

國立交通大學
工業工程與管理學系

碩士論文

應用基因演算法與家族式派工於傳輸整合
步進機在小批量情境下之排程問題

The logo of National Central University (NCU) is a circular emblem with a gear-like border. Inside the circle, there is a stylized figure of a person holding a torch, with the year '1896' at the bottom. The logo is semi-transparent and serves as a background for the English title.

**A family-based GA algorithm for
scheduling in-line stepper in small-lot
scenarios**

研 究 生：呂佳玟
指 導 教 授：巫木誠教授

中華民國九十八年六月

應用基因演算法與家族式派工於傳輸整合步進機在小

批量情境下之排程問題

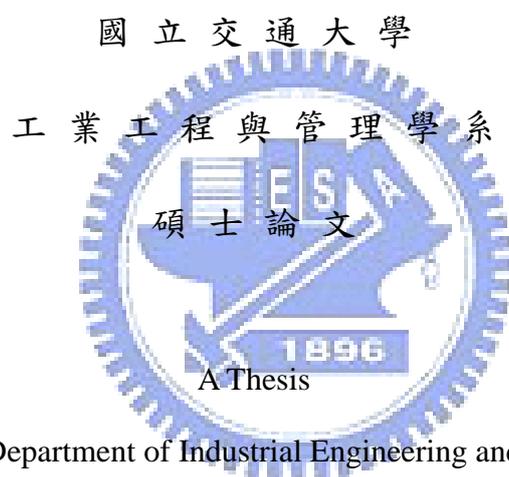
**A family-based GA algorithm for scheduling
in-line stepper in small-lot scenarios**

研究生：呂佳玟

Student：Chia-Wen Lu

指導教授：巫木誠 博士

Advisor：Dr. Muh-Cherng Wu



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 2009

Hsin-chu, Taiwan, Republic of China

中華民國九十八年六月

應用基因演算法與家族式派工於傳輸整合步進機在小 批量情境下之排程問題

研究生：呂佳玟

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

傳輸整合步進機為半導體工廠中的瓶頸機台，在小批量工件的情境下具有產能損失的問題。加上傳輸整合步進機在加工不同的電路佈局時需要更換光罩，若有過多的光罩更換次數也會導致生產效率不佳。為了解決上述議題，本論文發展一個新的排程方法，結合家族式派工與基因演算法 (Family-based Genetic Algorithm, GA-F)，來提高傳輸整合步進機的生產力。此排程方法可以使傳輸整合步進機在具有小批量和光罩設置的情境下提升機台利用率。我們並執行大量實驗比較GA-F與其他兩種的派工法則：GA-FT (traditional family-based genetic algorithm) 與GA-I (individual-based genetic algorithm)，實驗結果顯示，GA-F均勝過這兩種派工方法。

關鍵詞：排程、半導體、基因演算法、家族式派工

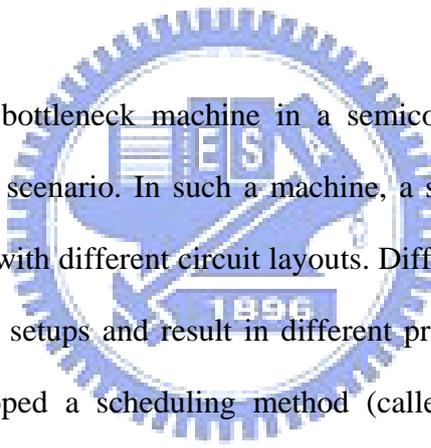
A family-based GA algorithm for scheduling in-line stepper in small-lot scenarios

Student : Chia-Wen Lu

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. In such a machine, a setup or mask change is required in processing jobs with different circuit layouts. Different job sequences may require different number of setups and result in different productivity for an in-line stepper. This paper developed a scheduling method (called family-based genetic algorithm, GA-F) in order to increase the machine utilization of in-line steppers, in a small-lot scenario which includes setup characteristics. Two other sequencing methods, called GA-FT (traditional family-based genetic algorithm) and GA-I (individual-based genetic algorithm), are taken as benchmarks and compared with the GA-F method. Numeric experiments indicate that the GA-F method outperforms these two benchmarks.

Keywords: scheduling, semiconductor, genetic algorithm, family-based

誌謝

本論文要感謝的是恩師巫木誠教授的悉心指導與教誨才得以順利完成。巫老師讓我可以同時兼顧家庭與學業，在老師的敦敦教誨下，使我在學術研究上獲益良多，並且教導我許多人生中做人處事的道理，得以終身受用。同時也感謝許錫美教授、彭德保教授和陳文智在論文口試時，所給予的寶貴意見與指導，讓本論文更臻完備。

在研究所的兩年中，要感謝同門的黃亮銓、林昭宏、林慈盈、陳文旻，陪我度過修課與撰寫論文的的日子，時常打氣與討論，使我在交大的生活變的愉快而充實。在此並特別感謝邱志文學長的幫助，指導我程式的撰寫，並時常解決在研究中遇到的困難，使我得以順利完成研究。

在此特別感謝我最愛的家人，在論文撰寫期間給我無數的關愛與體諒。特別是我的父母親呂文志先生與楊雅玲女士，感謝你們多年來的辛勞與關懷，對我總是無怨無悔的付出，對於你們的感謝，實非筆墨可以形容，還有我妹妹呂忻昀給我的加油與打氣，最後謹以此論獻給我最敬愛的家人、師長與朋友。

呂佳玟 于 新竹交大

2009'7'1

目 錄

中文摘要.....	i
Abstract.....	ii
誌 謝.....	iii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 問題特色.....	5
1.3 研究目的與方法.....	6
1.4 論文章節介紹.....	6
第二章 文獻探討.....	7
2.1 研究步進機派工的問題.....	7
2.3 流程式生產排程 (Flowshop scheduling).....	7
2.3 Family Setup Time 的分類.....	9
2.4 傳輸整合步進機的機台結構及製程.....	10
2.5 基因演算法 (Genetic algorithm).....	11
2.6 文獻探討結語.....	13
第三章 以基因演算法解傳輸整合步進機排序.....	14
3.1 研究問題描述.....	14
3.2 複雜度分析.....	14
3.3 研究方法.....	15
3.4 染色體的設計與解讀.....	15
3.4.1 染色體設計.....	15
3.4.2 染色體解讀.....	16
3.5 適合度函數定義.....	19
3.6 基因演算法求解.....	21
3.6.1 基因演算法的步驟.....	21
3.6.2 Crossover Operators 交配運算子.....	22

3.6.3 Mutation Operators 突變運算子	24
3.6.4 選擇策略.....	25
3.6.5 結束條件.....	25
3.7 本研究方法之特色.....	25
3.7.1 染色體設計與解讀方法.....	25
3.7.2 染色體交配與突變方式.....	27
第四章 實例驗證.....	29
4.1 測試情境的設計.....	29
4.2 標竿說明.....	30
4.3 基因演算法的參數設定.....	30
4.4 實驗結果與分析.....	31
4.4.1 比較不同情境下 GA-F 與 GA-FT 的的 Makespan.....	31
4.4.2 比較不同情境下 GA-F 與 GA-FT 的求解速度	36
4.4.3 比較不同情境下 GA-F 與 GA-I 的 Makespan	38
4.4.4 比較不同情境下 GA-F 與 GA-I 的求解速度.....	43
4.5 實驗結論.....	44
第五章 結論與未來研究方向.....	45
5.1 研究的結論.....	45
5.2 未來研究方向.....	46
參考文獻.....	47

表目錄

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)） ..	29
表 4.2 比較所有測試情境在不同 JOB 數下 R 的平均數.....	31
表 4.3 比較所有測試情境在不同設置時間下 R 的平均數.....	32
表 4.4 比較所有測試情境在不同光罩數量下 R 的平均數.....	33
表 4.5 比較所有測試情境在不同 YIELD 分佈下 R 的平均數.....	34
表 4.6 比較 GA-F 與 GA-FT 的求解時間差異（單位：秒）	37
表 4.7 比較所有測試情境在不同 JOB 數下 R 的平均數.....	38
表 4.8 比較所有測試情境在不同設置時間下 R 的平均數.....	39
表 4.9 比較所有測試情境在不同光罩數量下 R 的平均數和標準差.....	40
表 4.10 比較所有測試情境在比較不同 YIELD 分佈下 R 的平均數.....	42
表 4.11 比較 GA-I 和 GA-F 的求解時間差異（單位：秒）	43



圖目錄

圖 1.1 傳輸整合步進機的組態	2
圖 1.2 傳輸整合步進機 FULL-LOT 無產能損失範例	3
圖 2.1 埠區限制與流程式生產示意圖	8
圖 2.2 傳輸整合步進機的生產系統	10
圖 2.3 基因演算法的搜尋流程	11
圖 3.1 研究問題示意圖	14
圖 3.2 研究方法	15
圖 3.3 染色體的設計	16
圖 3.4 JOB FAMILY SEQUENCING 的解讀	17
圖 3.5 光罩 2 內工件的排序方法	18
圖 3.6 光罩 3 內工件的排序方法	18
圖 3.7 光罩 1 內工件的排序方法	18
圖 3.8 染色體解讀前與解讀後示意圖	18
圖 3.9 適應度函數求取較佳染色體流程	20
圖 3.10 基因演算法的步驟	21
圖 3.11 CROSSOVER C1 運算	22
圖 3.12 CROSSOVER LOX 運算	23
圖 3.13 CROSSOVER PMX 運算	24
圖 3.14 MUTATION 運算 A. SWAP B. INVERSE	24
圖 3.15 過去文獻的染色體設計	26
圖 3.16 GA-FT 的 C1 單點交配方式	27
圖 4.1 比較所有測試情境在不同 JOB 數下 R 的改變	32
圖 4.2 比較所有測試情境在不同設置時間下 R 的改變	33

圖 4.3 比較所有測試情境在不同光罩數量下 R 的改變.....	34
圖 4.4 比較所有測試情境在不同 YIELD 分佈下 R 的改變.....	35
圖 4.5 GA-FT1 染色體表達方式.....	36
圖 4.6 在各種 JOB 數目的情境下 GA-F 與 GA-FT 的求解時間.....	37
圖 4.7 比較所有測試情境在不同 JOB 數下 R 的改變.....	39
圖 4.8 比較所有測試情境在不同設置時間下 R 的改變.....	40
圖 4.9 比較所有測試情境在不同光罩數量下 R 的改變.....	41
圖 4.10 比較所有測試情境在不同 YIELD 分佈下 R 的改變.....	42
圖 4.11 比較 GA-F 和 GA-F 的求解時間差異.....	44



第一章 緒論

1.1 研究背景

近幾年來，隨著半導體產業技術不斷的進步，必須興建新廠房以應付市場與技術的需求。投資興建新工廠時會面臨一個重要的決策，就是機台的購買數量，因為在半導體晶圓廠中，大約有 80% 的投資金額都投入在機台的購買上，故必須要讓機台發揮最大的績效（也就是最大的產出），才可以減少購置多餘的機台。然而在生產現場中，瓶頸機台才是決定產出的關鍵，所以要充分利用瓶頸機台的產能，才能真正增進工廠的產量。在機台投資中，單價最高的機台為傳輸整合步進機（In-line Stepper），故在機台的購買與生產策略的配置上，會將瓶頸機台設定為傳輸整合步進機，所以只要能夠充分利用傳輸整合步進機，就可以減少機台的閒置，增加工廠的產能，減少不必要的投資。

圖 1.1 為傳輸整合步進機的組態，該機台是由在製品暫存區（Buffer）、埠區（Port）和反應室（Chamber）所組成的流程式生產（Flow Shop），其內部大約有 20 多個反應室所構成，生產方式會依照製程順序一個步驟緊接著下一個步驟，所以可視為流程式生產的一種。機台設計最多會有 4 個埠區，每一個埠區一次是放置一個晶圓批（Lot），故在埠區內的單位為晶圓批。而因為反應室一次只能加工一片晶圓（Wafer），故在反應室內的單位為晶圓。

傳輸整合步進機的設置時間發生在曝光製程的反應室中，由於晶圓需要在該機台加工不同的電路佈局，故晶圓會不斷反覆的進入傳輸整合步進機中加工，當加工不同加工層時（亦即不同的電路），則需要更換光罩，大約是在 1.5 至 6 分鐘。

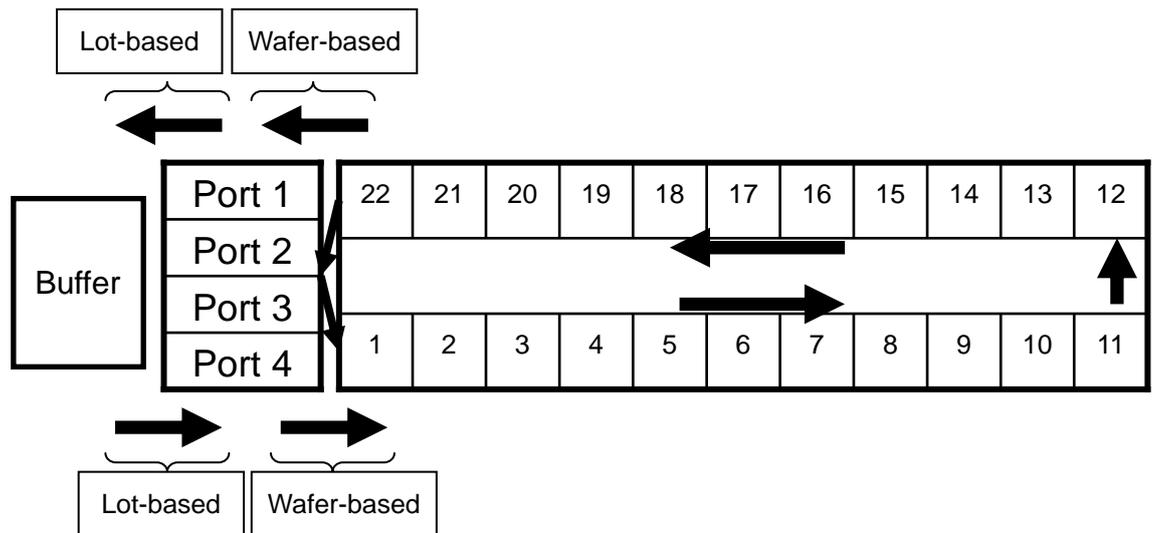


圖 1.1 傳輸整合步進機的組態

傳輸整合步進機的傳輸單位有兩種：Lot 或 Wafer，如圖 1.1 所示，從在製品暫存區到埠區的傳輸單位為 Lot；埠區到反應室為 Wafer；反應室到埠區為 Wafer；埠區到在製品暫存區為 Lot，形成傳輸單位不一致的問題。正因為機台具有傳輸單位不一致的問題，故在機台加工的工件必須要整批都加工完才可離開埠區，讓下一個晶圓批進入埠區開始加工。

一般傳輸整合步進機皆是以滿批量（Full-lot）為生產單位去設計，滿批量代表在晶圓批內的晶圓數量為 25 片晶圓，而過去的文獻也都是在滿批量的情境下做研究。

在滿批量的情況下，傳輸整合步進機不會有產能閒置的問題。如圖 1.2 所示，若傳輸整合步進機有 22 個反應室，今有 5 個晶圓批要加工，分別為 Lot 1、2、3、4 及 5，每個晶圓批中的晶圓數量皆為 25 片（因為滿批量為 25 片），圖 1.2 的情境為當 Lot 1 加工完第一片後，機台發生產能閒置的情形，由圖示可看出此時傳輸整合步進機內有 Lot 1、2、3 及 4 共 99 個晶片（佔 22 個反應室），並且沒有任何一個反應室是閒置。故在滿批量的情況下，不會有產能損失的問題，所以以往文獻也都將此情形看成單一機台來做研究。

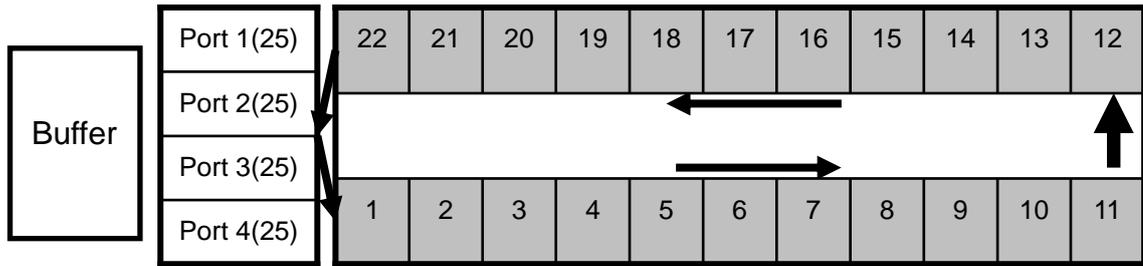


圖 1.2 傳輸整合步進機 Full-lot 無產能損失範例

但是在晶圓廠中並不是只有滿批量的情況，有時亦會發生有小批量 (Small-lot) 的情況，小批量代表在晶圓批內的晶圓數量不足 25 片晶圓，而過去的文獻亦很少在小批量的情境下做研究。小批量發生的原因有二：第一，低良率情況，產品在試產階段時，投料時一批是 25 片，加工時因製程或設備出現問題，造成廢品，導致工件在進入傳輸整合步進機前就形成小批量的情形。第二，為工程測試批情況，為了進行新產品或新製程驗證，在投料時即刻意安排較少的批量（譬如一批只放置 7 片）。如果一間晶圓工廠綜合了低良率加上新製程導入、新產品開發的情況，則發生小批量的情形可能會非常多。

在小批量的情況下，傳輸整合步進機很容易造成產能閒置。例如，若傳輸整合步進機有 22 個反應室，今有 5 個晶圓批要加工，分別為命名為 Lot 1、2、3、4 及 5，每個晶圓批中的晶圓數量為 3、5、7、2 及 12 片，圖 1.3 的情境為當 Lot 1 加工完第一片後機台發生產能閒置的情形，由圖示可看出此時傳輸整合步進機內有 Lot 1、2、3 及 4 共 16 個晶片(佔 16 個反應室)，但機台有 6 個 (1 至 6) 反應室閒置，這是因為 Lot 5 因為有埠區的限制，所以沒有多餘的埠區可以使用，故無法進入機台中加工，所以可以說明傳輸整合步進機在小批量的情境下可能會發生產能閒置的情形。

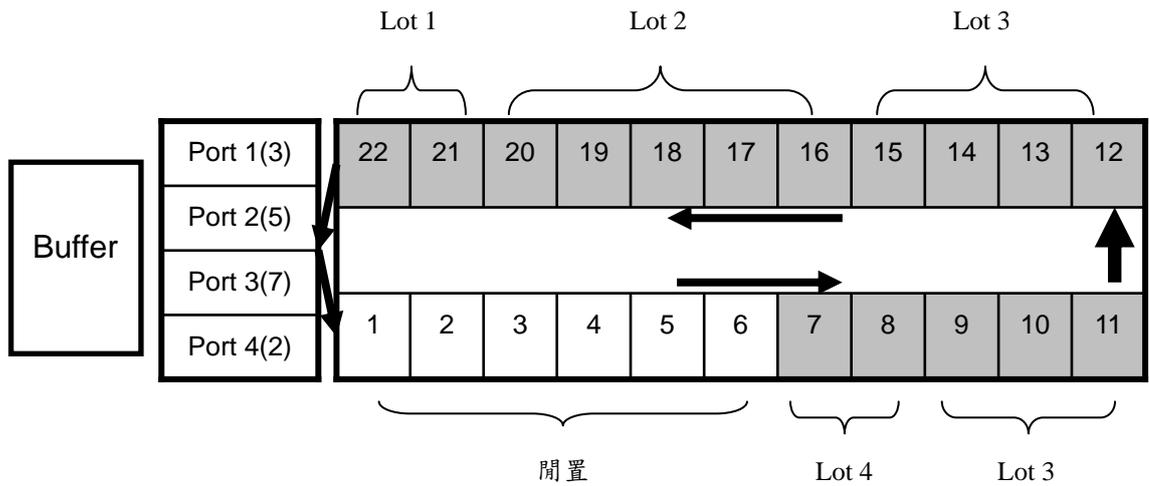


圖 1.3 傳輸整合步進機因小批量造成產能閒置的例子

要避免機台閒置，其最重要的就是要有好的工件排序，如圖 1.3 的例子為例，若工件順序為 1→2→3→4→5，則會有 6 個反應室閒置；但如果工件排序為 1→5→3→4→2 則沒有反應室會閒置，如圖 1.4 所示。因此，在小批量的情境下，如何在傳輸整合步進機上有效安排加工順序，避免機台閒置，為本研究所著手的方向。

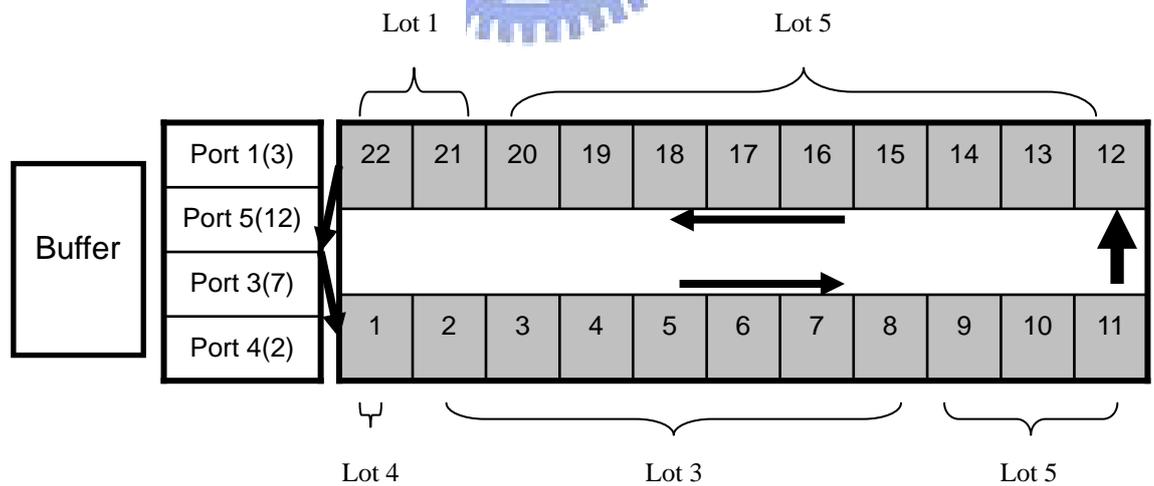


圖 1.4 工件排序 1→5→3→4→2 無產能損失範例

過去 Wu & Chiou (2009) 曾經針對小批量的情境作傳輸整合步進機的排程，以增加工廠的產能，但並無考慮到有些晶圓批會有使用相同光罩的情形，所以將情境假設為每一個晶圓批都不同，加工每一個晶圓批都需要更換光罩，即每一個晶圓批都需要設置時間。然而在現實情況中，半導體工廠是一座具有迴流製程特性的工廠，故工件會重複進入傳輸整合步進機中加工，然而每次皆為加工不同的電路佈局，在加工不同電路佈局時才需要更換光罩；換句話說，相同的產品在同一加工層時（加工相同的電路佈局），會使用相同光罩，故本研究將其條件考慮在內，使其更符合真實情形。

過去 Chern & Liu (2003) 為研究在 DRAM 的生產情境及換光罩需要設置時間（6 分鐘）的情境下，使用 Family-based SDA 的派工法則使產出最大化。其文中提及在換光罩需要設置時間時，使用家族式派工法則（Family-based）比使用單獨派工法則（Individual-based）的績效更好，使產出與機台利用率更大。

家族式派工的方法為將使用相同光罩之晶圓批排序一起，如用此排序方法，可以減少在加工時重覆更換光罩所產生多餘的設置時間，使其總完工時間越小，產出與機台利用率越大。由於本論文也有光罩設置問題，故在研究方法中亦會納入家族式派工的觀念，減少更換光罩的次數，其演算方法將會在第三章中詳細說明。

1.2 問題特色

在過去文獻的研究中，多數是在量產階段（滿批量）與高良率的假設下做派工，但鮮少在小批量與低良率的情境下做研究探討。在滿批量情境下，傳輸整合步進機的傳輸單位不一致與埠區限制的情況並不會造成機台發生產能閒置的情況，故以往文獻皆將此問題視為一部機台的問題來探討。但本研究則是考量在小批量情境中，傳輸整合步進機的傳輸單位不一致與埠區限制情況下，所造成的產能閒置問題，並且將問題納入了光罩設置的情境，使其與真實情境更符合，越能表現出傳輸整合步進機在工廠裡的運作，讓研究更有價值。

1.3 研究目的與方法

本研究之目的為利用模擬傳輸整合步進機的真實情況，藉以求出最佳化的排程組合，以使機台利用率及產能最大化。

由於在面對生產晶圓批件超過 20 批的排程問題時，求解時間會很長，所以一般線性規劃或整數規劃方法求解無法解此問題。因此本研究利用基因演算法（Genetic Algorithm, GA）與家族式派工方法求解，希望可以藉由基因演算法快速求出近似最佳解。本論文亦有比較各種排程方法的績效，求得一個最佳的工件排序組合。

1.4 論文章節介紹

本論文內容共含五章。第一章敘述研究主題、研究動機及研究目的等。第二章探討有關傳輸整合步進機的機台結構及製程，以及步進機和流程式生產製程目前研究的議題，最後針對本論文會使用到的演算法之相關研究進行探討。第三章描述模式介紹以及相關的參數設定，以基因演算法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

本章首先簡介探討步進機和流程式生產製程目前研究的議題、家族式設置時間的分類，接著會介紹傳輸整合步進機的機台結構及製程，最後則針對本論文會使用到的演算法之相關研究進行探討。

2.1 研究步進機派工的問題

傳輸整合步進機的構造主要是由埠區加上反應室所組合而成，以往的文獻中，總是在滿批量的情境下派工，所以並不會因為傳輸不一致與埠區限制而導致產能損失的問題，故在以往的文獻中，在滿批量情境下，埠區再加上反應室是看成一部單一機台，只需做單機排序，其次再針對不同的問題解決。

在步進機的派工問題中，大致上可以將問題分成兩大類：第一，單機排序，研究的問題為針對一部步進機做工件排序問題；第二，由步進機加上其他非瓶頸機台所組合而成的生產系統，再對整個生產系統做最佳之生產排序。近期的文獻大多是再做整個生產系統的派工，目標可能有 Throughput、Cycle Time、Hit Rate...等 (Ying & Lin, 2009; Wu et al., 2008; Wu et al., 2008; Wu et al., 2006; Chern & Liu, 2003)。

2.3 流程式生產排程 (Flowshop scheduling)

在第一章的研究背景曾經說明過傳輸整合步進機的內部是由一連串的反应室所組合而成的流程式生產，所以本章節將會探討流程式生產排程問題的相關研究，過去已有許多文章針對此問題整合與分類 (Cheng et al., 2000; Pinedo, 2002; Hejazi et al., 2005; Bagchi et al., 2006; Zhu et al., 2006; Quadt et al., 2007)，故依照文獻分類方式，可以將流程式生產排程問題區分成工作站類別、製程限制，與目標函數三種。而與本研究相關的流程式生產排程問題主要為製程限制，以往文獻將製程限制區分成四種，第一種為 No-wait，表示工件在生產過程中，不允許在機台中或在機台間中斷；第二種為 Blocking，表示在機台間不能有暫存區 (Buffer)，在製品亦不能留在機台中；第三種

為 Limited Buffer，表示在機台間的暫存區大小有限制；第四種為 Unlimited Buffer，表示在機台間的暫存區大小沒有限制。本研究可視為 Limited Buffer 製程限制的一種變形，但在過去文獻中並無探討，顯示出本論文重要之處。

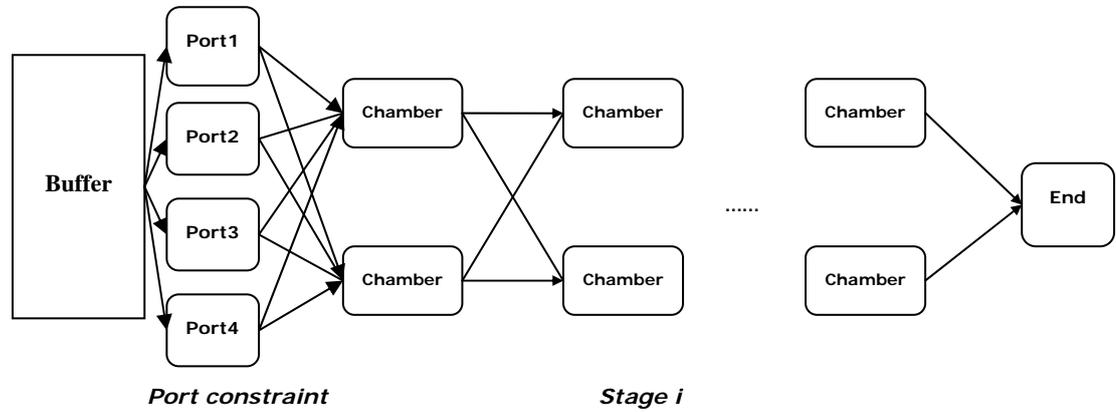


圖 2.1 埠區限制與流程式生產示意圖

進幾年研究流程式生產排程問題中 (Liao et al., 2006; Ying & Lin, 2006; Ruiz & Maroto, 2006; Ruiz & Stutzle, 2007; Lin et al., 2008)，亦無探討埠區限制的問題。其實埠區的限制可視為在流程式生產的工作站前增加入口的限制，如圖 2.1 所示，使其限制加工工件的數量，讓本研究問題形成具有產能損失的流程式生產排程問題，故本文後續將會針對此問題進行研究與探討。

2.3 Family Setup Time 的分類

產品族 (Job Family) 的定義是具有相同的設置，以本研究來說，相同的產品族會使用相同的光罩，故符合家族設置時間 (Family Setup Time) 的問題。

Potts & Kovalyov (2000) 將研究設置時間的問題區分為許多不同的種類，其中又特別將家族設置時間問題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)。

舉例來說，假設現在有兩個產品族，分別為家族一和家族二，在機台上先做家族一的設置時間為 S_{01} ，先做家族二的設置時間為 S_{02} ，做完家族一再做家族二的設置時間為 S_{12} ；而先做家族二再做家族一的設置時間為 S_{21} 。順序獨立家族設置時間研究的問題是 $S_{01} = S_{21}$ 與 $S_{02} = S_{12}$ ，也就是說不論做家族一還是家族二的設置時間都是一樣的；順序相依家族設置時間研究的問題是 $S_{01} \neq S_{21}$ 與 $S_{02} \neq S_{12}$ ，意思是先做家族一還是家族二其產生的設置時間是有差異的。

本研究的設置時間是發生在換光罩的時候，但是每換一次光罩所需要的設置時間皆相同，並不會因為所使用的光罩不同而使設置時間產生變化，故本研究將被分類為順序獨立家族設置時間的問題。

2.4 傳輸整合步進機的機台結構及製程

一個典型的傳輸整合步進機系統，主要由在製品暫存區 (WIP Buffers)、埠區 (Port)、以及傳輸整合步進機三大部分所構成，如圖 2.2 所示 (Quirk, 2001; Xiao, 2001)，而中間的傳輸主要透過 Robot 做傳輸。

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。自動化光阻塗佈及顯影系統包括六種站點 (Stage)，分別為氣相塗底 (Vapor prime)、光阻塗佈 (Photo-resist Coating)、軟烤(Soft-bake)、曝光後烘烤(Post-exposure bake)、顯影 (Develop)，以及硬烤(Hard-bake)，每一種站點有一個以上的反應室可以加工。而步進機包括有三種站點，分別為預對準 (Pre-alignment)、曝光 (Rxposure)，以及暫存區 (Buffer)，每一種站點只有一個反應室可以加工，而更換光罩的過程也在步進機中發生。

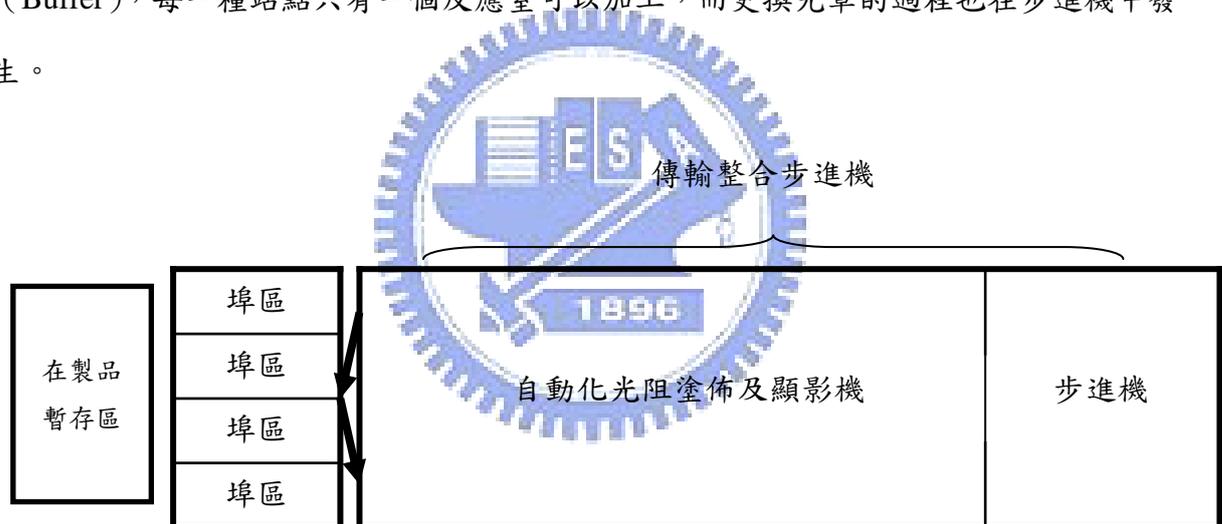


圖 2.2 傳輸整合步進機的生產系統

2.5 基因演算法 (Genetic algorithm)

基因演算法是由 Holland (1975) 所提出，其演算流程如圖 2.2 所示，首先隨機產生一初始的母代 (Population)，再針對母代中每一個染色體作適應函數值 (Fitness function) 的計算與評估。

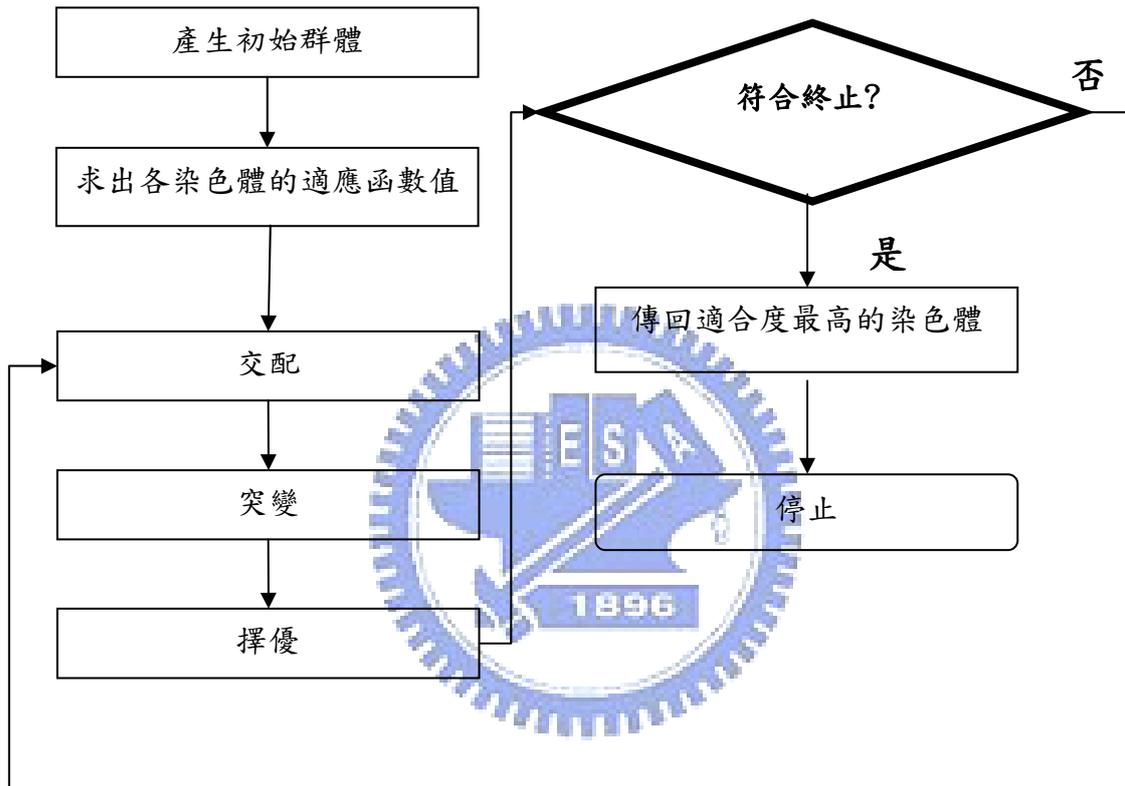


圖 2.3 基因演算法的搜尋流程

計算完適應函數值後，再評估此解是否滿足停止條件，如果是，則停止基因演算法，而此解即為最後所需要的近似最佳解；如果不符合停止條件，則會透過染色體交配 (Crossover) 與突變 (Mutation) 而成為新的下一代 (稱為子代)，混合上一代與新一代的染色體形成一個池 (pool)，在池中的染色體，適應函數值為較高者，就擁有較高

的機率被選取 (Selection) 保存到下一代當新的母代；在如此一代接著一代的進化過程，即可以保存每一代所產生的最好的染色體。整個基因演算法的搜尋過程如圖 2.2 所表示。

在使用基因演算法求解問題前，必須設計一條符合問題條件的染色體，而其中的每個字元稱為基因。染色體可以同時存在多個，而這些染色體所形成的群體，便稱為族群。

在族群中，染色體會根據不同的問題而有不同的好壞，經由計算出適應函數值，來評估此染色體的優劣。適應函數值較佳的，則代表此染色體對於問題的適應能力較佳，而被保留到下一代的機會也比較大；反過來說，適應函數值較差的染色體，則代表此染色體對於問題的適應能力較差，被保留到下一代的機會也變小。在計算完每一條染色體的適應函數值後，再選取適應函數值最高的染色體值當做下一代的母代。

交配的過程為隨機選取兩條染色體，再藉由交換內部之基因，找出適應能力較佳之下一代，經由此方法來找出更好的染色體。而交配通常會配合突變一起使用，即可避免在基因演算法求解過程中，可行解落入區域最佳解 (local optimum)。

染色體完成交配與突變後，池中會有母代的染色體加上子代的染色體，池中的染色體會依據各個染色體的適應函數值來選出下一代。選擇的方式則使用輪盤法 (Roulette wheel 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，適應函數值越好則所分配到的輪盤面積越大，當隨機選取一點時，佔輪盤面積越大的染色體則代表被選擇的機會越大，成為下一代母代的機率也越高。

2.6 文獻探討結語

過去文獻做的研究，均假設為滿批量，只把傳輸整合步進機看為生產系統中的一部瓶頸機台，而本研究為在小批量情境下對傳輸整合步進機做派工。

有關傳輸整合步進機中埠區的限制與傳輸單位不一致現象，所造成的機台產能損失，目前只有 Wu & Chiou (2009) 探討研究，而本研究增加了光罩限制，使其更符合真實情況。

本研究在考量在光罩與小批量情境下，進行傳輸整合步進機在小批量情境下之排程規劃。回顧過去相關之文獻，並無相同之研究，如此更可顯示出本研究之獨特性與重要性。分析先前介紹之文獻，本研究之特色在於，在考量光罩設置和小批量情境下對於傳輸整合步進機排程的影響。確定本論文之獨特性與重要性後，將於第三章說明詳細的求解過程。



第三章 以基因演算法解傳輸整合步進機排序

在本章探討如何利用基因演算法，來求解傳輸整合步進機的排程問題，本章分成染色體解碼、建構適應性函數和使用基因演算法的架構及相關參數。

3.1 研究問題描述

本研究問題是針對具有流程式生產問題特性的傳輸整合步進機做排程規劃，將埠區限制加入流程式生產問題中，加上有小批量與傳輸單位不一致的情境下，如圖 3.1 所示，欲求出最佳之工件排序組合，績效指標為最小之總完工時間。

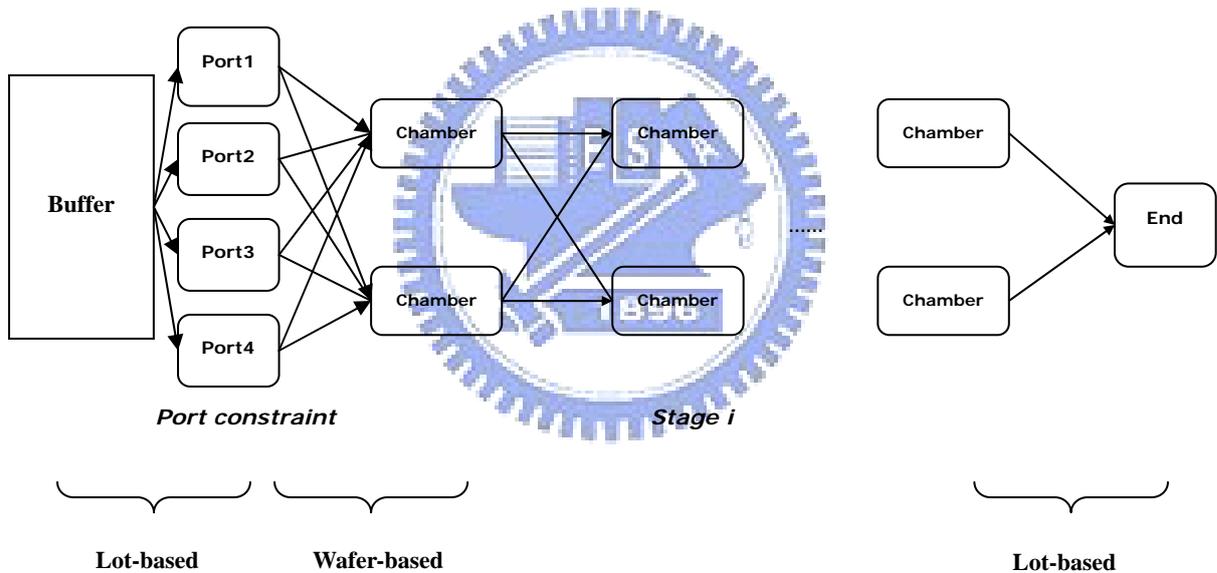


圖 3.1 研究問題示意圖

3.2 複雜度分析

本研究之目的為求出最佳之工件排序組合， n 個晶圓批就有 $n!$ 個排序組合，假設現有 20 個晶圓批，所有解的排序會有 $20!$ 種組合，解空間非常龐大，故求解會非常複雜且時間過長。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，所以本研究擬用基因演算法來求解此問題，可在較短的時間下求得一個績效良好的解。

3.3 研究方法

此節將說明本研究應用基因演算法來求解傳輸整合步進機排程問題，而本研究基因演算法的規劃步驟與演算流程如圖 3.2 所示，首先會先介紹染色體的解碼和適應性函數的定義，接著介紹基因演算法的流程：

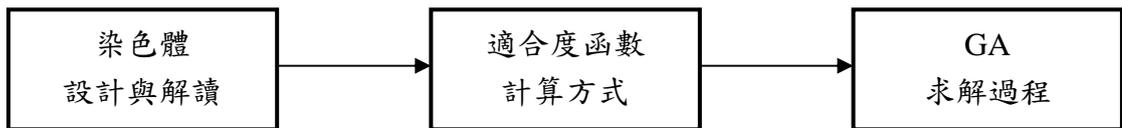


圖 3.2 研究方法

3.4 染色體的設計與解讀

本章節將會介紹在小批量與光罩設置的情境下，所設計的染色體。其染色體會符合此問題的情境，將光罩設置的條件納入設計的理念中，以下將說明染色體設計的概念與解讀的方法。

3.4.1 染色體設計

本研究利用基因演算法求解此傳輸整合步進機排程問題。使用一條染色體 (chromosome) 代表一個工件排序，為一個基因(gene)。每一個工件皆會有一個屬性，為光罩類別，如圖3.3所示，本論文將此染色體設計稱為GA-F (Family-based GA)，意即結合家族式派工與基因演算法的概念。在本研究中，相同產品在相同的加工層加工時，會使用相同的光罩，即具有相同的 Job Family。以下範例將說明染色體設計的概念，假設現在有10個Lots，3種光罩，圖3.3為此情境之染色體設計範例：

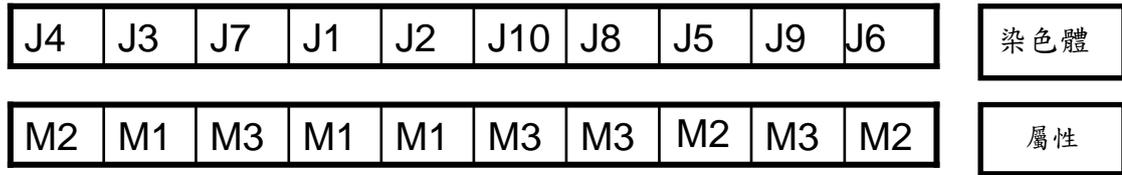


圖3.3 染色體的設計

在染色體中，基因的名稱代表工件名稱，例如染色體第一個基因為J4，則代表Job 4；工件順序則代表加工順序，在圖3.3的染色體，加工順序依序為Job 4、Job 3、Job 7、Job 1、Job 2、Job 10、Job 8、Job 5、Job 9，最後做Job 6；而染色體長度則代表工件數目，圖3.3的染色體有10格基因，代表有10個工件欲加工。

而在工件的屬性中，則代表每一個工件所使用的光罩類別，但此屬性並不為染色體的一部分，為工件所附屬之屬性。屬性之說明利用圖3.3的染色體和屬性來說明，Job 4所使用的為M2(光罩2)，Job 3所使用的為M1(光罩1)，而Job 7所使用的為M3(光罩3)，其他以此類推。



3.4.2 染色體解讀

在本研究中所使用的績效評估指標為總完工時間，而總完工時間是由總加工時間、等候時間和設置時間所組合而成的，所以減少其中任何一個因素，則可降低總完工時間。在過去文獻中，Chern & Liu (2003) 曾利用Family-based的觀念來減少具光罩設制的問題，Family-based的想法為：將使用相同光罩的工件排序在一起，如此一來即可減少更換光罩的次數，而因為更換光罩所造成多餘的設置時間，也可隨著更換光罩之頻率降低而減少。因為設置時間越少，而總完工時間也會越短，故在此文獻中Family-based的方法對於具光罩限制下的問題有很良好的成效，因此本研究在染色體解讀的概念，會將Family-based的概念納入染色體的設計解讀中，如此設計解讀方法可以減少設置時間，使總完工時間變短。

在染色體解讀中，會分成兩部份來說明：首先會先決定 Job Family的順序（Among Job Family Sequencing），也就是光罩的順序；第二，是決定 Job Family內各工件的順序（Within Job Family Sequencing）。

決定 Job Family 是根據光罩的順序來決定的，依圖3.4來說，在Job的光罩別可以看出，光罩排列的順序依次為2、3、1，所以 Job Family 的順序依序為 2、3、1，代表必須要優先加工使用光罩2的所有工件後才能加工屬於光罩3的工件，最後才能加工使用光罩1的工件。

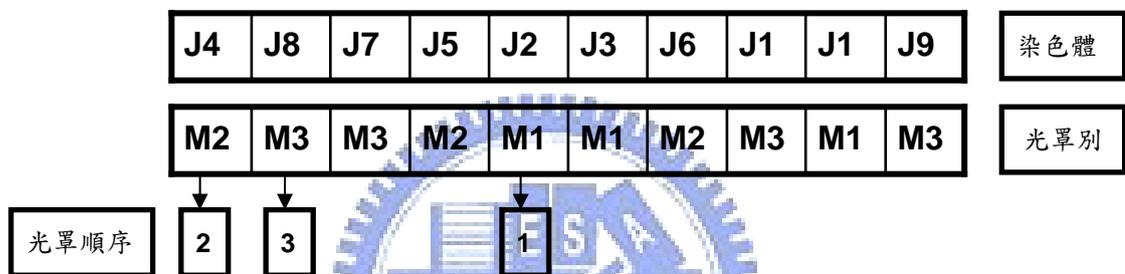


圖 3.4 Job Family Sequencing 的解讀

確定光罩的加工順序後，接著就要排序出在每個光罩內各工件的加工順序。決定 Job Family內Job的序是根據基因的順序來決定的，如圖3.5所示，在Job Family M2內的基因順序為4、5、6，代表在光罩2內必須將加工順序排序為Job 4、 Job 5 、Job 6；如圖3.6所示，Job Family M3內的基因順序為8、7、10、9，代表在光罩3內必須將加工順序排序為Job 8、 Job 7 、Job 10、 Job 9；如圖3.7所示，Job Family M1內的基因順序為2、3、1，代表在光罩1內必須將加工順序排序為Job 2、 Job 3 、Job 1。排序完各光罩內的工件順序後，即可將光罩順序與工件順序結合，得到一組新的染色體，解讀後的染色體的順序依序為為4、5、6、8、7、10、9、2、3、1。

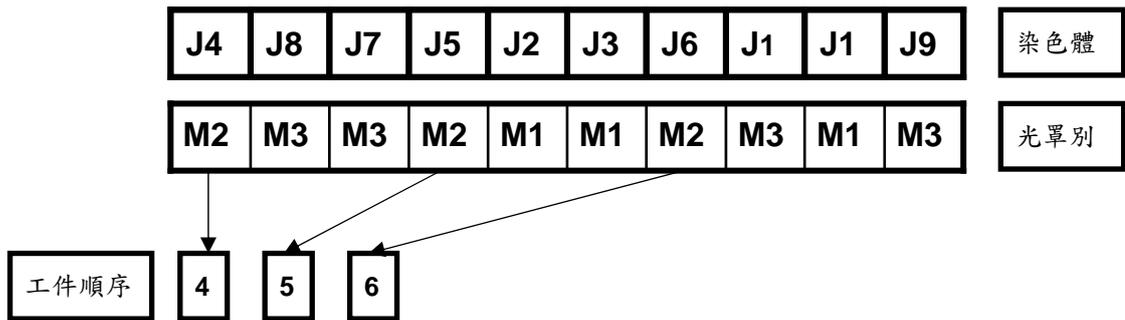


圖 3.5 光罩 2 內工件的排序方法

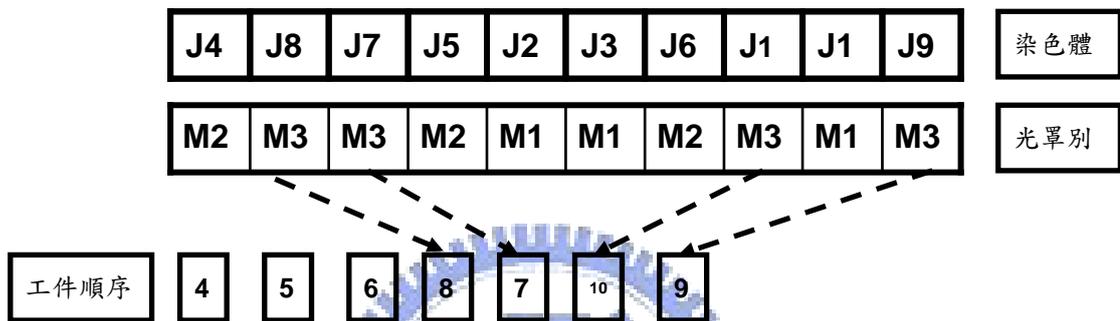


圖 3.6 光罩 3 內工件的排序方法

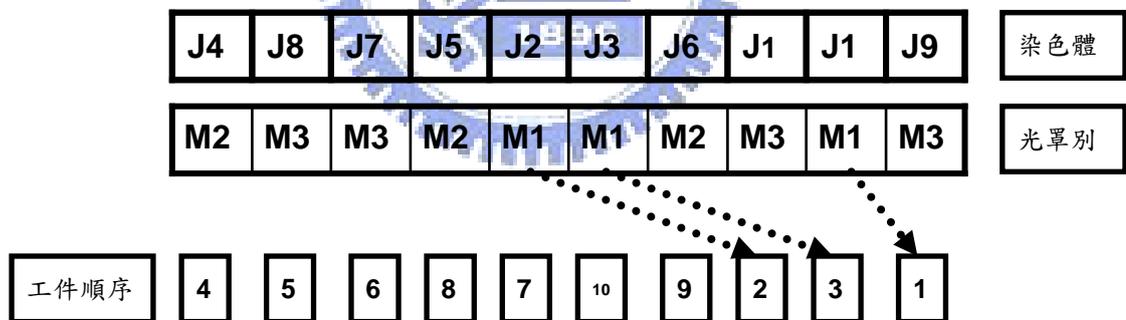


圖 3.7 光罩 1 內工件的排序方法

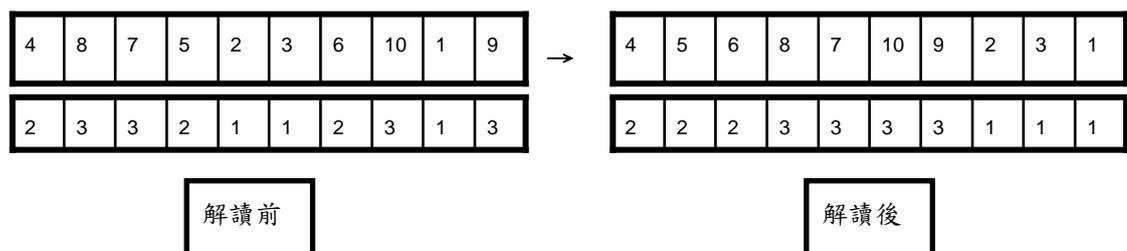


圖 3.8 染色體解讀前與解讀後示意圖

由圖3.8可以清楚的顯示，此一染色體解讀前後明顯的改變了順序，在經過解讀後工件已經依照光罩的類別排序，將設置時間降到最低。

3.5 適合度函數定義

定義適應度函數是為了決定每一個染色體適應環境的能力，也就是染色體是否能夠生存的一個依據，適應度越高的染色體就越能夠存活下來，反之則淘汰。由於本研究是探討傳輸整合步進機排程問題，目標是產出最大的排程問題；戰術作法：降低工件的總完工時間，使產出最大，以維護排程的最佳效益。目標式為加工排序 π 的最小總完工時間，因此將定義適應度函數定義如下：

$$\text{目標式} \Rightarrow \text{Min}\{C_{\max(\pi)}\}$$

$$C_{\max(\pi)} = \text{加工排序 } \pi \text{ 的總完工時間}$$



圖 3.9 為適應度函數求取較佳染色體流程，首先先將染色體解讀後決定工序，接著進入用 C++所寫的模擬程式中，算出染色體的總完工時間，而總完工時間越小，則此染色體的適應性越好。



圖 3.9 適應度函數求取較佳染色體流程

3.6 基因演算法求解

本研究利用基因演算法求解此傳輸整合步進機排程問題。基因演算法可以在龐大的解題空間 $n!$ 工件排序中，找尋近似解，圖3.10為基因演算法求解過程。

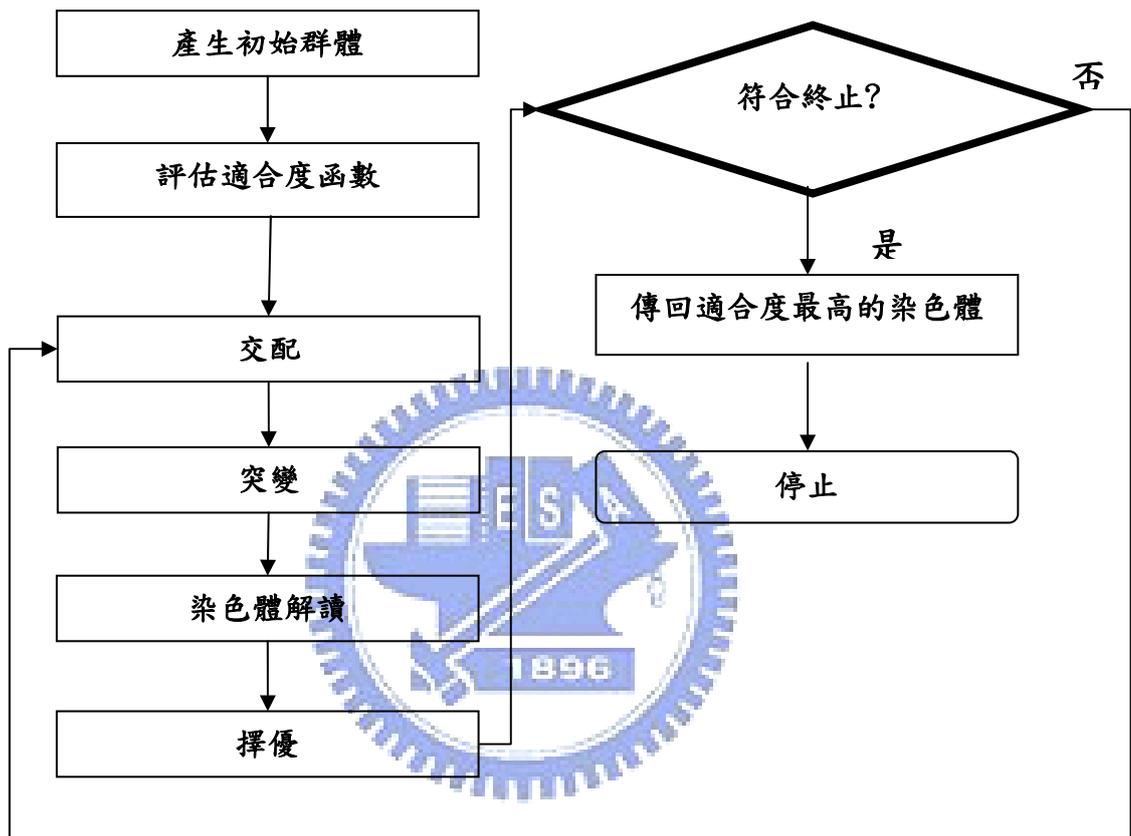


圖 3.10 基因演算法的步驟

3.6.1 基因演算法的步驟

Step 1：隨機產生初始群體：以隨機的方式選擇染色體（chromosomes），形成一個啟始的母體（population）。

Step 2：評估每一個染色體的適合度函數，並視其是否符合終止條件，如果符合，則停止尋找；反之，則繼續尋找最佳解。

Step 3：使用交配運算法則（crossover operators）產生兩個新染色體。

Step 4：使用突變運算法則（mutation operators）產生一個新染色體。

Step 5：將原始母代和運用交配運算子所產生的子代聯合，當作一個新的交配池，經由染色體解讀後再使用選擇策略（selection strategy）去篩選染色體。

Step 6：若產生的解符合結束條件則停止尋找最佳解，而此解亦為最終之近似最佳解。

3.6.2 Crossover Operators 交配運算子

在本研究所使用的基因演算法中，使用三種交配運法則，進行染色體的交配，分別為 C1 (單點交配) (Reeves, 1995)、線性順序交配(LOX) (Croce et al., 1995)，以及部分相應交配(PMX) (Goldberg, 1989)。

每一次的交配，會選擇兩個染色體，稱為母代 (parents)。經過交配後，會產生兩個染色體，稱為子代(children)。以下描述各個交配運算方法，本研究將母代以 parent-1 及 parent-2 來命名，子代以 child-1 及 child-2 來命名。

單點交配(C1 Operator)：如圖 3.11 所示，隨機選擇二個母代的染色體，每一個母代染色體中的隨機選取一個點，將染色體分成二部分 (head-section and tail-section)。為了產生子代，如 child-2，其首部 (head-section) 是從 parent-2 複製一個字串 (3, 5)；其尾部 (tail-section) 是由 parent-1 所得到，只要不是與既有的首部基因重複，將 parent-1 的基因依序放入 child-2。結果可得新的 child-2 其尾部為 (1, 2, 4)。child-1 亦可使用相同方式得到。

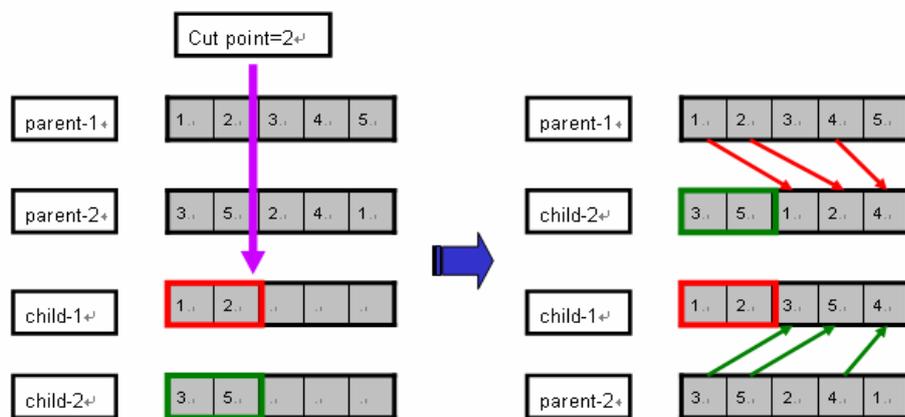


圖 3.11 Crossover C1 運算

線性順序交配(LOX: linear Order Crossover Operator)：如圖3.12所示，選二個母代的染色體，每一個母代染色體中的隨機選取二個點，將染色體分成三部分。母代 parent-1 及parent-2 之中間部分為 (2,3) 及 (4,5)，藉由複製 parent-1 的基因，將重複在parent-2 中間部分的基因取代成“H”基因，此時產生x-child-1 為(1, 2, 3, H, H)。此時將“H”基因往染色體的中間部分集中，而得 y-child-1 為 (1, H, H, 2, 3)。最後將y-child-1中間部分以 parent-2 中間部分取代，可得子代 child-1 為(1, 4, 5, 2, 3)。另一個子代child-2 可用相同方式產生。

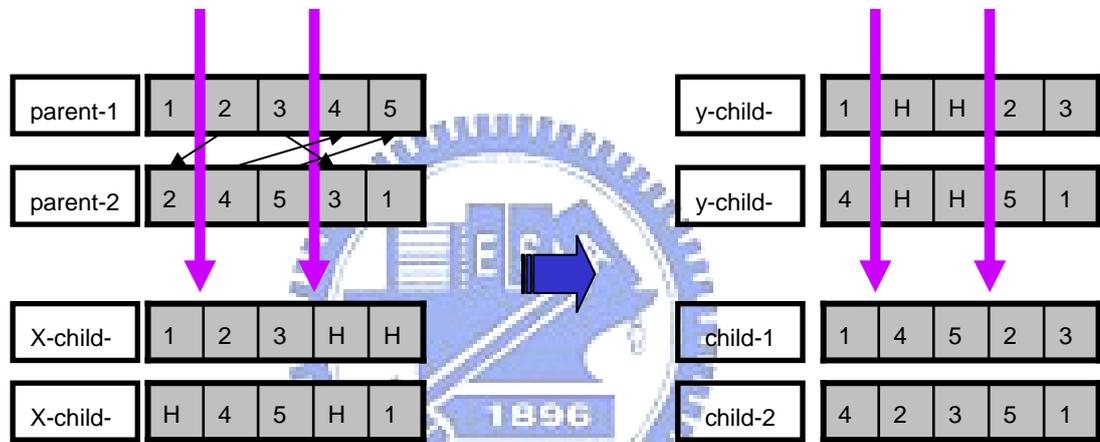


圖 3.12 Crossover LOX 運算

部分相應交配(PMX: Partially matched crossover Operator)：如圖 3.13 所示，選二個母代的染色體，每一個母代染色體中的隨機選取二個點，將染色體分成三部分。子代 x-child-1 的中間部分複製母代 parent-2 的中間部分，即 (4, 5, 3)，子代 child-1 的前部及後部複製母代 parent-1 的前部及後部。若基因在母代 parent-1 的前部及後部沒有在 child-1 出現，則複製它們到 child-1 的相同位置。最後對 child-1 空缺的基因，以 parent-1 未被分配的基因值取代。此時產生 child-1 為(1, 4, 5, 3, 2)。child-2 使用相同方法亦可得到。

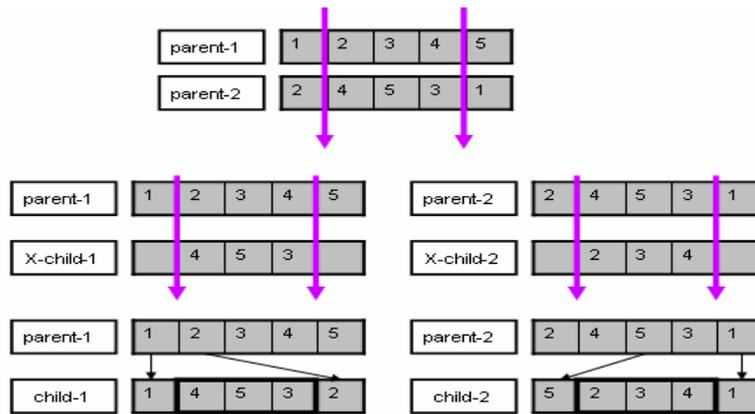


圖 3.13 Crossover PMX 運算

3.6.3 Mutation Operators 突變運算子

染色體突變的運算是為了避免在基因演算法的搜尋過程中，過早陷入局部最佳化的方法。本研究所使用的突變方式有兩種，分別為 Swap (交換)及 Inverse (反轉) (Wang & Uzsoy, 2002)。

交換：如圖 3.14.a 所示，在母代染色體中隨機選取兩個不同的基因，將其對調產生新的子代染色體。

反轉：如圖 3.14.b 所示，在母代染色體中隨機選取兩個切點，將染色體分成三部分，並將中間部分反轉，產生新的子代染色體。

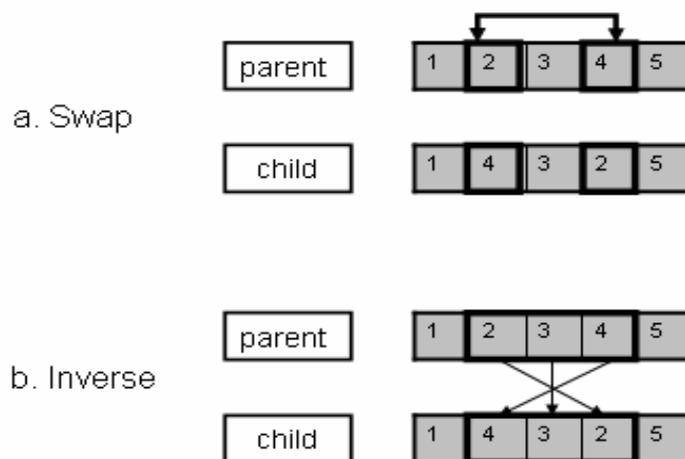


圖 3.14 Mutation 運算 a. SWAP b. Inverse

3.6.4 選擇策略

在完成基因運算後，會先經過染色體解讀的動作將其解讀成 Family-based，完成此程序後再使用輪盤法的選擇策略，挑選出適合度較佳的下一代。首先求出全部染色體的適應函數值大小，再由小到大排序。適應函數值最佳的必需要保留到下一代，其餘的染色體則依序輪盤法的方式被選取。

使用輪盤法的目的是讓每一條染色體都有機會被選擇，但適合度越佳(總完工時間越小)越容易被選取。它的選擇原理是依照染色體適合度之機率分割輪盤，其機率總合為 1，適合度越佳，則其機率越大，在輪盤上所佔的面積也越大，被選取的機會也就更大。

3.6.5 結束條件

若產生的解符合結束條件則停止尋找最佳解，而此解亦為最終之近似最佳解。本研究使用兩種結束條件，第一種條件為相同的最佳解一直保持 N 代，則此解為最終之最佳解；另一種條件為設定一個 Y 代，到達 Y 代即結束。



3.7 本研究方法之特色

此節將介紹本研究與過去文獻所使用的染色體之比較，在兩種染色體皆為結合基因演算法與家族式派工的設計概念下，試圖比較此兩染色體之優劣。以下將會介紹過去文獻的染色體設計解讀方法，以及交配與突變方式。

3.7.1 染色體設計與解讀方法

在過去文獻中，有使用基因演算法排序 Job Family 和 Job 的研究 (Wilson et al., 2004; Franca et al., 2005; Lin et al., 2009)，皆是在染色體上多設計了 Job Family 的排列，其後再加上原本 Job 之排列，本研究將此設計方法稱為 GA-FT (Traditional Family-based GA)，舉例來說，假設現在有 10 個 Job，3 個 Job Family (與圖 3.3 之情境相同)，GA-FT

的方法是在 Job 之排序前加上 Job Family 之排序，其後再排序 Job 之順序。

如圖 3.15 所示，F2、F3 以及 F1 為 Job Family 之順序，也就是說，必須先加工完為 F2 家族的 Job 後才可再加工為 F3 家族的 Job，最後才可加工為 F1 家族的 Job。而在 Job Family 1 內 Job 的加工順序為 Job2、Job3 及 Job1，在 Job Family 2 內 Job 的加工順序為 Job4、Job5 及 Job6，在 Job Family 3 內 Job 的加工順序為 Job8、Job7、Job10 以及 Job9，故依照過去文獻的排序後可以得到 Job 的加工順序為 Job4、Job5、Job6、Job8、Job7、Job10、Job9、Job2、Job3 及 Job1。

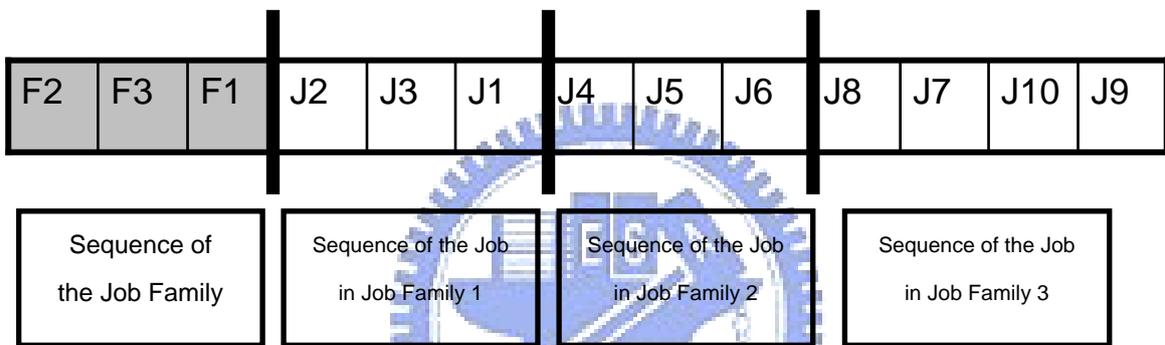


圖 3.15 過去文獻的染色體設計

而本研究所使用之染色體解讀方法，亦可在相同情境下，得到與過去文獻所使用染色體解讀方法的排序結果相同，如圖 3.8 所示。但經由比較圖 3.8 以及圖 3.15 可以發現雖然這兩種染色體解讀方式所得到的結論是相同的，但是染色體所呈現的方式是截然不同，依照圖 3.8 之範例，將相同情境之染色體比較，可以看出本研究之解題空間只有 10 階層，而過去文獻設計的染色體（圖 3.15）則會有 13 階層，這是因為過去文獻的染色體設計多了 Job Family 的排序，解空間更龐大且複雜。再舉另一個規模更龐大的例子，假設現在有 50 個 Job，可以依照 Job 的屬性區分為 10 個 Job Family，以往文獻的染色體設計方式總共會有 (10+50) 階層的解題空間，而本研究的染色體設計方式只會有 50 階層的解題空間，足足少了 Job Family 所產生的解題空間。

總結此兩種染色體設計與解讀方法可以得到一些規則，假設現在有 N 個 Job

Family、 M 個 Jobs，本研究所設計的染色體只會有 M 階層的解空間，而過去的染色體則會有 $(N+M)$ 階層個解空間，也就是說，當 Job Family 的個數越多時，會使得以往的染色體設計更複雜，但由於本研究的染色體在設計時即減去此憂慮，讓 Job Family 的數目不會影響到染色體的長度。

3.7.2 染色體交配與突變方式

GA-FT 染色體交配與突變的方式較為複雜，由於 GA-FT 將 Job Family 的排序加入染色體中，而每個 Job 也都必須依照 Job Family 1、Job Family 2，以及 Job Family 3 的順序排入，故染色體交配與突變的方式會較為繁瑣。

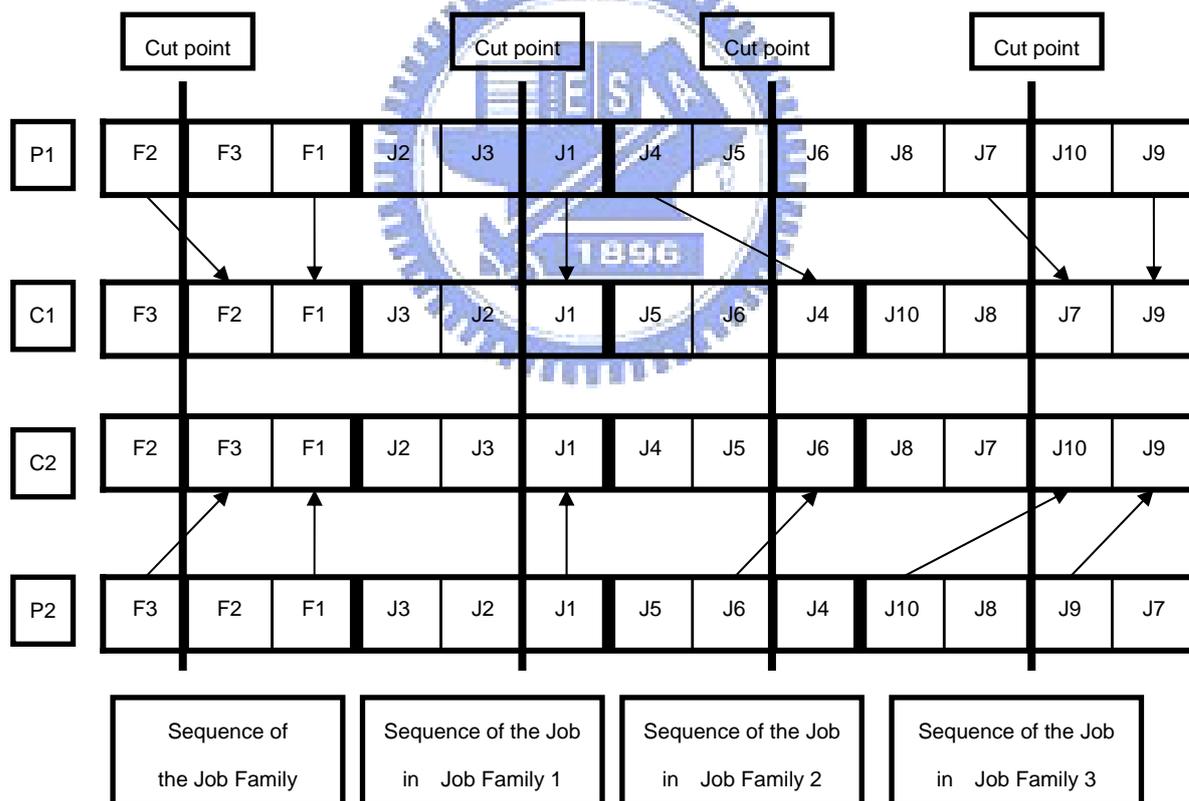


圖 3.16 GA-FT 的 C1 單點交配方式

以 C1 單點交配的方式為例，如圖 3.16 所示，GA-FT 將一條染色體視為 4 條染色體，分別為 Job Family 染色體、Job Family 1 染色體、Job Family 2 染色體，及 Job Family 3 染色體，Job Family 染色體只能跟 Job Family 染色體交配；Job Family 1 染色體只能跟 Job Family 1 染色體交配；Job Family 2 染色體只能跟 Job Family 2 染色體交配；Job Family 3 染色體只能跟 Job Family 3 染色體交配，故 GA-FT 的染色體必須要交配 4 次。

而 GA-FT 染色體的突變原理亦與交配原理相同，將染色體依照 Job Family 與 Job Family 1、2、3 分別突變，即可得到新的子代。

總結以上論述，GA-F 與 GA-FT 染色體交配突變的差異為，GA-F 的染色體視為一條，故每次交配突變時只需進行一次，而 GA-FT 的染色體視為（1+Job Family 數量）條染色體，故每次交配突變時，則需要進行（1+Job Family 數量）次，比 GA-F 更為繁瑣與複雜。



第四章 實例驗證

為了驗證本研究所提出的基因演算法 GA-F 的有效性，在本章節中將針對不同的派工法則做比較，探討其績效的差異。

4.1 測試情境的設計

對於傳輸整合步進機的機台配置與生產作業作如下的假設：

- (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，假設其值以均勻分配(uniform distribution)分佈 $[a_i, b_i]$ (表 4.1)。
- (3) 傳輸整合步進機的整備時間設定為 0 至 6 分鐘。
- (4) 每一個反應室對於到站加工的晶片，採先到先服務。
- (5) 若為不同加工層的生產批，其使用的光罩會不相同，故換批需要換光罩的整備時間。

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	暫存區到埠區(1)	埠區到塗佈 與顯影機(2)	HMDS(3)	冷卻室(4)	塗佈(5)	軟烤(6)	冷卻室(7)
反應室	1	1	2	2	2	2	2
加工時間 (min)	2.5	0.1	1.2	1.2	1.2	[1.2,2.8]	1
加工順序	曝光(8)	晶片邊緣曝光(9)	曝光 後烘烤(10)	冷卻室(11)	顯影(12)	硬烤(13)	冷卻室(14)
反應室	1	1	2	2	2	2	1
加工時間 (min)	[0.75,1.65]	1	[1.2,2.8]	1	[1.2,2.8]	[1.2,2.8]	0.5

為了建構 Job 在不同加工層時需要更換光罩的情境，本研究使用五種不同設置時間的大小，分別為 0、1.5、3、4.5 以及 6 分鐘；以及五種不同光罩數目的大小，分別為 2、3、5、7、以及 10 種。而其他之測試情境為沿用 Wu & Chiou(2009)數據：批數大小為 20 Jobs、40 Jobs 以及 60 Jobs（共三種），批量大小之分佈為 0.15、0.2、0.25、0.3、0.4、0.5、0.6、0.7、0.8、0.9（共十種），全部共有 750 種情境。

4.2 標竿說明

過去 Wu & Chiou (2009) 為研究在小批量情境下作傳輸整合步進機的排程，但因為所使用的方法沒有考量光罩的特性，所以只使用基因演算法，並無納入家族式派工的概念，本研究將此方法稱為 GA-I，故後續將會針對 GA-I 與 GA-F 做探討。但因為以往文獻 (GA-FT) 結合基因演算法與家族式派工所使用的染色體設計方式與本研究的 GA-F 不同 (Wilson et al., 2004; Franca et al., 2005; Lin et al., 2009)，故本研究將會先進行 GA-F 與 GA-FT 之比較，實驗出何種家族式派工的染色體設計方法會最好，接著再針對結果最好的方法與 GA-I 進行比較，探討出使用家族式派工是有必要的。以下為本研究所使用之標竿介紹，分別為 GA-FT 和 GA-I。

第一種標竿為 GA-FT (Traditional Family-based GA)，為第三章所提及的過去染色體設計方式。

第二種標竿為 GA-I (Individual-based GA)，也就是直接經由基因演算法後，無 Family-based 過程，直接選出最佳之排序 (Wu & Chiou(2009)所使用的方法)。

4.3 基因演算法的參數設定

每一個基因演算法的值代表實驗 15 次後取其平均值，目的是為了得到更準確的近似最佳解，避免遇到特殊情形發生而影響最佳解之結果。本研究之基因演算法的參數設定為下：

1. 初始母代染色體數 100 條，交配率：0.8，突變率：0.2。
2. 子代數量 = 交配率 * 母代數量 + 突變率 * 母代數量。
3. 演算停止條件：最多執行的代數 = 100,000 或最佳解維持代數 = 1,000。
4. 交配運算方法為 C1、Lox、Pmx。
5. 突變運算方法為 Swap、Inverse。

4.4 實驗結果與分析

本研究將會先進行 GA-F 與 GA-FT 之比較，實驗出何種家族式派工的染色體設計方法會最好，接著再針對結果最好的方法與 GA-I 進行比較，探討在小批量情境下作傳輸整合步進機的排程證明使用家族式派工是有必要的。

4.4.1 比較不同情境下 GA-F 與 GA-FT 的 Makespan

本研究定義 C_{GA-F} 為 GA-F 的總完工時間， C_{GA-FT} 設為標竿排程指標 GA-FT 的總完工時間。指標為改善率 $r = (C_{GA-FT} - C_{GA-F}) / C_{GA-FT}$ ， r 被設定為量測基因演算法的好壞。當 r 愈高，代表 GA-F 的效果愈好。

圖 4.1 和表 4.2 比較所有測試情境在不同 Job 數下 r 的改變，圖 4.1 表現出在不同 Job 數下，改善率 r 的平均數 (Average)，由圖可以看出在 20 Jobs 時改善率為負值，代表在此情形時使用 GA-FT 的效果會較好，但在 40 Jobs 以上時，GA-F 的改善率為正值，並且會隨著 Job 數的增加而上升，亦代表當 Job 數越多的時候，GA-F 的效果就愈好。

表 4.2 比較所有測試情境在不同 Job 數下 r 的平均數

工件數	20	40	60
改善率平均	-0.0057 %	0.0854 %	0.4515 %

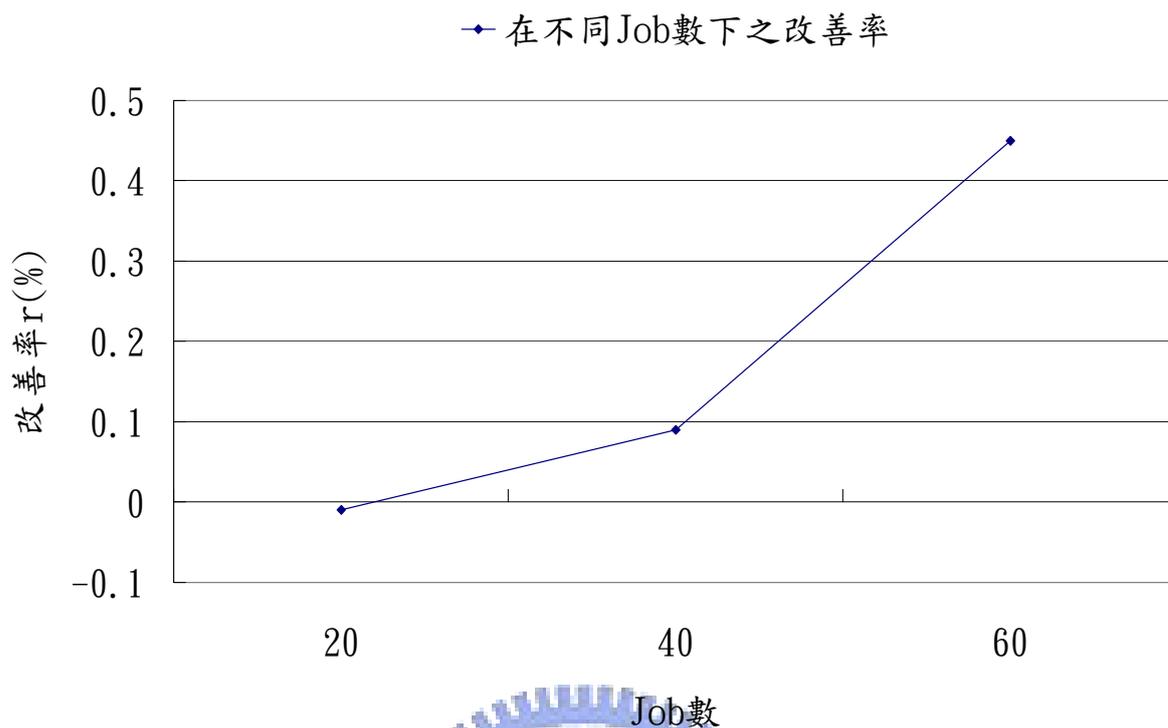


圖 4.1 比較所有測試情境在不同 Job 數下 r 的改變

圖 4.2 和表 4.3 為比較所有測試情境在不同設置時間下 r 的改變，圖 4.2 表現出在不同的設置時間下，改善率 r 之平均數 (Average)，由圖可看出改善率皆為正值，並且在有設置時間下，改善率 r 會隨著設置時間的增加而上升，代表當設置時間越長的時候，GA-F 的效果就愈好。

表 4.3 比較所有測試情境在不同設置時間下 r 的平均數

設置時間	0 分鐘	1.5 分鐘	3 分鐘	4.5 分鐘	6 分鐘
改善率平均	0.1833%	0.1667%	0.1700%	0.1800%	0.1867%

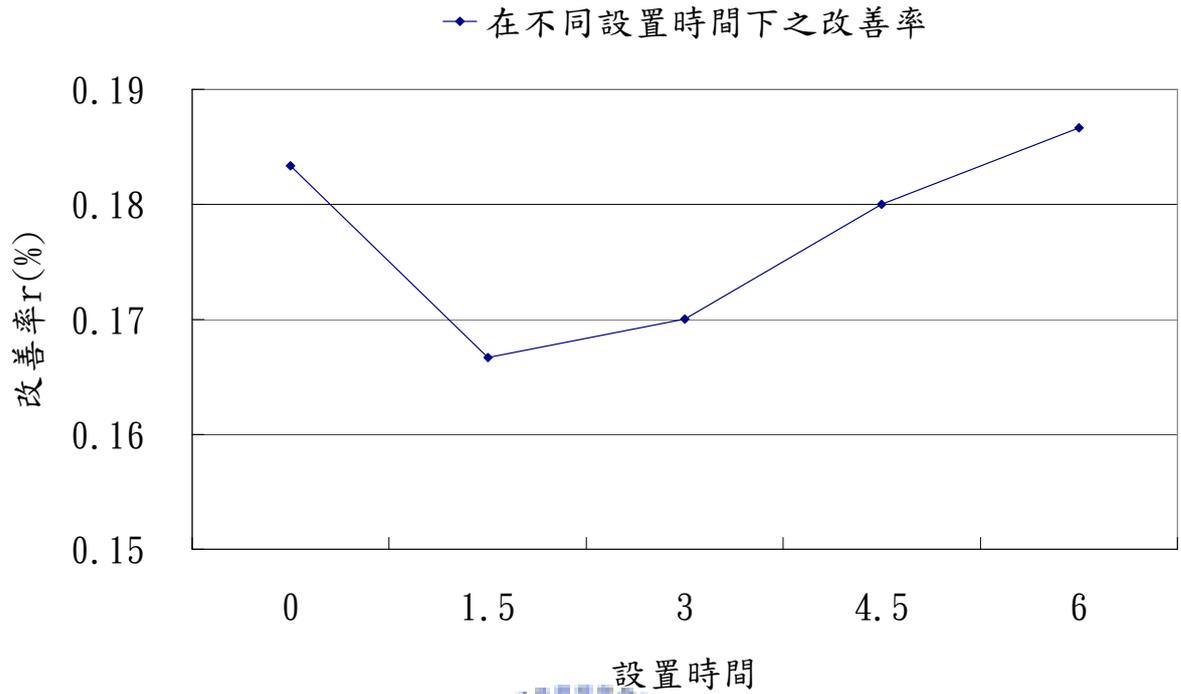


圖 4.2 比較所有測試情境在不同設置時間下 r 的改變

圖 4.3 和表 4.4 為比較所有測試情境在不同光罩數量下 r 的改變，圖 4.3 表現出在不同光罩數量下，改善率 r 之平均數 (Average)，由圖可以看出改善率 r 會隨著光罩數量的減少而上升，亦代表當光罩數量越少的時候，GA-F 的效果就愈好。

表 4.4 比較所有測試情境在不同光罩數量下 r 的平均數

家族數目	2	3	5	7	10
改善率平均	0.2733%	0.2567%	0.1767%	0.1067%	0.0667%

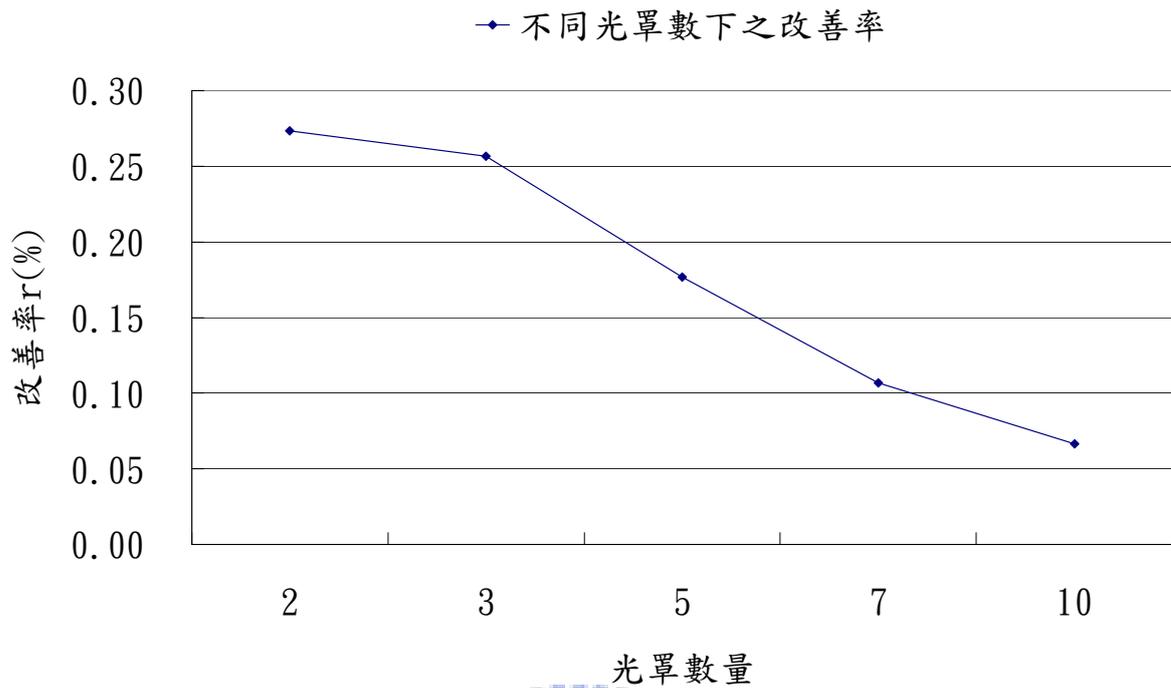


圖 4.3 比較所有測試情境在不同光罩數量下 r 的改變

圖 4.4 和表 4.5 為比較所有測試情境在不同 Yield 分佈下 r 的改變，圖 4.4 表現出在不同 Yield 下，改善率 r 之平均數 (Average)，由圖可以看出改善率 r 會隨著 Yield 的減少而上升，亦代表當 Yield 越小的時候，GA-F 的效果就愈好。

表 4.5 比較所有測試情境在不同 Yield 分佈下 r 的平均數

良率	0.15	0.2	0.25	0.3	0.4
改善率平均	0.4802 %	0.5246 %	0.3461 %	0.1477 %	0.0294 %
良率	0.5	0.6	0.7	0.8	0.9
改善率平均	0.0249 %	0.0194 %	0.0164 %	0.0111 %	0.0099 %

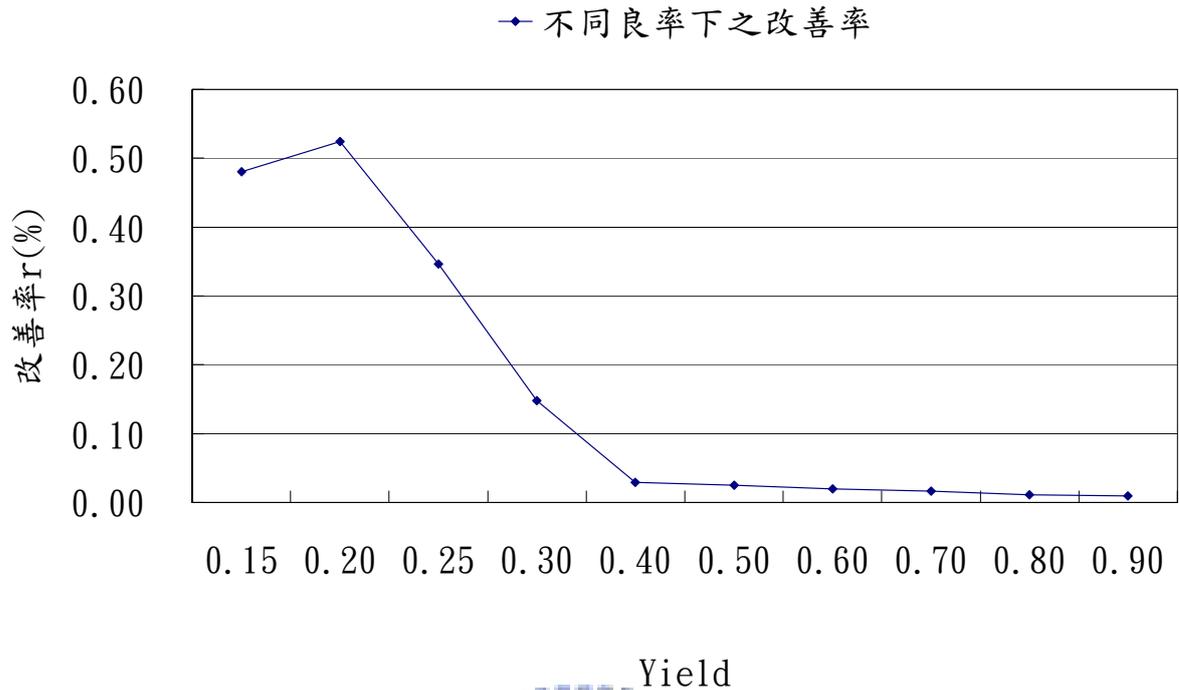


圖 4.4 比較所有測試情境在不同 Yield 分佈下 r 的改變

完成所有實驗情境後，如圖 4.1 至圖 4.4，可以觀察出實驗數據有著明顯的趨勢，並且依照不同的因素可以觀察出四種現象如下：

當 Job 數目越多、設置時間越長、Family 數目越少和 Yield 下降時，改善率 r 會增加，亦表示 GA-F 方法的效果就會越好。

由實驗結果可知，GA-F 比 GA-FT 的解品質好，原因可能是 GA-FT 染色體在每一次基因演化後的改變太大，導致不易產生出好的解，故本研究將 GA-FT 的演化方式改變，其方法稱為 GA-FT1，如圖 4.5 所示，每一次只突變其中一段基因，即 1、2、3、4 擇一，將變動的程度縮小，再視其與 GA-FT 之結果比較。

實驗情境為 40 個 Job，光罩數為 3，良率為 0.25，設置時間為 6 分鐘。在此情境下 GA-FT 的總完工時間為 336.139 分鐘，而 GA-FT1 的總完工時間為 333.929 分鐘，由此可知，GA-FT1 比 GA-FT 的解品質好，故可證明先前的論點。

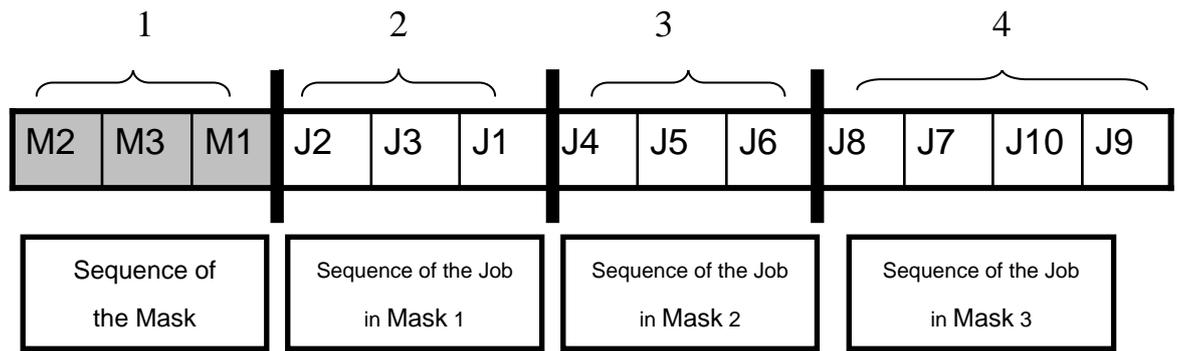


圖 4.5 GA-FT1 染色體表達方式

雖然實驗結果顯示 GA-F 相對於 GA-FT 的改善績效平均都在 1% 以內，但在 Job 數目大且 Yield 低時，改善率可以突破 2%，雖然看似不多，但因為傳輸整合步進機為晶圓廠中最昂貴的機台，所以縱使只有 1% 的改善，也可提升不少獲益。

4.4.2 比較不同情境下 GA-F 與 GA-FT 的求解速度

本研究定義 T_{GA-F} 為 GA-F 的求解秒數；而 T_{GA-FT} 為 GA-FT 的求解秒數。指標 $r_T = T_{GA-FT} - T_{GA-F}$ 被設定為量測 GA-F 求解速度的快慢。當 r_T 為正值，代表 GA-F 的求解速度較快；反過來說，當 r_T 為負值，代表 GA-F 的求解速度較慢。

表 4.6 與圖 4.6 為在各種 Job 數目的情境下 GA-F 與 GA-FT 的求解速度，Average 代表在所有的測試情境中 GA-F 平均比 GA-FT 快的秒數；Max 代表在所有的測試情境中 GA-F 最多比 GA-FT 快的秒數；Min 代表在所有的測試情境中 GA-F 最少比 GA-FT 快的秒數，在 Min 中出現負值則代表 GA-FT 最多比 GA-F 快的秒數，由表 4.6 可觀察出在 20 Jobs 時，為 GA-F 的平均求解時間較快，但 40 Jobs 以上時，GA-FT 的平均求解速度就會比 GA-F 快，並會隨著 Job 數的增加讓求解速度差異加大，但最多在三分鐘以內（176.87 秒）。

表 4.6 比較 GA-F 與 GA-FT 的求解時間差異 (單位：秒)

Jobs	20			40			60		
	Max	Average	Min	Max	Average	Min	Max	Average	Min
$r_T = T_{GA-FT} - T_{GA-F}$	22.73 秒	1.22 秒	-8.13 秒	36.2 秒	-13.13 秒	-87.67 秒	37.47 秒	-69.88 秒	-176.8 秒

在不同Job數下的求解時間比較

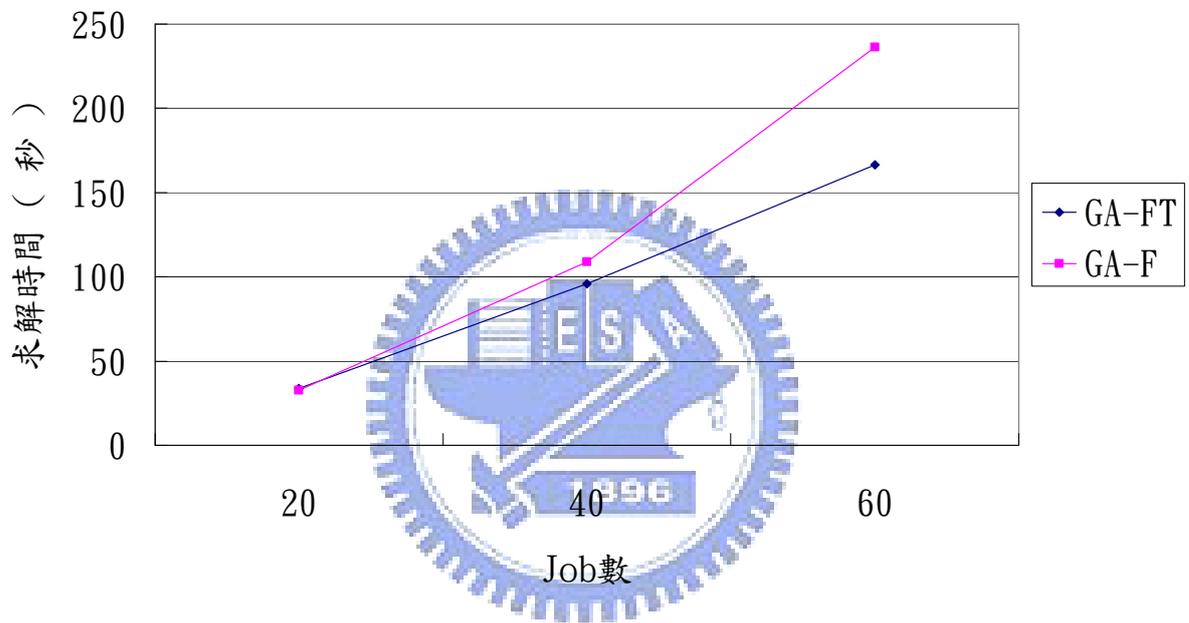


圖 4.6 在各種 Job 數目的情境下 GA-F 與 GA-FT 的求解時間

在晶圓廠中，一天為二班，也就是一天必須排序工件兩次，每次皆排序 40 至 60 個工件，雖然 GA-FT 的平均求解速度比 GA-F 快，但在 60 Jobs 的情況下，至多快 176.87 秒，不到三分鐘，此差異是可以被接受的，因此會將重點放在總完工時間的改善上。

而雖然 GA-F 相對於 GA-FT 的改善效率不大，但在良率低且工件數大時，改善率可達 2.32%，即可提升不少效益，故在家族式派工的方法中，會選擇總完工時間較小且求解時間可接受的 GA-F 染色體設計方式。

4.4.3 比較不同情境下 GA-F 與 GA-I 的 Makespan

上一小節已提過 GA-F 會優於 GA-FT 的染色體設計方式，故本小節將會採 GA-F 與 GA-I 做比較，試驗在含有小批量情境與光罩限制的條件下，使用 Family-based 是否會比使用 Individual-based 會有著更小的總完工時間。接下來將會呈現 GA-F 與 GA-I 比較的實驗結果，首先定義 C_{GA-F} 為 GA-F 總完工時間， C_{GA-I} 設為標竿排程指標 GA-I 的總完工時間，指標 $r = (C_{GA-I} - C_{GA-F}) / C_{GA-I}$ 被設定為量測基因演算法的好壞。當 r 愈高，代表 GA-F 的效果愈好。

表 4.7 和圖 4.7 比較所有測試情境在不同 Job 數下 r 的改變，圖 4.7 表現出在不同的 Job 數下，改善率 r 的平均數 (Average)，由圖可以看出改善率 r 皆為正值，並會隨著 Job 數的增加而上升，亦代表當 Job 數越多的時候，GA-F 的效果就愈好。

表 4.7 比較所有測試情境在不同 Job 數下 r 的平均數

工件數	20	40	60
改善率平均	0.3809%	1.0291%	1.4022%

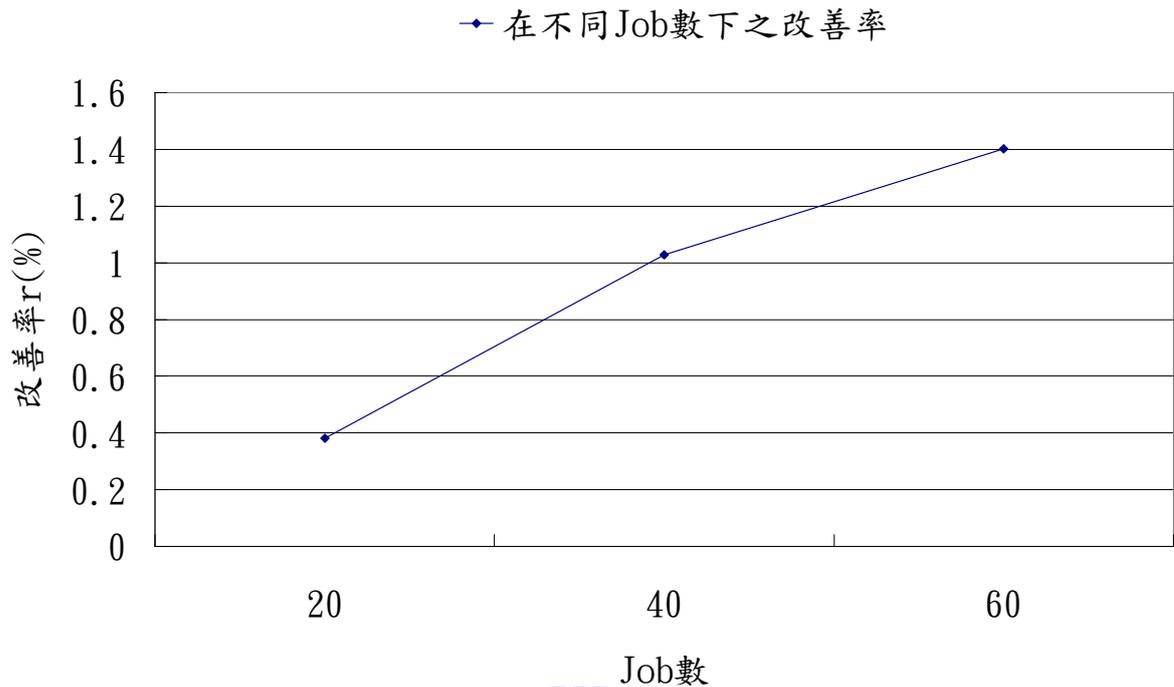


圖 4.7 比較所有測試情境在不同 Job 數下 r 的改變

表 4.8 和圖 4.8 為比較所有測試情境在比較不同設置時間下 r 的改變，圖 4.8 表現出在不同的設置時間下，改善率 r 之平均數 (Average)，由圖可以看出在 Setup Time ≠ 0 時 GA-F 對於其他演算法皆有改善，並且可以觀察出改善率 r 會隨著設置時間的增加而上升，亦代表當設置時間越長的時候，GA 的效果就愈好。

唯獨在 Setup Time=0 分鐘時為 GA-I 的結果最好，這是因為如果不需要設置時間，則任何 Family-based 的方法都無意義，故在不需要設置時間時，本研究的 GA-F 無明顯效果。

表 4.8 比較所有測試情境在不同設置時間下 r 的平均數

設置時間	0 分鐘	1.5 分鐘	3 分鐘	4.5 分鐘	6 分鐘
改善率平均	-0.3567%	0.5800%	1.1233%	1.3167%	1.5167%

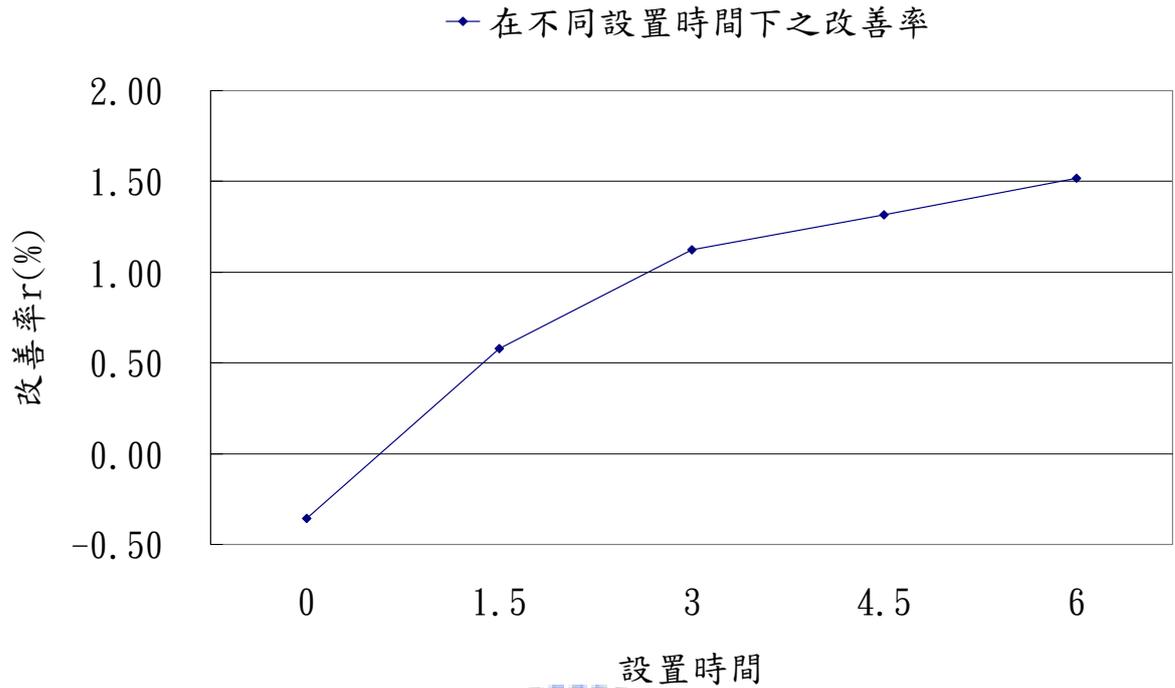


圖 4.8 比較所有測試情境在不同設置時間下 r 的改變

圖 4.9 和表 4.9 為比較所有測試情境在比較不同光罩數量下 r 的改變，圖 4.9 表現出在不同的光罩數量下，改善率 r 之平均數 (Average)，由圖可以看出改善率 r 會隨著光罩數量的增加而上升，亦代表當光罩數量越多的時候，GA-F 的效果就愈好。

表 4.9 比較所有測試情境在不同光罩數量下 r 的平均數和標準差

家族數目	2	3	5	7	10
改善率平均	0.4567%	0.9000%	1.3967%	1.5700%	1.3433%

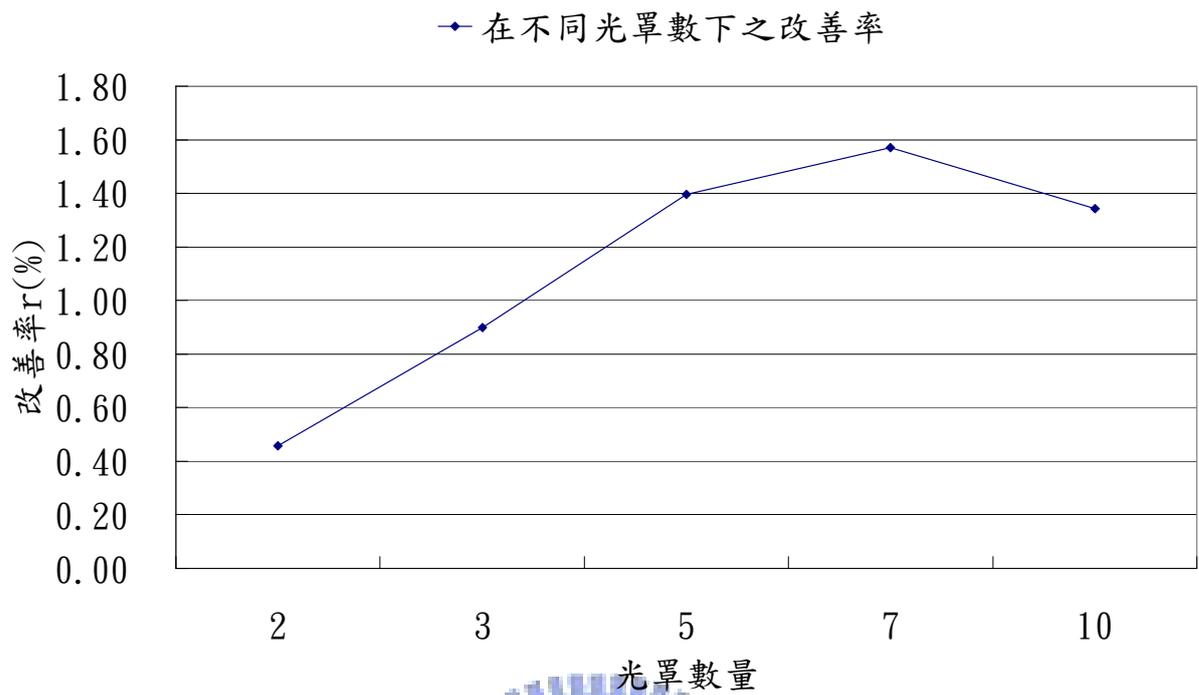


圖 4.9 比較所有測試情境在不同光罩數量下 r 的改變

但當光罩數量較多時，GA-F 的方法就會形成少量 Job 為一個 Job Family，就會與 GA-I 的方法非常相近(因為 GA-I 為 1 個 Job 為一個 Job Family)，故績效會隨著在光罩數量與 Job 數相近時而下降。

圖 4.10 和表 4.10 為比較所有測試情境在比較不同 Yield 分佈下 r 的改變，圖 4.10 表現出在不同 Yield 下，改善率 r 之平均數 (Average)，由圖可以看出改善率 r 會隨著 Yield 的減少而上升，亦代表當 Yield 越小的時候，GA-F 的效果就愈好。

表 4.10 比較所有測試情境在比較不同 Yield 分佈下 r 的平均數

良率	0.15	0.2	0.25	0.3	0.4
改善率平均	2.7778 %	1.6601 %	1.3733 %	0.8483 %	0.4928 %
良率	0.5	0.6	0.7	0.8	0.9
改善率平均	0.3550 %	0.2777 %	0.2356 %	0.1998 %	0.1473 %

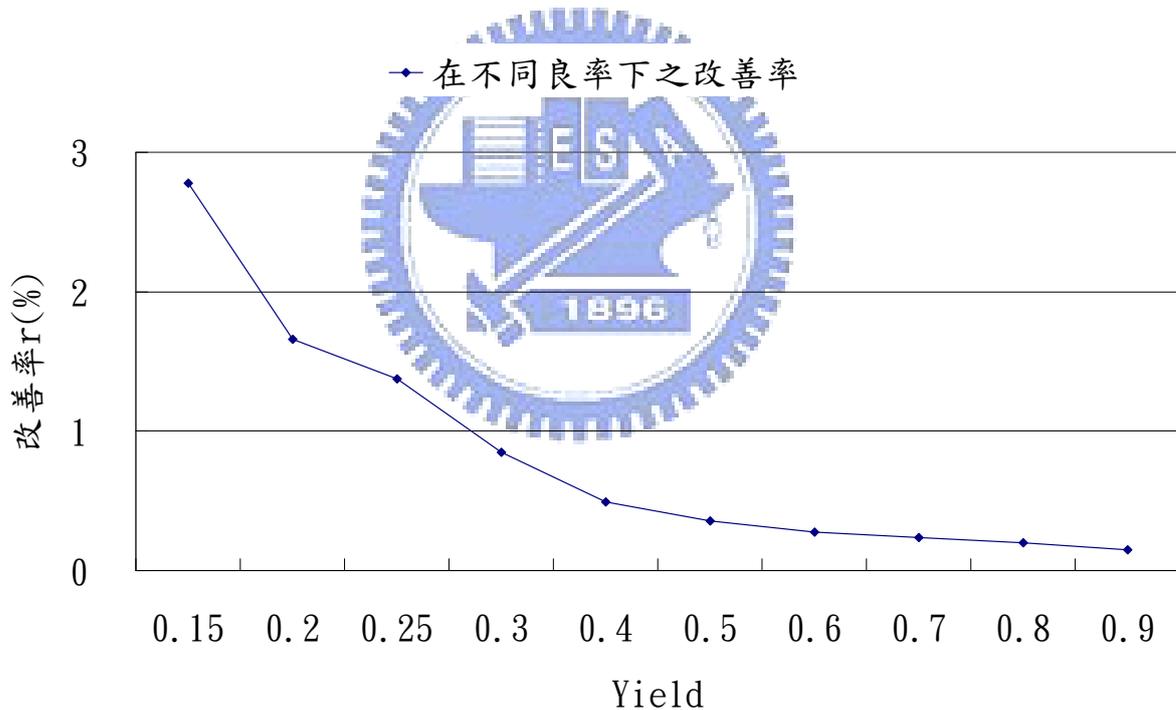


圖 4.10 比較所有測試情境在不同 Yield 分佈下 r 的改變

完成所有實驗情境後，如圖 4.7 至圖 4.10，可以觀察出實驗數據有著明顯的趨勢，並且依照不同的因素可以觀察出四種現象如下：

故當 Job 數目越多、Setup Time 越長、光罩數目越多和 Yield 下降時，改善率 r 會增加，亦表示 GA-F 方法的效果就會越好。

4.4.4 比較不同情境下 GA-F 與 GA-I 的求解速度

在大多數的情形下，本研究的 GA-F 方法會有最小的總完工時間，唯獨在 Setup Time=0 時為 GA-I 的結果最好，這是因為如果不需要設置時間，則任何 Family-based 的方法都無意義，故在 Setup Time=0 時，本研究的 GA-F 無明顯效果。

本研究定義 T_{GA-F} 為 GA-F 的求解秒數；而 T_{GA-I} 為 GA-I 的求解秒數。指標 $r_T = T_{GA-I} - T_{GA-F}$ 被設定為量測 GA-F 求解速度的快慢。當 r_T 為正值，代表 GA-F 的求解速度較快；反過來說，當 r_T 為負值，代表 GA-F 的求解速度較慢。

表 4.11 與圖 4.11 為在各種 Job 數目的情境下 GA-F 與 GA-I 的求解速度，Average 代表在所有的測試情境中 GA-F 平均比 GA-I 快的秒數；Max 代表在所有的測試情境中 GA-F 最多比 GA-I 快的秒數；Min 代表在所有的測試情境中 GA-F 最少比 GA-I 快的秒數，由表 4.11 可觀察出 GA-F 的平均求解速度會比 GA-I 快，並會隨著 Job 數的增加讓求解速度差異加大，求解速度最少快 2.27 秒，最多快十分鐘（591.47 秒）。

表 4.11 比較 GA-I 和 GA-F 的求解時間差異（單位：秒）

Jobs	20			40			60		
	Max	Average	Min	Max	Average	Min	Max	Average	Min
$r_T = T_{GA-I} - T_{GA-F}$	73.67 秒	26.35 秒	2.27 秒	321.67 秒	127.21 秒	4.8 秒	591.47 秒	220.44 秒	4.93 秒

在不同Job數下求解時間之比較

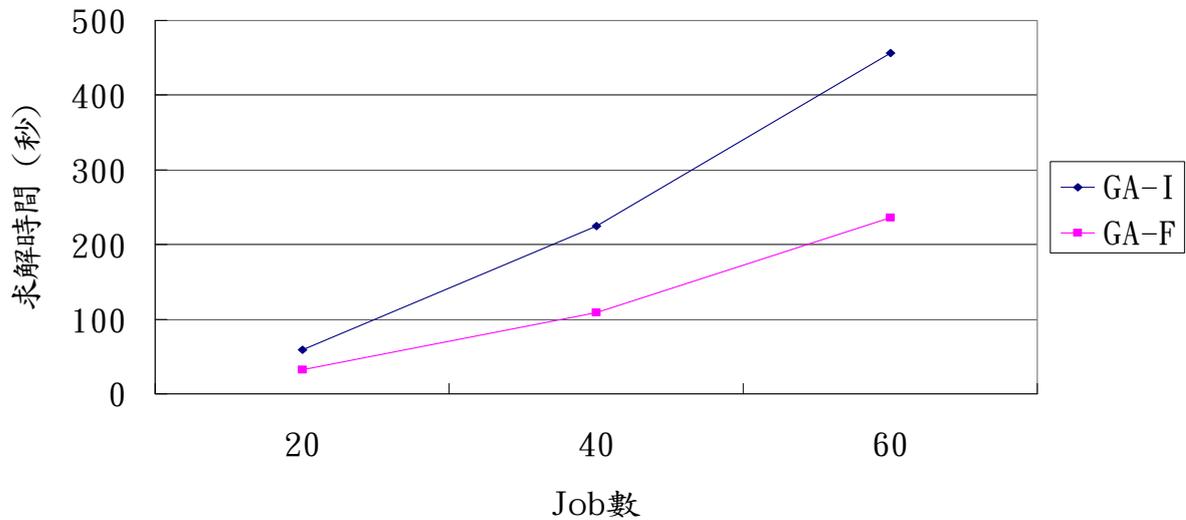


圖 4.11 比較 GA-F 和 GA-F 的求解時間差異

由實驗結果可以發現，GA-F 比 GA-I 的求解速度更快，最多相差 591.47 秒；在解品質方面，也是 GA-F 會比 GA-I 有更小的總完工時間，改善率最高可達 10%，由此可見，在有光罩限制與小批量的情境下，使用家族式派工是必要的，效果比單獨派工法則更好，故使用本研究的 GA-F，可以使半導體傳輸整合步進機的閒置最少，產出最大。

4.5 實驗結論

綜合以上實驗結果，可以將結論整理成以下數項：

- (1) GA-F 可以比 GA-FT 求出更好的工件順序，並且求解速度差異亦不大。
- (2) 在具有光罩限制的條件下，家族式派工法則比單獨派工法則效果更好。
- (3) 在設置時間不等於 0 時，為本研究的 GA-F 效果為最佳，會有最小的總完工時間。
- (4) GA-F 可以比 GA-I 更快速求得近似最佳工序。
- (5) 本研究可以在有光罩限制與小批量情境下求得近似最佳的生產排程組合。

第五章 結論與未來研究方向

在實際業界中，大部分晶圓廠都忽略了傳輸整合步進機在小批量情境下會有產能損失的問題，而過去文獻也都只探討在滿批量下做步進機的派工，故本論文所研究的議題是值得被探討的。

雖然過去 Wu & Chiou (2009) 曾經針對低良率的情境作傳輸整合步進機的排程，但並無考慮到有些晶圓批會有使用相同光罩的情形，然而在現實情況中，晶圓廠是具有迴流製程特性的工廠，故工件會重複進入傳輸整合步進機中加工，而每次皆為加工不同的電路佈局，在加工不同電路佈局時則需要更換光罩，換句話說，相同的產品在同一加工層時（加工相同的電路佈局），會使用相同光罩，故本研究為在考慮光罩限制下，針對小批量的情境來減少產能損失的問題。而本研究所提出之 GA-F (Family-based GA) 可以在短時間內，得到一組近似最佳解的加工順序與總完工時間。

5.1 研究的結論

過去 Wu & Chiou (2009) 為研究在小批量情境下作傳輸整合步進機的排程，但因為所使用的方法沒有考量光罩的特性，所以只使用基因演算法，並無納入家族式派工的概念，也就是本研究的標竿 GA-I。本研究欲探討在此情境下，使用家族式派工是否會比使用單獨派工法則有著更少的機台閒置情形，在第四章中已可得知 GA-F 可以比 GA-FT 求出更好的工件順序，故結論將會針對 GA-I 與 GA-F 做探討。在經過大量的實驗後，可以將本研究的結論歸納如下：

1. 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，而本研究的 GA-F 可以減少更多餘的設置時間。
2. 當設置時間越長時，家族式派工法則的效果越顯著。因為設置時間越長，則 GA-I 所產生的設置時間會更多，造成總完工時間拉長，而本研究所使用的 GA-F 可以減少多餘的設置時間，減少總完工時間。

3. 當所使用的光罩種類越多，家族式派工法則的效果越顯著。因為光罩種類數量越多，工件在加工時遇到使用不同光罩的工件機會也越大，而本研究所使用的 GA-F 可以減少工件遇到使用不同光罩的工件，使總完工時間更短。但值得注意的是，當使用光罩數量與工件數量接近時，則使用 GA-F 效果會趨近於 GA-I，故績效下降。
4. 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，而本論文的 GA-F 減少了多餘的設置時間，故 GA-F 在低良率時成效會比 GA-I 顯著。

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. 本研究目前只探討單一機台的研究，故未來可朝向雙機或多機，甚至是生產系統來做延伸。
2. 本研究目前只探討以基因演算法求解傳輸整合步進機的排程問題，尚未探討以不同的巨集演算法 (Meta-heuristic) 來求解此一問題，故未來可嘗試使用不同的巨集演算法來解決傳輸整合步進機的排程問題。
3. 由於在小批量情境下發生產能閒置問題的主要原因是因為 Port 數量不足，故未來可朝向 Port 數量的研究，探討 Port 達到多少數量時，即可以使傳輸整合步進機不會發生產能的損失。
4. GA-F 的解品質會比 GA-FT 好的主要原因是 GA-FT 的染色體在每次的基因演化後變動太大，導致不易產生好的解，故未來研究方向可朝向針對不同的問題情境的 Family-based，去改善它們的染色體設計。

參考文獻

Bagchi TP, Gupta NDJ and Sriskandarajah C (2006). A review of TSP based approaches for flowshop scheduling. *European Journal of Operational Research* **169**: 816-854.

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.

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.

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

Hejazi SR and Saghaifian S (2005). Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research* **43**: 2895-2929.

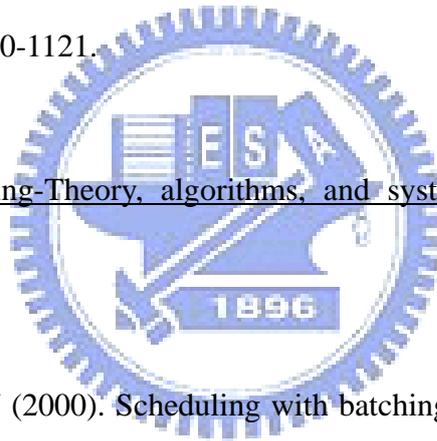
Holland JH (1975). Adaptation in Neural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan .

Liao CJ, Liao LM and Tseng CT (2006). A performance evaluation of permutation vs. non-permutation schedules in a flowshop. *International Journal of Production Research* **44**: 4297-4309.

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.

Lin SW, Ying KC and Lee ZJ (2009). Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times. *Computers & Operations Research* **36**: 1110-1121.

Pinedo M (2002). Scheduling-Theory, algorithms, and systems. 2nd, Prentice Hall, New Jersey.



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

Quadt D and Kuhn H (2007). A taxonomy of flexible flow line scheduling procedures. *European Journal of Operational Research* **178**: 686-698.

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

Reeves CR (1995). A genetic algorithm for flowshop sequencing, *Computer Operation Research* **22**: 5-13.

Ruiz R and Stutzle T (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* **177**: 2033-2049.

Ruiz R and Maroto C (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research* **169**: 781-800.

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

Wang L and Zheng DZ (2003). An effective hybrid heuristic for flowshop scheduling. *International Journal of Advanced Manufacturing Technology* **21**: 38-44.

Wein LM (1988). Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing* **1**: 115-130.

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.

Xiao H (2001). Introduction to semiconductor manufacturing technology. Prentice Hall, New Jersey.

Ying KC and Lin SW (2006). Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach. *International Journal of Production Research* **44**: 3161-3177.

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.

Zhu X and Wilhelm WE (2007). Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transaction* **38**: 987-1007.

