使得平均產出

降低。而分配之方法為：累加工件於中曝光機的加工時間，得到總加工時間，再將總加工時間平均分給各機台，因此可獲得一部機台平均加工時間。以家族為單位，累積家族工件於曝光機的加工時間，當累積至最接近機台平均加工時間時，即為分割點，取分割點前段的基因排列組合於第一台機。再重新累積未分配之家族工件於曝光機之加工時間，依照上述之原則進行分配，直到分配至最後一台機台。

以處理二部機台為例，假設有七個工件欲進行排程，所屬的家族類別有四種，隨機產生一組染色體（工件加工順序），透過 Family-based 解讀的概念決定家族在各個機台內的排序 (Among Family Sequence) 和家族內各工件的加工順序 (Within Family Sequence) 後，接著進行各工件族之分配，如圖 3.5 所示，首先累積工件於曝光機的加工時間，獲得總加工時間為 1.8 分鐘，再將其平均分配給二台機台，可知機台平均之加工時間為 0.9 分鐘，由於家族不可分割，故以家族為單位，累積家族在曝光機之時間，獲得家族 2 之總加工時間為 0.4 分鐘，家族 2 和家族 3 之總加工時間為 1.2 分鐘，由於 1.2 分鐘較為接近機台平均之加工時間 0.9 分鐘，故作為分配之分割點，將家族 2 和家族 3 之工件分配至機台一，加工順序為 $J_1 \rightarrow J_6 \rightarrow J_2 \rightarrow J_5 \rightarrow J_4$ ；而剩下的家族 1 則分配至機台二，加工順序為 $J_3 \rightarrow J_7$ ；處理三部機台時，亦依據相同分割原則進行分配。

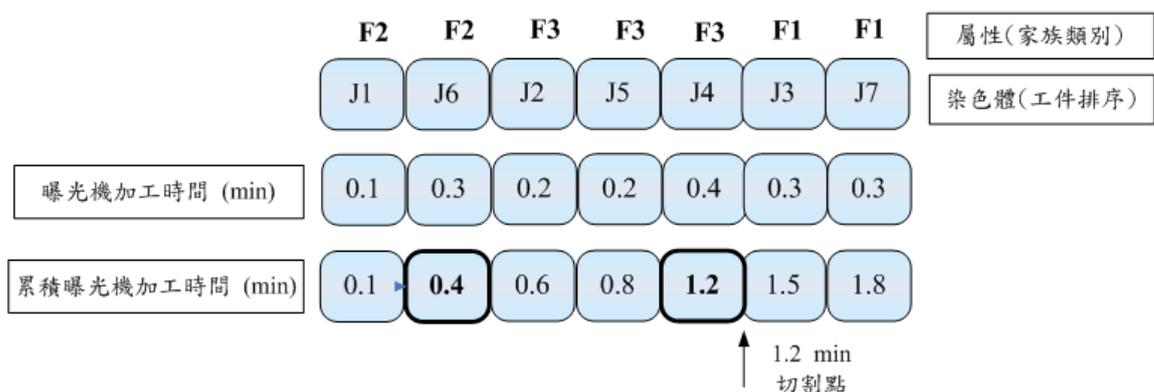


圖 3.5 Family Assignment 的解讀範例

3.3.3 適合度函數

染色體會依照研究問題之定義，計算其適合度函數值（績效指標）以評估各條染色體之好壞，故可知適合度函數為評估各種解之優劣依據，即適合度函數值可代表染色體適應環境之能力，稱為解題品質 (solution quality)。因此當適合度函數值佳，代表染色體的適應力較強，存活機率也較高，即解較好。

由於本研究是探討多傳輸整合步進機排程問題，決策變數為工件的派工和排序，目標為最小化機台總完工時間，使機台平均產出最大，故染色體所代表的為不同之派工和排程方式，而適合度函數為機台平均產出率最大，此值為大，以二台機台為例說明，如表 3.1 所示，故機台一總完工時間為 T_1 ，在此完工時間內所完成加工之晶圓片數為 w_1 ，因此其產出為 $\frac{w_1}{T_1}$ ；機台二總完工時間為 T_2 ，在此完工時間內所完成加工之晶圓片數為 w_2 ，其產出為 $\frac{w_2}{T_2}$ ，而本研究的適合度函數數值為最大化平均機台之產出，故計算試子為 $Max\left\{\frac{1}{2}\left(\frac{w_1}{T_1} + \frac{w_2}{T_2}\right)\right\}$ ，化為一般式子為 $Max\left\{\frac{1}{X} \sum_{x=1}^X \frac{w_x}{T_x}\right\}$ ，其中 X 代表機台數， x 代表一部傳輸整合步進機。

表 3.2 適合度函數值之範例

機台分類	總完工時間 (min)	晶圓片 (片)	產出 (片/min)
機台一	T_1	w_1	$\frac{w_1}{T_1}$
機台二	T_2	w_2	$\frac{w_2}{T_2}$

於本研究中，首先藉由前面章節介紹的染色體解讀法，將染色體解讀成三項重要決策，分別為 Family 需要分配給哪一個機台 (Family Assignment)、Family 在各個機台內的排序 (Among Family Sequence) 以及各 Family 內各工件的順序 (Within Family Sequencing)。接著依據各工件在各機台之加工時間和整備時間的資訊，計算並得知各個機台之總完工時間。最後，再平均各機台在總完工時間內之產出作為本研究之適合

度函數，代表解題品質，即評估不同之派工和排程方式的根據，本研究主要利用 Visual C++ 撰寫求解適合度函數的程式，求出染色體的適應值，上述求解之過程如圖 3.6 所示。



圖 3.6 求解染色體的適應式值



3.3.4 基因演算法求解過程

圖 3.7 為基因演算法的求解流程，首先設計問題解的表示方式進行編碼，再以隨機或是啟發式的方法產生初始的母體，並計算與評估母代中每一個染色體之適應函數值，接著評估此解是否滿足終止條件，若滿足，則停止基因演算法，即代表此解為最終近似最佳解；反之，則透過交配、突變等演化方式產生下一代（稱為子代），依據選擇策略從染色體池 (pool) 中選取下一世代之演化個體，週而 始得進行一代一代的演化，直至達到設定之終止條件，此解即為最終近似最佳解。

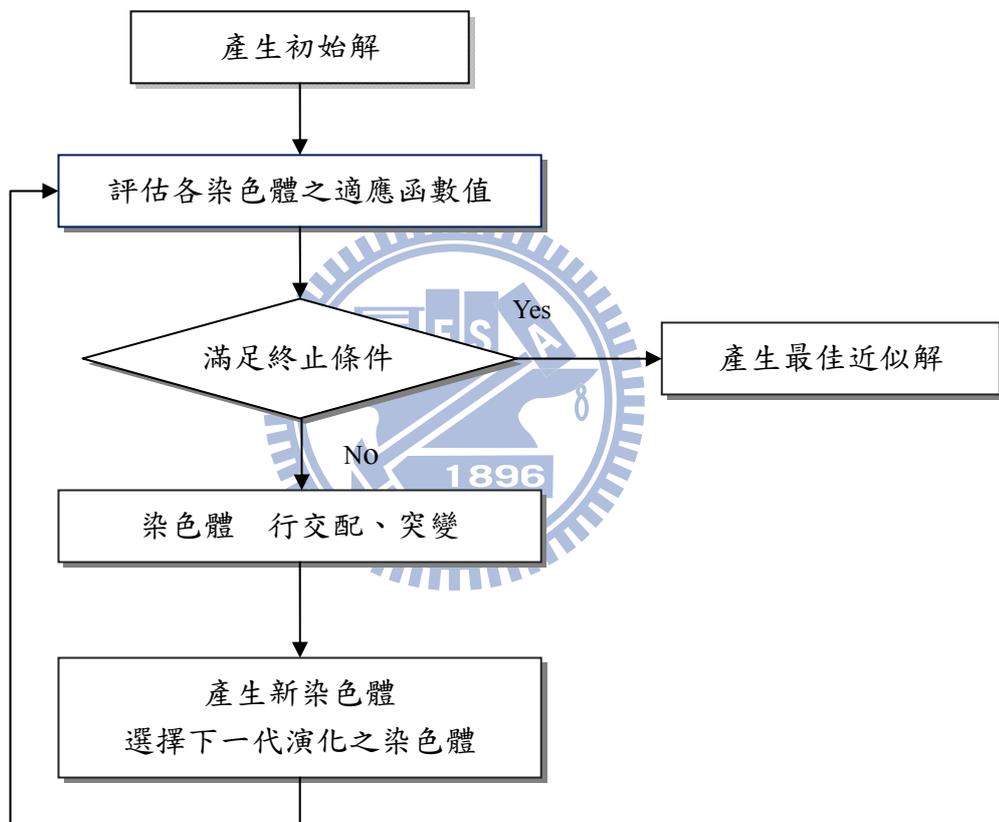


圖 3.7 基因演算之求解流程

染色體交配

交配 (Crossover) 是隨機在個體間相互 合與交換 此特 的過程，其為基因演算法中主要的運算元。交配主要為隨機選擇成對之染色體，進行部分基因交換，以增產更好的染色體。換句話說，交配為透過母代 (Parent) 染色體間之訊息交換，產生二條具有親代特性的子代 (Child) 染色體，以達到基因演算法搜尋之目的。本研究所使用的交配方式有三種，各個交配運算之方法，整理如下所示：

單點交配 (C1 Operator) 此法由 Reeves (1995) 提出，以圖 3.8 為例，步驟說明：

步驟一. 隨機挑選兩條母代染色體，並於此兩條母代染色體中，隨機選取一個 點，將染色體皆分成二部分。

步驟二. 將子代 1 繼 母代 1 點前之基因格 (J_1, J_2)，子代 2 繼 母代 2 兩 點前之基因格 (J_3, J_5)。

步驟三. 選取母代 2 中未和子代 1 重複之基因 (J_7, J_4, J_3, J_5, J_6)，由 往 依序入子代 1 之空 基因格，完成新的子代染色體之產生 ($J_1, J_2, J_7, J_4, J_3, J_5, J_6$)。

步驟四. 選取母代 1 中未和子代 2 重複之基因 (J_1, J_2, J_4, J_6, J_7)，由 往 依序入子代 2 之空 基因格，完成新的子代染色體之產生 ($J_3, J_5, J_1, J_2, J_4, J_6, J_7$)。

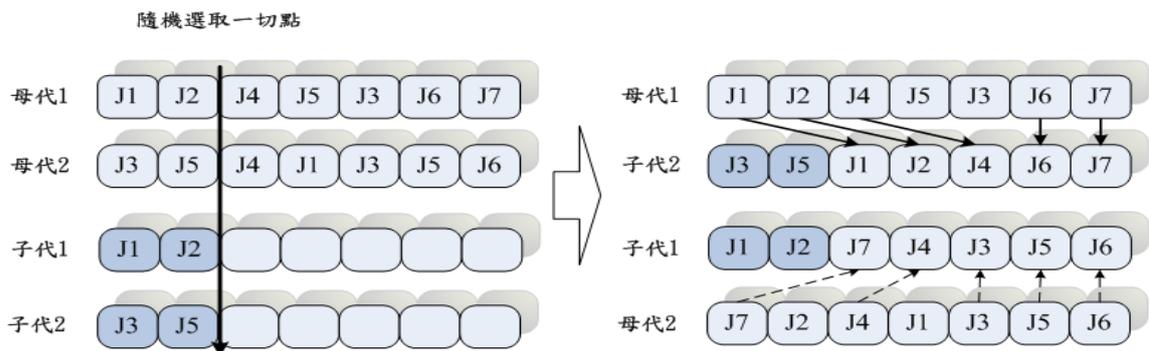


圖 3.8 染色體交配：C1 運算

線性順序交配 (LOX: linear Order Crossover Operator) ，此法由 Croce et al (1995) 提出。LOX 之染色體交配方式，以圖 3.9 為例，進行步驟式說明：

步驟一. 隨機挑選兩條母代母代染色體，並於此兩條母代染色體中，隨機選取兩個點，將染色體皆分成三部分。

步驟二. 將子代 1 繼 母代 1 兩 點內之基因格 (J_2, J_5)，子代 2 繼 母代 2 兩 點內之基因格 (J_4, J_1)。

步驟三. 選取母代 2 中未和子代 1 重複之基因 (J_7, J_4, J_1, J_3, J_6)，由 往 依序入子代 1 之空 基因格，完成新的子代染色體之產生 ($J_7, J_4, J_2, J_5, J_1, J_3, J_6$)。

步驟四. 選取母代 1 中未和子代 2 重複之基因 (J_2, J_5, J_3, J_6, J_7)，由 往 依序入子代 2，完成新的子代染色體之產生 ($J_2, J_5, J_4, J_1, J_3, J_6, J_7$)。

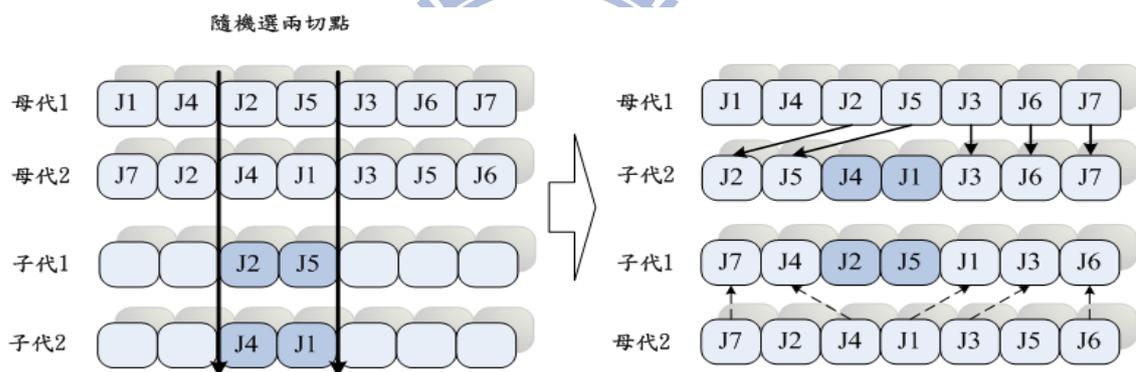


圖 3.9 染色體交配：LOX 運算

部分相應交配 (PMX: Partially matched crossover Operator)，此法由 Goldberg (1989) 提出，以圖 3.10 為例，進行步驟式說明。

步驟一. 隨機挑選兩條母代母代染色體，並於此兩條母代染色體中，隨機選取兩個點，將染色體皆分成三部分。

步驟二. 將子代 1 繼承母代 1 兩點內之基因格 (J_2, J_3, J_4)，子代 2 繼承母代 2 兩點內之基因格 (J_4, J_5, J_3)。

步驟三. 選取母代 2 中未和子代 1 重複之基因 (J_7, J_5, J_1, J_6)，直接填入相對於子代 1 之空基因格，若相對位置非空基因格之基因 (J_5)，則保留並依序分配至子代 1 餘之空，完成新的子代染色體之產生 ($J_7, J_2, J_3, J_4, J_1, J_5, J_6$)。

步驟四. 選取母代 1 中未和子代 2 重複之基因 (J_1, J_2, J_6, J_7)，直接填入相對於子代 2 之空基因格，若相對位置非空基因格之基因 (J_2)，則保留並依序分配至子代 2 餘之空，完成新的子代染色體之產生 ($J_1, J_4, J_5, J_3, J_2, J_6, J_7$)。

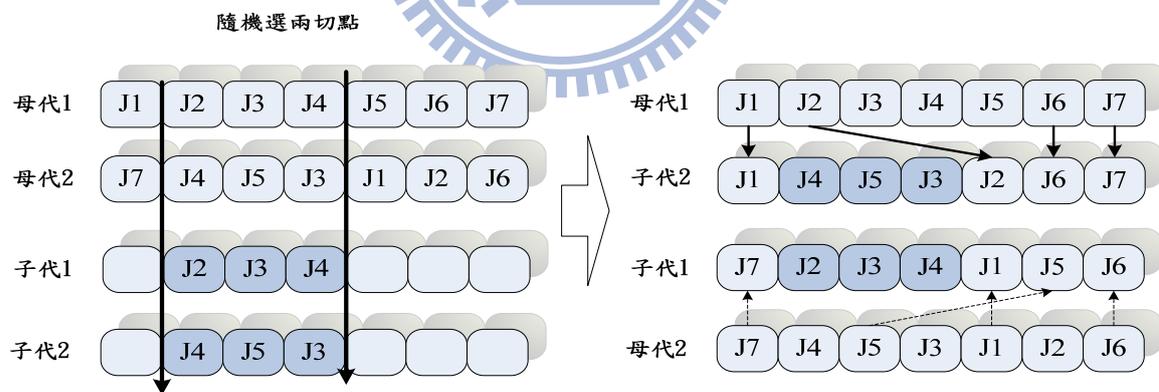


圖 3.10 染色體交配：PMX 運算

染色體突變

突變 (Mutation) 的功能在於增加群體的變異度，以提供新的搜尋方向，使基因演算法避免陷入局部最佳解，因此突變通常會與交配一同使用。本研究所使用的突變方式交換為 (SWAP) 以及反轉 (Inverse) (Wang & Uzsoy, 2002)，交換為在母代染色體中隨機挑選兩個不同之基因格，將其對調後可產生一條新的子代染色體，如圖 3.11 所示。反轉則在母代染色體中隨機選取兩個點，使染色體分割成三部分，再選取中間部分之基因格作前後順序的反轉，以產生一條新的子代染色體，如圖 3.12 所示。

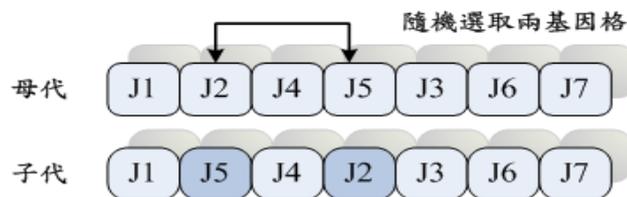


圖 3.11 染色體突變：SWAP 運算

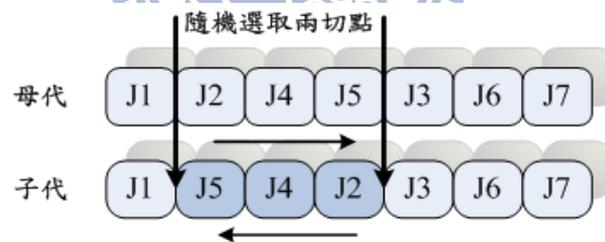


圖 3.12 染色體突變：Inverse 運算

選擇策略

染色體進行交配與突變後，在交配池中，存有原本母代之染色體，以及交配與突變後產生之子代染色體，這些染色體會透過染色體解讀的動作，獲取工件分配至哪一個機台，以及工件在各別機台內的排序之資訊，再照基因演算法物競天擇的概念，依據各別之適合度函數值，決定一些染色體可存活並保留，以更新重組母群體。在染色體選擇策略中，本研究採用策略搭配 Goldberg (1989) 所提出的俄羅斯輪盤法 (Roulette wheel selection)，其主要先保留交配池中最佳的染色體到下一代，其餘的採俄羅斯輪盤法。

終止條件

基因演算在程式正常運行下，母群體演化過程為趨向全域最適解發展，且當演化的世代數越長，其染色體可進化的越佳，但電行所花費的時間相對的也越長，因此需要定終止條件，使終止條件和最佳解之條件上能顧時間和效率。本研究採用行世代終止法則，並參考過去文獻使用的代數，已知續 1000 代最佳解皆未改變已足代表此解達至收，故本研究假設當最佳解持 1,000 世代都沒有被換，或是演化世代達到 100,000 世代時即終止。

3.3.5 模擬退火法求解過程

圖 3.13 為模擬退火法的求解流程，首先隨機產生一初始解，透過交換產生鄰近解，若此解比現行解好，則接受；反之，則隨機給該鄰近解一個機率，依照此機率之大小判斷是否接受，如不接受，則就目前的解繼續搜尋。當接受的解達到設定的數目時，並予以降溫，再依據上一個溫度之最佳解作為起始解繼續搜尋，直到符合終止條件。

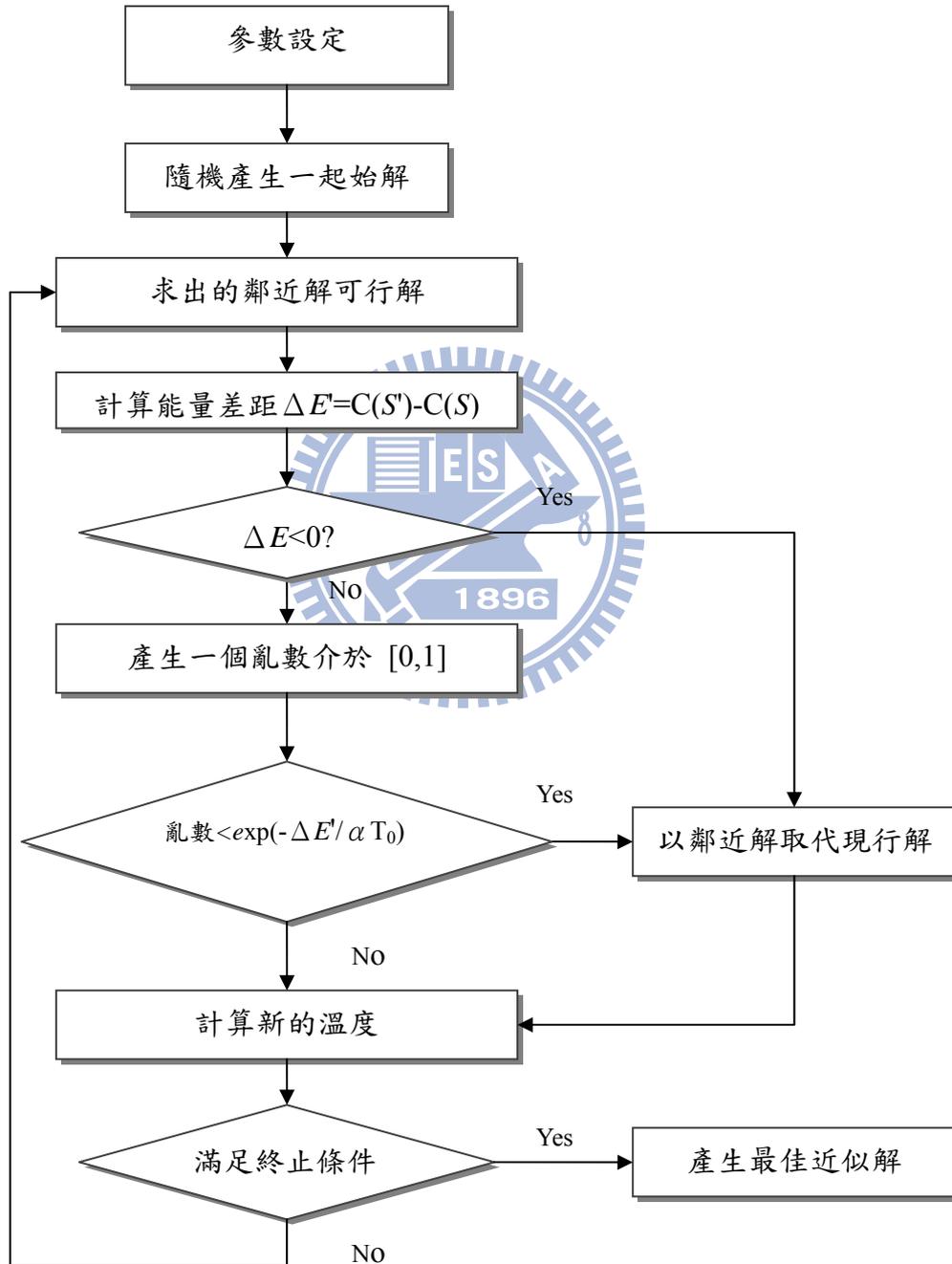


圖 3.13 模擬退火法之求解流程

3.5.6 禁忌搜尋法求解過程

圖 3.14 為禁忌搜尋法的求解流程，首先隨機產生一初始解，透過移步產生鄰近解，若此解比現行解好，則取代現行解且錄在禁忌名單中；反之，檢查此鄰近解是否已存在禁忌名單中，若已在禁忌名單中，則必須重新搜尋，不斷的重複直到符合終止條件，而當名單內成員數達至上限，本研究會依據先進先出的原則更新名單之成員。

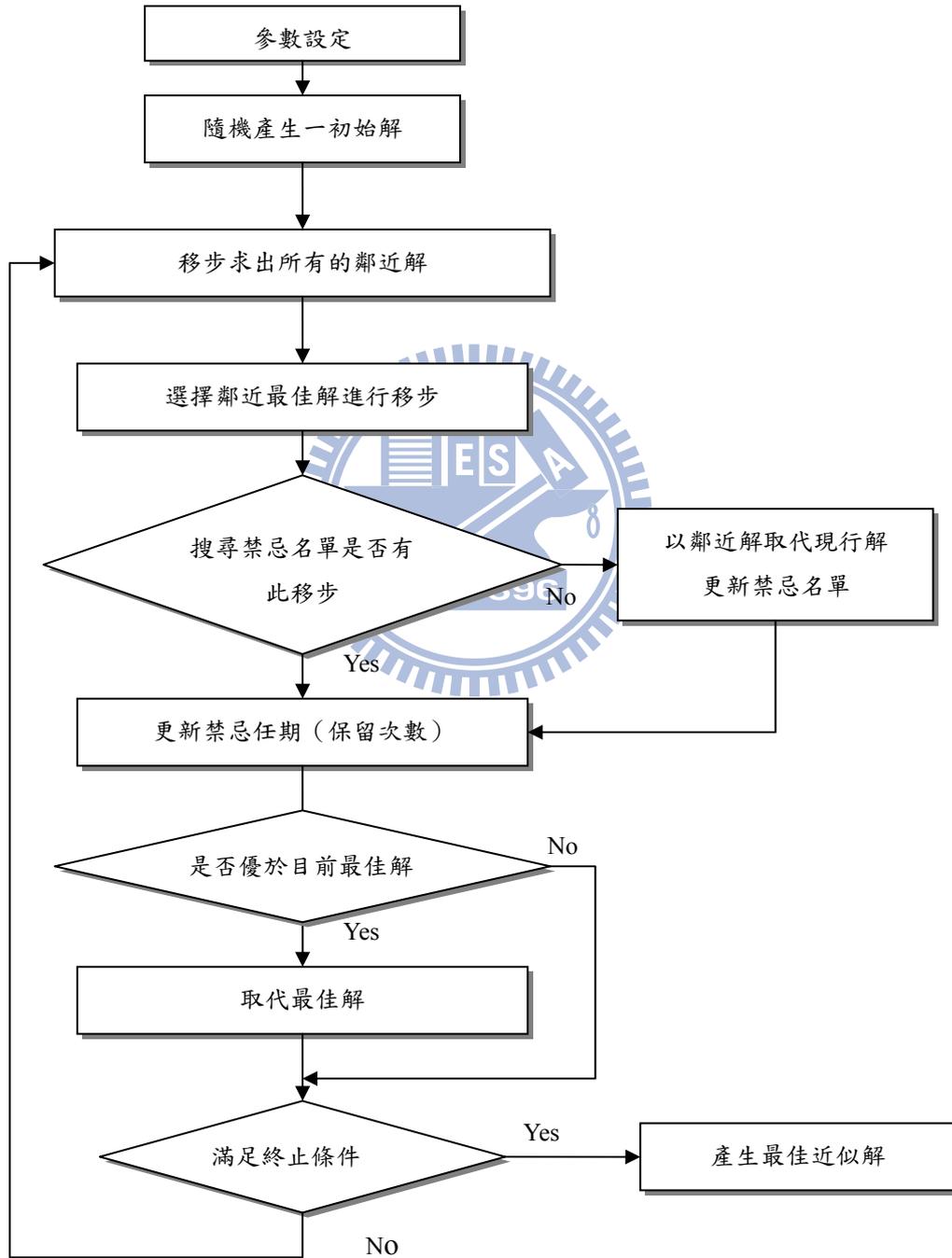


圖 3.14 禁忌搜尋法之求解流程

3.3.7 螞蟻演算法求解過程

圖 3.15 為螞蟻演算法的求解流程，首先將問題轉為路徑之問題形式，產生起始的費洛蒙路徑及其參數，透過局部搜尋改善解，並更新費洛蒙路徑及其強度，直到達至設定之終止條件後，便傳回最佳解。

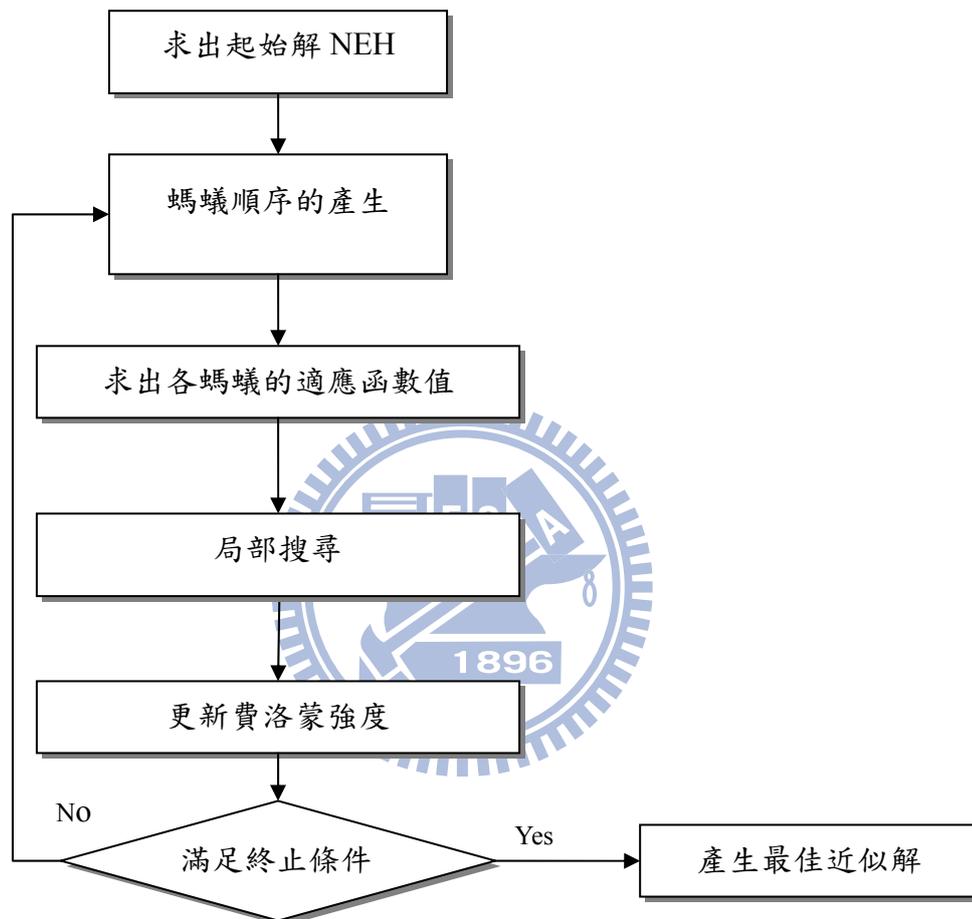


圖 3.15 螞蟻演算法之求解流程

3.3.8 粒子群最佳化求解過程

圖 3.16 為粒子群最佳化求解流程，首先隨機產生初始粒子群，再分別計算每個粒子之適合度，並且將最好之個體經驗記錄下來，每個粒子會依照個體與群體最好之經驗進行每一代的演化，直到符合終止條件為止。

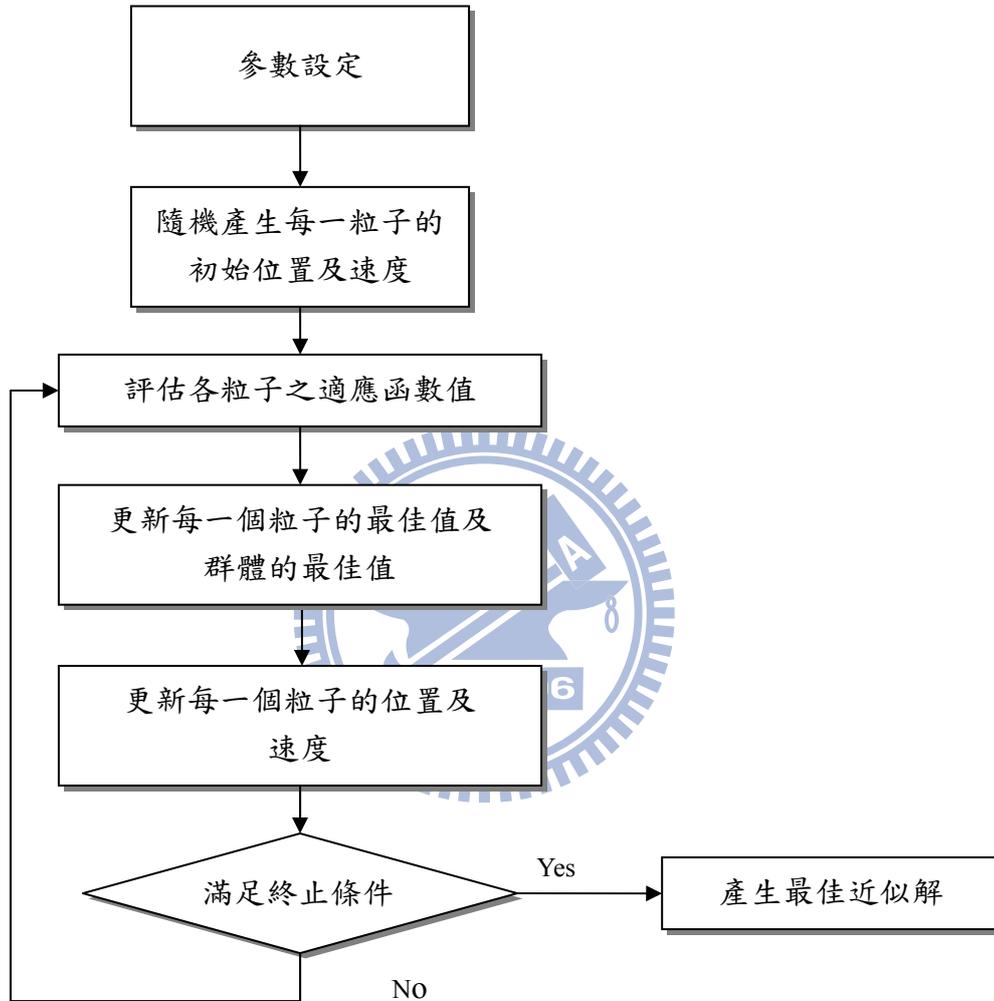


圖 3.16 粒子群最佳化之求解流程

3.3.9 GA-Tabu 求解過程

本研究主要採用 Wu & Chiou (2009) 的 GA-Tabu 演算法，其求解流程如圖 3.17，
詳 的求解步驟和 程序於第二章已作說明。

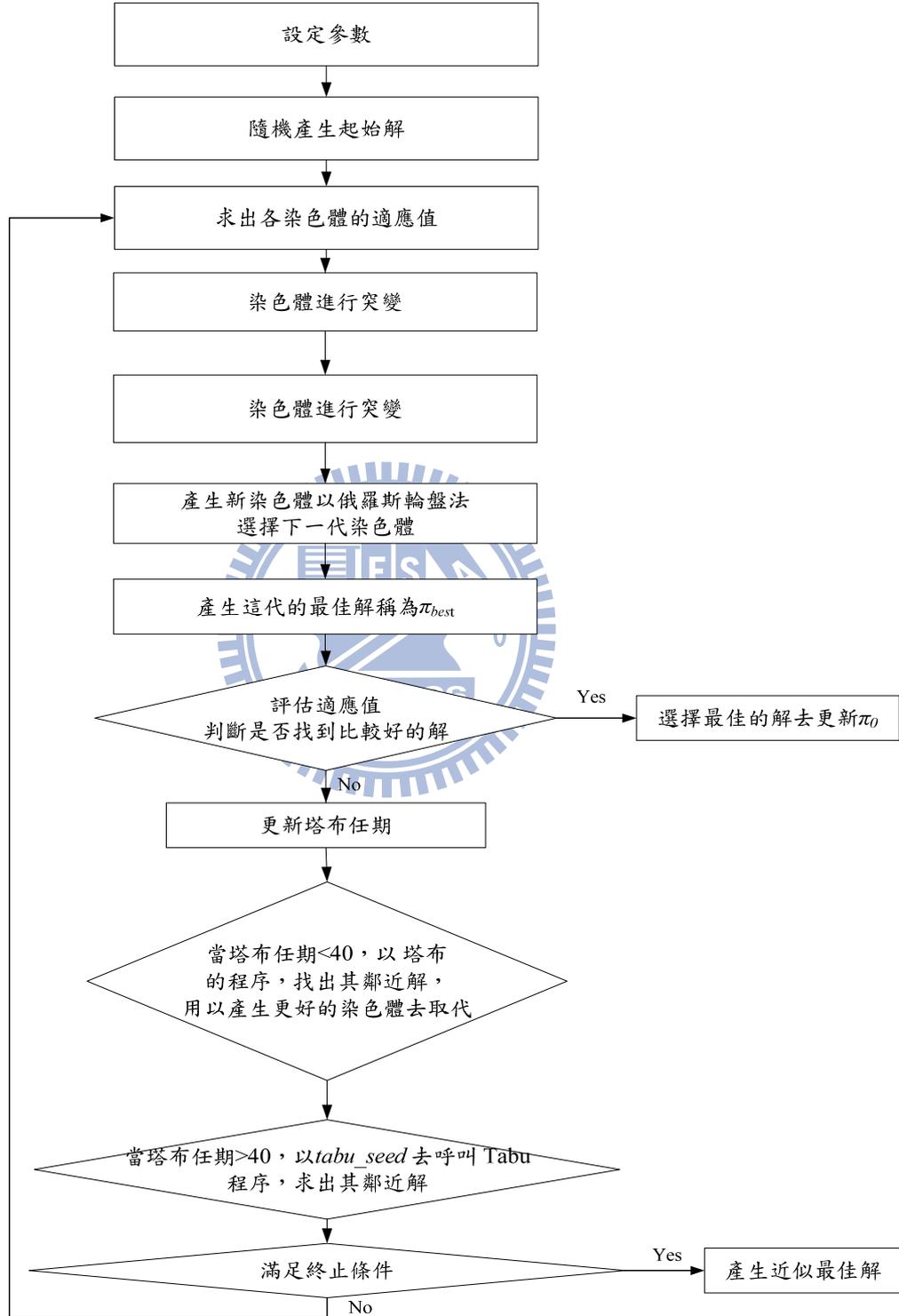


圖 3.17 GA-Tabu 之求解流程

第四章 實例驗證

本章節主要為驗證本研究所提出的七種啟發式演算法，分別於多部傳輸整合步進機的排程問題之有效性，而本章主要分成實驗目的、測試情境設計、演算法參數設定和實驗結果與分析。

4.1 實驗目的

過去 Wu & Chiou (2009) 為研究在小批量情境下，對多傳輸整合步進機的排程問題，其使用的方法並未考慮 Job-family 之情況，亦未將光罩之特性納入考量，故採用單獨式派工 (Individual-based) 的概念，運用七種演算法選出最佳排序，其使用的演算法分別為 GA-Tabu、基因演算法 (GA)、禁忌搜尋法 (Tabu)、模擬退火法 (SA)、螞蟻演算法 (PACO、MMAX) 以及微粒子最佳化法 (PSO)，本研究將以上的方法分別命名為 mGA-Tabu-I (Individual-based GA-Tabu)、mGA-I (Individual-based GA)、mTabu-I (Individual-based Tabu)、mSA-I (Individual-based SA)、mPACO-I (Individual-based PACO)、mMMAX-I (Individual-based MMAX) 和 mPSO-I (Individual-based PSO)，其命名基於各個演算法搭配單獨式派工法則於多機之情境。

因此本研究考慮 Job-family 之情況，將光罩之特性納入考量，將上述七種啟發式演算法結合家族式派工 (Family-based) 的概念，分別將本研究所提出之方法命名為 mGA-Tabu-F (Family-based GA-Tabu)、mGA-F (Family-based GA)、mTabu-F (Family-based Tabu)、mSA-F (Family-based SA)、mPACO-F (Family-based PACO)、mMMAX-F (Family-based MMAX) 和 mPSO-F (Family-based PSO)，其命名基於各個演算法搭配家族式派工法則於多機之情境，並與 Wu & Chiou (2009) 所使用的方法做比較，以驗證使用家族式派工之必要性，接著再進一步量各個演算法間的績效差異。

4.2 測試情境的設計

為了驗證所提出各種演算法的有效性，本研究模擬傳輸整合 進機實際在晶圓廠之運作模式，對於傳輸整合步進機的機台配置與生產作業之假設如下：

1. 傳輸整合步進機配置 4 個埠，具有 14 個站點，共有 22 個反應室。
2. 在每一個反應室的生產時間，假設其值以均 分配 (uniform distribution) 分佈 $[a_i, b_i]$ (表 4.1)，其中曝光反應式之生產時加假設為固定值 (1.0 min)。
3. 傳輸整合步進機的整備時間設定為 1.5 至 6 分鐘。
4. 每一個反應室對於到站加工的晶片，採先到先 務。
5. 不同加工圖 的生產批，其使用的光罩會不同，故換批時即需要換光罩的整備時間。
6. 工件的良率為工件到達暫存區已形成批量之大小，反應室的良率為 100%。
7. 本研究為 態排程。

表 4.1 傳輸整合步進機反應室的加工時間 (資料來源：Wu & Chiou (2009))

加工順序	反應室	加工時間 (min)	加工順序	反應室	加工時間 (min)
(1)暫存區到埠區	1	2.5	(8)曝光	1	1
(2)埠區到塗佈與顯影機	1	0.1	(9)晶片 曝光	1	1
(3)HMDS	2	1.2	(10)曝光後烘烤	2	[1.2,2.8]
(4) 卻室	2	1.2	(11) 卻室	2	1
(5)塗佈	2	1.2	(12)顯影	2	[1.2,2.8]
(6)軟烤	2	[1.2,2.8]	(13)硬烤	2	[1.2,2.8]
(7) 卻室	2	1	(14) 卻室	1	0.5

本研究為多機排程問題，主要分為二部機台和三部機台兩種情境作探討。為了建構 Job 在不同加工 時需要更換光罩的情境，本研究使用 Wu, Lu, & Chiou (2009)的情境假設：

1. 四種不同設置時間之大小，分別為 1.5、3、4.5 以及 6 分鐘。
2. 四種不同光罩數目之大小，分別為 3、5、7 以及 10 種。
3. 二種批數大小，分別為為 20 Jobs、40 Jobs。
4. 十種製程良率的批量大小分配為 0.15、0.2、0.25、0.3、0.4、0.5、0.6、0.7、0.8、0.9。

因此本研究以(M, N, Y, S, F)來代表測試之情境，其中 M 代表機台數目，N 代表批量數之大小，Y 代表製程良率，S 代表設置時間，F 代表 Job 所屬之光罩數目，即可視為 Job 所屬之家族(family)。由於研究假設光罩數目必須大於機台數目，故在三部機台的情境下，光罩數目之大小僅探討 5、7、以及 10 這三種情境，故代表每一個演算法共有 560 種測試情境，而每一個測試情境皆做 15 次取平均，其為了獲得更準確的近似最佳解，避免因隨機實驗，遇到特 情形所產生的 高或 低之 值，而影響最佳解之結果。

由於每一個測試情境皆做 15 次後取平均，故每一種演算法之解品質之定義如下： C_{mGA_Tabu-F} 代表 m GA-Tabu-F 的平均產出之平均； $C_{mTabu-F}$ 代表 mTabu-F 平均產出之平均； C_{mGA-F} 代表 mGA-F 平均產出之平均； $C_{mPACO-F}$ 代表 mPACO-F 平均產出之平均； $C_{mMMAX-F}$ 代表 mMMAX-F 平均產出之平均； C_{mSA-F} 代表 mSA-F 平均產出之平均； C_{mPSO-F} 代表 mPSO-F 平均產出之平均。另外，Wu & Chiou (2009)所提出之演算法之解品質亦以此方式定義，如 C_{mGA_Tabu-I} 代表 m GA-Tabu-I 的平均產出之平均，以此類 出 C_{mGA-I} 、 $C_{mPACO-I}$ 、 $C_{mMMAX-I}$ 、 C_{mSA-I} 與 C_{mPSO-I} 。

每一個演算法所需要的平均電 計算時間以 t 符 代表，如：mGA-Tabu-F 演算法所需要的平均電 計算時間定義為 t_{mGA_Tabu-F} ，其他演算法之平均電 計算時間亦以此方式定義。

4.3 演算法的參數設定

本研究之七種演算法的參數設定為參照 Wu & Chiou (2009)，如表 4.2。

表 4.2 本主題各演算法相關參數之設定值（資料來源：Wu & Chiou (2009)）

演算法	參數項目	數值
GA	交配率	0.8
	突變率	0.2
	代數	100,000
	收 代數	1,000
GA-tabu	交配率	0.8
	突變率	0.2
	禁忌任期	7
	代數	100,000
	收 代數	3,000
Tabu	禁忌任期	7
	代數	150
SA	初始溫度	500
	α	0.9998
M-MMAX	ρ	0.75
	代數	150
PACO	ρ	0.75
	代數	100
PSO	粒子群	60
	V_{\max}	3
	V_{\min}	-3
		0.8
	c	2
	c2	2
	代數	100

4.4 實驗結果與分析

本研究主要探討在小批量情境下作多部傳輸整合步進機的排程問題，針對七種啟發式演算法在每一種情境下，分別結合家族式派工法則 (Family-based) 與結合單獨式派工法則 (Individual-based) 作解題品質之比較，實驗出何種派工法之解題品質較好，接著再針對結果最好的派工法則，進行七種演算法的績效量。

首先針對演算法結合家族式派工法則 (Family-based) 與單獨式派工法則 (Individual-based) 作之解題品質定義一個改善績效指標： $\gamma_x = (C_{xF} - C_{xI}) / C_{xI}$ ，其中 C_{xF} 代表 x 演算法結合家族式派工法則之解品質，而 C_{xI} 代表 x 演算法結合單獨式派工法則之解品質，當 γ_x 越高，表示該演算法結合家族式派工法則較好，反之則否。

表 4.3 主要為兩部機台、工件數為 20 之情境下，各個演算法結合不同派工法則之績效，由表可知除了 γ_{mGA_Tabu-I} 為負值，其餘演算法之改善率皆為正值，又屬 γ_{mSA-I} 的值最高，故可知於此情境下，GA-tabu 結合家族式派工法則在解品質上有未有改善，而 SA 結合家族式派工法則在解品質上有大的改善。整體來說，演算法結合家族式派工法則在此情境下均有不一的表現，且當良率為 0.15 時，改善率有顯著的差異。

表 4.3 (M,N) = (2,20)，演算法與不同派工法則之差異

Yield	γ_{mGA_Tabu-I}	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	2.00%	0.00%	0.00%	0.00%	2.00%	0.00%
80%	-1.96%	0.00%	0.00%	0.00%	0.00%	4.08%	2.00%
70%	0.00%	0.00%	0.00%	0.00%	0.00%	4.17%	2.04%
60%	-2.00%	-2.00%	-2.00%	-2.00%	-2.00%	4.26%	0.00%
50%	-2.04%	0.00%	0.00%	0.00%	0.00%	8.89%	2.08%
40%	-6.67%	0.00%	0.00%	0.00%	0.00%	15.38%	4.65%
30%	-2.13%	0.00%	0.00%	0.00%	0.00%	11.90%	2.17%
25%	-4.76%	0.00%	0.00%	0.00%	0.00%	17.14%	5.13%
20%	-7.69%	0.00%	0.00%	2.63%	2.63%	18.75%	5.56%
15%	0.00%	2.86%	2.86%	2.86%	0.00%	40.00%	6.25%
Average	-2.73%	0.29%	0.09%	0.35%	0.63%	12.66%	2.99%

表 4.4 主要為兩部機台、工件數為 40 之情境下，各個演算法結合不同派工法則之績效，由表可知各演算法之改善率皆為正值， γ_{mSA-I} 的改善率亦為最高，故可知於此情境下，演算法結合家族式派工法則皆有好的改善，且當良率越低時，改善率越有顯著的差異。

表 4.4 (M,N) = (2,40)，演算法與不同派工法則之差異

Yield	γ_{mGA_Tabu-I}	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	1.92%	0.00%	1.92%	0.00%	1.92%	1.92%
80%	0.00%	0.00%	0.00%	0.00%	0.00%	1.96%	1.96%
70%	0.00%	1.96%	0.00%	0.00%	0.00%	4.00%	1.96%
60%	0.00%	1.96%	0.00%	1.96%	0.00%	6.12%	4.00%
50%	1.96%	0.00%	0.00%	0.00%	0.00%	6.25%	4.08%
40%	2.04%	0.00%	0.00%	4.26%	2.08%	11.36%	8.89%
30%	0.00%	2.00%	0.00%	4.08%	2.00%	8.70%	6.38%
25%	0.00%	0.00%	0.00%	6.67%	4.35%	11.90%	9.30%
20%	0.00%	0.00%	2.17%	6.82%	6.82%	15.00%	12.20%
15%	0.00%	2.63%	2.63%	8.33%	5.41%	15.15%	8.82%
Average	0.40%	1.05%	0.48%	3.40%	2.07%	8.24%	5.95%

表 4.5 主要為三部機台、工件數為 20 之情境下，各個演算法結合不同派工法則之績效，由表可知除了 γ_{mSA-I} 為正值，其餘演算法之改善率皆為負值，故整體來說，在此情境下演算法結合單獨式派工法則會比演算法結合家族式派工法則來得好。

表 4.5 (M,N) = (3,20)，演算法與不同派工法則之差異

Yield	γ_{mGA_Tabu-I}	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	-2.00%	-2.04%	-2.00%	-2.00%	-2.00%	0.00%	0.00%
80%	-2.00%	-4.00%	-2.00%	-2.00%	-2.00%	2.08%	0.00%
70%	-2.04%	-2.04%	-2.04%	-2.04%	-2.04%	2.13%	0.00%
60%	-2.08%	-2.08%	-2.08%	-2.08%	-2.08%	2.17%	0.00%
50%	0.00%	0.00%	0.00%	0.00%	0.00%	2.22%	0.00%
40%	-2.22%	0.00%	-2.22%	-2.22%	-2.22%	2.38%	2.33%
30%	-2.33%	-2.33%	0.00%	0.00%	0.00%	2.56%	2.50%
25%	-2.56%	0.00%	-2.56%	0.00%	-2.56%	2.86%	2.78%
20%	-2.78%	-2.78%	-2.78%	0.00%	0.00%	6.25%	3.03%
15%	0.00%	-3.03%	0.00%	0.00%	0.00%	6.90%	3.33%
Average	-1.80%	-1.83%	-1.57%	-1.03%	-1.29%	2.96%	1.40%

表 4.6 主要為三部機台、工件數為 40 之情境下，各個演算法結合不同派工法則之績效，由表可知演算法之改善率皆為正值， γ_{mSA-I} 和 γ_{mPSO-I} 的值皆高於 5%，故可知於此情境下，演算法結合家族式派工法則均有不 的表現，且又以 SA 和 PSO 結合家族式演算法表現最好，且當良率越低時，改善率越有顯著的差異。

表 4.6 (M,N) = (3,40)，演算法與不同派工法則之差異

Yield	γ_{mGA_Tabu-I}	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	6.12%	0.00%	0.00%	0.00%	1.96%	1.96%
80%	0.00%	2.00%	-1.92%	0.00%	0.00%	2.00%	2.00%
70%	0.00%	2.00%	0.00%	0.00%	0.00%	4.08%	2.00%
60%	0.00%	2.00%	0.00%	2.00%	2.00%	4.17%	2.04%
50%	0.00%	0.00%	0.00%	2.04%	0.00%	4.26%	6.38%
40%	0.00%	0.00%	0.00%	4.26%	2.08%	9.09%	6.67%
30%	2.13%	0.00%	0.00%	4.44%	2.17%	9.76%	6.98%
25%	0.00%	0.00%	2.22%	6.98%	2.27%	10.26%	10.00%
20%	2.27%	0.00%	2.27%	7.32%	2.33%	10.53%	7.69%
15%	0.00%	2.78%	5.56%	8.82%	2.86%	12.90%	9.38%
Average	0.44%	1.49%	0.81%	3.59%	1.37%	6.90%	5.51%

由以上的表格可獲得的結論歸納如下：

- 當欲排序的工件數目越多，家族式派工法則的效果越顯著。因為工件數目越多，加工時所需要更換光罩的次數也越多，設置的次數也更多，故演算法結合家族式派工法則可以減少更多餘的設置時間。
- 當欲排序工件的良率越低，家族式派工法則的效果越顯著。因為良率越低時，總完工時間會越小，此時設置時間對於總完工時間的影響就會越大，由於演算法結合家族式派工法則減少了多餘的設置時間，故其在低良率時成效會比演算法結合單獨式派工法則來得顯著。

由於演算法結合家族式派工法則在大部份的情境下，會優於演算法結合單獨式派工法則，故本研究將針對演算法結合家族式派工法則，進行七種演算法的績效量。針對演算法結合家族式派工法則 (Family-based) 作之解題品質，定義一個改善績效指標： $\gamma_x = (C_{mGA-Tabu-F} - C_x) / C_x$ ，其中 $C_{mGA-Tabu-F}$ 代表 $mGA-Tabu-F$ 之解品質，而 C_x 代表其他六種演算法結合家族式派工法則之解品質，當 γ_x 越高，表示該 $mGA-Tabu-F$ 法則較好，反之則否。

表 4.7 主要為兩部機台、工件數為 20 之情境下，各個演算法結合不同派工法則之績效，由表可知各演算法之改善率皆為負值，故可知於此情境下， $mGA-Tabu-F$ 之解品質較其他的演算法來得差。

表 4.7 (M,N) = (2,20)，各個演算法間之解品質差異

Yield	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
80%	-1.96%	-1.96%	-1.96%	-1.96%	-1.96%	-1.96%
70%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50%	-2.04%	-2.04%	-2.04%	-2.04%	-2.04%	-2.04%
40%	-6.67%	-6.67%	-6.67%	-6.67%	-6.67%	-6.67%
30%	-2.13%	-2.13%	-2.13%	-2.13%	-2.13%	-2.13%
25%	-4.76%	-4.76%	-4.76%	-4.76%	-2.44%	-2.44%
20%	-7.69%	-7.69%	-7.69%	-7.69%	-5.26%	-5.26%
15%	-2.78%	-2.78%	-2.78%	0.00%	0.00%	2.94%
Average	-2.80%	-2.80%	-2.80%	-2.53%	-2.05%	-1.76%

表 4.8 主要為兩部機台、工件數為 40 之情境下，各個演算法結合不同派工法則之績效，由表可知各演算法之改善率皆為正值，故可知於此情境下，*mGA-Tabu-F* 之解品質較其他的演算法來得好。

表 4.8 (M,N) = (2,40)，各個演算法間之解品質差異

Yield	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
80%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
70%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50%	1.96%	1.96%	1.96%	1.96%	1.96%	1.96%
40%	0.00%	2.04%	2.04%	2.04%	2.04%	2.04%
30%	0.00%	0.00%	0.00%	0.00%	2.00%	2.00%
25%	0.00%	0.00%	0.00%	0.00%	2.13%	2.13%
20%	0.00%	0.00%	0.00%	0.00%	2.17%	2.17%
15%	0.00%	0.00%	0.00%	0.00%	2.63%	5.41%
Average	0.20%	0.40%	0.40%	0.40%	1.29%	1.57%

表 4.9 主要為三部機台、工件數為 20 之情境下，各個演算法結合不同派工法則之績效，由表可知各演算法之改善率皆為正值，故可知於此情境下，*mGA-Tabu-F* 之解品質較其他的演算法來得好，且改善率皆大於 0.25%。

表 4.9 (M,N) = (3,20)，各個演算法間之解品質差異

Yield	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	2.08%	0.00%	0.00%	0.00%	0.00%	0.00%
80%	2.08%	0.00%	0.00%	0.00%	0.00%	0.00%
70%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50%	0.00%	0.00%	0.00%	0.00%	2.17%	2.17%
40%	-2.22%	0.00%	0.00%	0.00%	2.33%	0.00%
30%	0.00%	0.00%	0.00%	0.00%	5.00%	2.44%
25%	-2.56%	0.00%	0.00%	0.00%	5.56%	2.70%
20%	0.00%	0.00%	0.00%	0.00%	2.94%	2.94%
15%	3.13%	3.13%	3.13%	3.13%	6.45%	6.45%
Average	0.25%	0.31%	0.31%	0.31%	2.44%	1.67%

表 4.10 主要為三部機台、工件數為 40 之情境下，各個演算法結合不同派工法則之績效，由表可知各演算法之改善率皆為正值，故可知於此情境下，*mGA-Tabu-F* 之解品質較其他的演算法來得好。

表 4.10 (M,N) = (3,40)，各個演算法間之解品質差異

Yield	$\gamma_{mTabu-I}$	γ_{mGA-I}	$\gamma_{mPACO-I}$	$\gamma_{mMMAX-I}$	γ_{mSA-I}	γ_{mPSO-I}
90%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
80%	1.96%	0.00%	0.00%	0.00%	0.00%	0.00%
70%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
60%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
50%	0.00%	1.96%	1.96%	1.96%	1.96%	1.96%
40%	0.00%	2.04%	2.04%	2.04%	2.04%	2.04%
30%	0.00%	0.00%	0.00%	0.00%	2.00%	2.00%
25%	0.00%	0.00%	0.00%	0.00%	2.13%	2.13%
20%	0.00%	0.00%	0.00%	0.00%	2.17%	2.17%
15%	0.00%	0.00%	0.00%	0.00%	2.63%	5.41%
Average	0.20%	0.40%	0.40%	0.40%	1.29%	1.57%

由以上的表格可獲得的結論：在多機情境下，除了兩部機台、工件數為 20 之情境下，*mGA-Tabu-F* 演算法的運算品質績效比其他演算法結合家族式派工法則有更優的績效。

第五章 結論與未來研究方向

5.1 研究的結論

對半導體晶圓廠來說，傳輸整合步進機是單價最貴的機台，因此成為工廠之瓶頸機台，故傳輸整合步進機之排程之議題相當重要，然而在過去的研究上，多在滿批量之情境作探討，因此往往將其視為一個機台，而未了解該機台之內部結構，可視為一種流程式生產型態；隨著新製程和新產品之導入，晶圓廠出現小批量之情境，使得傳輸整合步進機會因為機台結構之設計，造成產能之損失，若一間晶圓廠綜合了新製程和新產品之情況，小批量之情境將漸趨增加，也就意味著產能損失之發生率將隨之增加，而過去對於小批量情境之傳輸整合步進機的排程研究僅有 Wu & Chiou(2009) 和 Wu, Lu, & Chiou (2009)作探討。

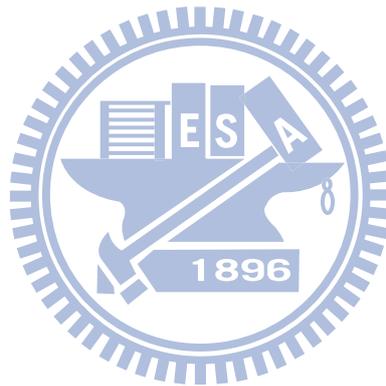
Wu & Chiou (2009) 所探討的情境為單機和多機，皆未考慮 Job-family 之情況，即未將光罩之限制納入考量，其透過七種演算法搭配單獨式派工法則進行求解； Wu, Lu, & Chiou (2009)為研究單機且考慮 Job-family 之情境，運用基因演算法搭配家族式派工法則進行求解。然而實際的半導體晶圓廠存在著多機和 Job-family 之情境，故本研究提出之啟發式演算法搭配家族式派工法則可以在短時間內，得到一組近似最佳解（工件加工順序），使得平均產出最大。而在經過大量的實驗後，可以將本研究的結論歸納如下：

- 當欲排序的工件數目越多及排序工件的良率越低，應使用演算法結合家族式派工法則。
- 在處理多個步進機的排序問題，本研究提出新的染色體設計，可在不增加染色體的大小情況下，減少解題空間，使得解題品質及求解的速度獲得改善。
- *mGA-Tabu-F* 演算法的運算品質績效在大部份的情境下，比其他演算法結合家族式派工法則有更優異的績效表現。

5.2 未來研究方向

本研究的未來研究方向，整理如下列數項：

1. 本研究以啟發式演算法 (meta-heuristic algorithm) 找出近似解，因此未來可以試運用其他的方法求解，如：合整數規劃、動態規劃或分界限法等。
2. 本研究主要考慮以最大產出為主，但對於部分代工晶圓廠之生產目標則主要以達交率為目標，故可以交為目標式作為探討。
3. 由於在小批量情境下，發生產能閒置問題的主要原因為 Port 數量不足，故未來可針對 Port 數量作研究，如：探討 Port 數量應設定多少，方可使傳輸整合步進機不會發生產能損失的問題。
4. 發展新的演算法。



參考文獻

Cheng TCE, Gupta NDJ and Wang G (2000). A review of flowshop scheduling research with setup times. *Production and Operations Management* **9**: 262-282.

Chern CC and Liu YL (2003). Family-based scheduling rules of a sequence-dependant wafer fabrication system. *IEEE ctions on semiconductor manufacturing* **16**: 15-25.

Croce FD, Tadei R and Volta G (1995). A genetic algorithm for the job shop problem. *Computer Operation Research* **22**: 15-24.

Dorigo M (1992). Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy

Franca PM, Gupta JND, Mendes AS, Moscato P and Veltink KJ (2005). Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers & Industrial Engineering* **48**: 491-506.

Glover F (1989). Tabu Search - Part 1, *ORSA Journal on Computing* **1**: 190-206.

Goldberg DE (1989). Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston.

Gonzalez T and Sahni S (1978). Flowshop and jobshop schedules: complexity and approximation. *Operations Research* **26**: 36-52.

Kennedy J and Eberhart R (1995). Particle swarm optimization. Proceedings of IEEE

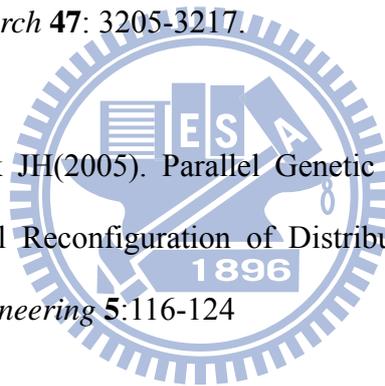
International Conference on Neural Networks-IV:1942-1948.

Holland JH (1975). Adaptation in Neural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan.

Kirkpatrick S, GELATT JR, CD and VECCHI MP (1983). Optimization by simulated annealing. *Science* **220**: 671-680.

Lin SW, Gupta JND, Ying KC and Lee ZJ (2009). Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times. *International Journal of Production Research* **47**: 3205-3217.

Mun KJ, Lee HS and Park JH(2005). Parallel Genetic Algorithm-Tabu Search Using PC Cluster System for Optimal Reconfiguration of Distribution Systems. *KIEE International Transactions on Power Engineering* **5**:116-124



Potts CN and Kovalyov MY (2000). Scheduling with batching: A review. *European Journal of Operational Research* **120**: 228-149.

Quirk M and Serda J (2001). Semiconductor manufacturing technology. Prentice Hall, New Jersey.

Rajendran RC and Hans Z (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research* **155**: 426-438.

Reeves CR (1995). A genetic algorithm for flowshop sequencing. *Computer Operation*

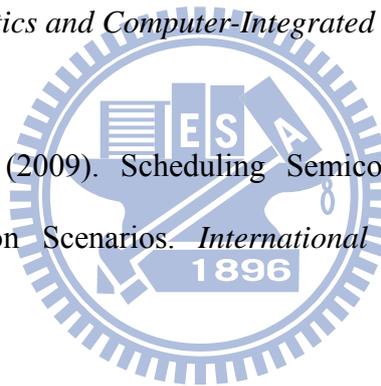
Research **22**: 5-13.

Sundarambal M, Chithara BJ and Anbalagan P (2008). Meta-Heuristic Techniques for Bandwidth Optimization through Dynamic Routing in ATM networks. *International Journal of Soft Computing* **3**: 189-194.

Wang CS and Uzsoy R (2002). A genetic algorithm to minimize maximum lateness on a batch processing machine. *Computers & Operations Research* **29**: 1621-1640.

Wilson AD, King RE and Hodgson TJ (2004). Scheduling non-similar groups on a flow line: multiple group setups. *Robotics and Computer-Integrated Manufacturing* **20**: 505-515.

Wu MC and Chiou CW (2009). Scheduling Semiconductor In-line Steppers in New Product/Process Introduction Scenarios. *International Journal of Production Research* (accepted).



Wu MC, Chiou SJ and Chen CF (2008). Dispatching for make-to-order wafer fabs with machine-dedication and mask set-up characteristics. *International Journal of Production Research* **46**: 3993 - 4009.

Wu MC, Huang YL, Chang YC and Yang KF (2006). Dispatching in semiconductor fabs with machine-dedication features. *International Journal of Advanced Manufacturing Technology* **28**: 978-984.

Wu MC, Jiang JH and Chang WJ (2008). Scheduling a hybrid MTO/MTS semiconductor fab with machine-dedication features. *International Journal Production Economics* **112**: 416-426.

Wu MC, Lu CW and Chiou CW (2009). A Family-based GA Algorithm for Scheduling an In-line Stepper in a Low-Yield Scenario. *Submitted to IEEE Transactions on Semiconductor Manufacturing* (under review).

Ying KC and Lin SW (2009). Raising the hit rate for wafer fabrication by a simple constructive heuristic. *Expert Systems with Applications* **36**: 2894-2900.

