

行政院國家科學委員會補助專題研究計畫  成果報告  
 期中進度報告

新產品/製程導入情境下 TFT-LCD 產業的製造管理

計畫類別： 個別型計畫  整合型計畫

計畫編號：NSC 96 - 2221 - E - 009 - 034 - MY3

執行期間：2007 年 8 月 1 日至 2010 年 7 月 31 日

執行機構及系所：國立交通大學工業工程與管理學系

計畫主持人：巫木誠

共同主持人：無

計畫參與人員：盧威豪、吳政翰、陳德珊、呂佳玟、林慈盈、陳文旻、  
林昭宏、陳曉嚴、戴邦豪、謝佩芸、凌金傳、鐘崇毓

成果報告類型(依經費核定清單規定繳交)： 精簡報告  完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：

赴國外出差或研習心得報告

赴大陸地區出差或研習心得報告

出席國際學術會議心得報告

國際合作研究計畫國外研究報告

處理方式：除列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年  二年後可公開查詢

中 華 民 國 99 年 10 月 31 日

## 國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

## 1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

達成目標

未達成目標（請說明，以 100 字為限）

實驗失敗

因故實驗中斷

其他原因

說明：

## 2. 研究成果在學術期刊發表或申請專利等情形：

論文：已發表 未發表之文稿 撰寫中 無

專利：已獲得 申請中 無（已經獲得四項發明專利）

技轉：已技轉 洽談中 無

其他：（以 100 字為限）

## 3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

本研究目前已經獲得四項專利，三項為中華民國專利，一項為大韓民國專利。

茲將所獲得之四項專利列述如下：

[1] 巫木誠、盧威豪、吳佳恩「物料報廢架構及物料報廢準則產生方法」，中華民國發明專利（證書號數：I323427）

[2] 巫木誠、陳德珊：「基板排序派工之方法」，大韓民國專利，（證書號數：10-0816624）

[3] 巫木誠、陳德珊：「基板排序派工之方法」，中華民國專利，（證書號數：I315476）

[4] 巫木誠、吳政翰：「相依排程之派工方法」，中華民國發明專利（證書號數：I313437）

## 國科會補助計畫衍生研發成果推廣資料表

日期：99年10月28日

國科會補助計畫	計畫名稱：新產品/製程導入情境下 TFT-LCD 產業的製造管理 計畫主持人：巫木誠 計畫編號：NSC 96-2221-E-009-034-MY3 領域：工業工程		
研發成果名稱	(中文) TFT-LCD 產業製造管理相關專利四件		
	(英文) Four Patents for Enhancing the Manufacturing Management of TFT-LCD Industries		
成果歸屬機構	國立交通大學	發明人 (創作人)	巫木誠、盧威豪、吳佳恩、陳德珊、吳政翰
技術說明	<p>[1] 巫木誠、盧威豪、吳佳恩「物料報廢架構及物料報廢準則產生方法」，中華民國發明專利(證書號數：I323427)</p> <p>[2] 巫木誠、陳德珊：「基板排序派工之方法」，大韓民國專利，(證書號數：10-0816624)</p> <p>[3] 巫木誠、陳德珊：「基板排序派工之方法」，中華民國專利，(證書號數：I315476)</p> <p>[4] 巫木誠、吳政翰：「相依排程之派工方法」，中華民國發明專利(證書號數：I313437)</p>		
	This research has developed several methods for enhancing the manufacturing management of TFT-LCD industries. Four patents have been granted, in which three are granted by Taiwan, ROC and one is granted by South Korea.		
產業別	TFT-LCD 製造業		
技術/產品應用範圍	<p>(1) 低良率情境下，基板重工決策門檻之決策</p> <p>(2) 低良率情境下，基板重新配對之決策</p> <p>(3) 基板重新配對情境下，排序機之派工</p> <p>(4) 異質專案之排程</p>		
技術移轉可行性及預期效益	在新製程導入時，TFT-LCD 的良率通常很低，運用本研究所開發之方法，可有效提升產出量，增加獲利。		

註：本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。

## 國科會補助專題研究計畫項下出席國際學術會議心得報告

日期：99 年 10 月 28 日

計畫編號	NSC 96 - 2221 - E - 009 - 034 - MY3		
計畫名稱	新產品/製程導入情境下 TFT-LCD 產業的製造管理		
出國人員姓名	巫木誠	服務機構及職稱	國立交通大學工業工程與管理系
會議時間	98 年 12 月 14 日 至 98 年 12 月 16 日	會議地點	Kitakyushu, Japan (北九州、日本) -
會議名稱	Proceedings of Asia Pacific Industrial Engineering & Management Systems Conference (APIEMS) 2009, Kitakyushu, Japan, Dec. 14-16, 2009.		
發表論文題目	(中文)大型半導體廠房之物料搬運系統分析 (英文) An Analysis of Material-Handling Capacity for a Large Scale Semiconductor Fab		

一、參加會議經過：APIEMS（亞太工業工程與管理會議）是亞太地區最重要的工業工程會議，我國是創始會員國，國內有多位教授擔任理事，我國工業工程界在此會議頗有影響力。此會議到 2009 年已經舉辦 9 屆，此次會議約有 300 百人參加，最主要是我國、日本、韓國的學者，其他還有大陸、香港、泰國、馬來西亞、菲律賓、印尼等國家，APIEMS 這個會議的重要性越來越大。此次會議，本人除了發表學術論文之外，也同時參加 APIEMS 的理事會會議，也參加 IFPR-Asia Region 的理事會會議；在此會議中，本人也同時受邀擔任 IEMS Journal Special Issue 的 Co-editor。

## 二、與會心得

- 韓國的工業工程學術水準大幅提昇，許多研究都做得很好，產業和學術價值兼具，實在很值得學習和肯定。

## 三、考察參觀活動(無是項活動者略)：無

## 四、建議

- 國內學者宜加強與產業合作，研究的題材可以本土化(Local Impact)，期能對國內產業的競爭力有幫助；但是研究成果要能國際化(Global Excellence)，亦即能在一流國際期刊發表，以提升我國學術研究水準，也驗證本土化研究成果之水準。

## 五、攜回資料名稱及內容：

- 2009APIEMS 大會學術論文集光碟

## 六、其他

## 國科會補助專題研究計畫項下赴國外(或大陸地區)出差或研習心得報告

日期：99 年 10 月 28 日

計畫編號	NSC 96 - 2221 - E - 009 - 034 - MY3 -		
計畫名稱	新產品/製程導入情境下 TFT-LCD 產業的製造管理		
出國人員 姓名	巫木誠	服務機構 及職稱	國立交通大學工業工程與管理系
出國時間	年 月 日至 年 月 日	出國地點	

- 不適用：在本專題計畫下，本人未曾到國外（或大陸）出差或參加研習

一、國外(大陸)研究過程

二、研究成果

三、建議

四、其他

## 國科會補助專題研究計畫項下國際合作研究計畫國外研究報告

日期:99年10月28日

計畫編號	NSC 96 - 2221 - E - 009 - 034 - MY3		
計畫名稱	新產品/製程導入情境下 TFT-LCD 產業的製造管理		
出國人員 姓名	巫木誠	服務機構 及職稱	國立交通大學工業工程與管理系
合作國家		合作機構	
出國時間	年 月 日至 年 月 日	出國地點	

- 不適用：在本專題計畫下，本人未參加任何國際合作。

一、國際合作研究過程

二、研究成果

三、建議

四、其他

## 中文摘要

薄膜電晶體顯示器(TFT-LCD)已成為我國一個重要的產業，此產業在新製程或新產品剛導入時，通常良率不高，而且良率出現的位置和時點，都具有隨機的性質。由於市場變動很快，新產品新製程會不斷導入，TFT-LCD 產業在低良率情境時，應如何有效進行製造管理，對我國 TFT-LCD 產業實在很重要；但是過去的學術文獻甚少探討。吾人乃針對此議題進行一個三年的研究計畫，研究 TFT-LCD 產業在低良率情境的製造管理議題，此計畫包括下列三個研究子題：(1)基板重工決策、(2)基板配對與重組決策、(3) 產品組合與投料決策。本研究成果豐碩，目前已經獲得四個國內外專利，對提升我國 TFT-LCD 產業的競爭力略有貢獻。

子題一（基板重工決策）的研究問題與成果可簡述如下：在 TFT-LCD 廠面臨低良率情境時，TFT-Plate 製程及 CF-Plate 製程的各加工檢測站，應如何根據其品質產出狀況，決定是否繼續加工或是重工，以使總獲利最大。本研究建構一數學模式，考量範圍包含 TFT 製程、CF 製程及 Cell 製程的特性，利用基因演算法，可以在整體獲利最大的準則下，訂定個別產品在 TFT 及 CF 兩製程，各檢測站的最佳重工決策。本子題所發展之決策方法已經獲得一項中華民國專利。

子題二（基板配對與重組決策）的研究問題與成果可簡述如下：TFT-Plate 和 CF-Plate 都是批次(Batch)加工作業，一個 TFT-Plate 和 CF-Plate 組合後，才能變成一個 TFT-LCD Plate。由於各基板的良率不同，為提升組合的良率，各批次的基板必須重組，基板重組的抽換是藉排序機(Sorter)進行。此研究情境有兩個決策問題，其一是基板重組決策，其二是排序機的派工決策。本子題運用動態規劃法求解，此法分別獲得中華民國專利與大韓民國專利。

子題三（產品組合與投料決策）的研究問題與成果可簡述如下：給定一產品組合和產出量，根據前二子題的研究，我們可求得最佳化的重工決策、重組決策、派工決策，進而求其最大之獲利。若將各種產品組合與產出量的所有可能解，視為一個解空間，吾人可以基因演算法，求解出最佳的產品組合與產出量。在研究此問題時，本研究也發展出一個異質專案系統的排程方法，並獲得一項中華民國專利。

關鍵詞：低良率情境、TFT-LCD 產業製造管理、基板重工、基板配對與重組、產品組合、專案排程、

## Abstract

TFT-LCD has become an important industry in Taiwan. In the new product/process introduction (NPI) stages, the process yield is usually low and behaves in a stochastic manner. Due to volatile market demand, NPI appear constantly. Developing manufacturing management techniques for NPI in TFT-LCD industry is thus very important. Yet, rare literature has addressed such research issues. We therefore carried out a 3-year research project to address the manufacturing management issues in a low-yield TFT-LCD scenario. We focused on three research topics in NPI scenarios: (1) TFT/CF plate rework decision, (2) TFT/CF plate mapping/sorting decision, and (3) product mix/releasing decisions. The research results yield four patents, three granted by Taiwan, ROC; and one granted by South Korea. These patented techniques would help enhancing the productivity of Taiwan TFT-LCD industry.

Research problems and results of the first topic (TFT/CF plate rework decision) are briefly described below. For a TFT-LCD factory in a low-yield scenario, we need to determine whether a low-yield TFT-Plate/CF-plate should be reworked or proceed for further processing. We develop a mathematical model for computing the profit given a portfolio of rework decisions, and use a genetic algorithm to find an optimal portfolio rework decisions in order to maximize the total profit. The developed method has been granted a patent by Taiwan, ROC.

Research problems and results of the second topic (plate matching and sorter scheduling) are briefly described below. TFT-Plates and CF-Plates are both manufactured in batches. Each batch may include various qualities of plates. To make an optimal pairing match among TFT-Plate and CF-Plate batches, the plates in the batches should be sorted (or plates are exchanged among batches). This leads two decisions: plate sorting, and scheduling of sorter. We develop a dynamic programming model to solve the two decisions, which has received two patents, one granted by Taiwan, ROC and the other by South Korea.

Research problems and results of the third topic (product mix/releasing decisions) are briefly described below. Based on the developed methods of the first two issues, we can compute the profit for a particular product mix and releasing portfolio. Based on a meta-heuristic algorithm, we can find a near-optimal product mix and releasing portfolio. On examining this issue, we unexpectedly develop a meta-heuristic algorithm for solving the scheduling of a heterogeneous project, which has also received a patent granted by Taiwan, ROC.

Keywords: Low yield scenario, TFT-LCD, rework, plate mapping/sorting, product mix, lot release



## 圖目錄

圖 3.1	基板重工問題描述	8
圖 3.2	TFT 基板與 CF 基板良率影響配對良率說明	9
圖 3.3	TFT-LCD 基板等級分類	9
圖 3.4	模組一運算流程	12
圖 4.1	TFT 基板和 CF 基板的配對良率	15
圖 4.2	基板抽換作業	16
圖 4.3	排序機設備	16
圖 4.4	配對矩陣	17
圖 4.5	排序機派工決策	18
圖 4.6	建構 DP 網路流程	19
圖 4.7	基本網路架構	20
圖 4.8	三種狀態組態	20
圖 4.9	完整的基本網路架構	21
圖 5.1	DAG 任務	24
圖 5.2	3 個機器的異質系統	24
圖 5.3	有傳輸時間的 DAG 任務	25
圖 5.4	由任務的優先值轉換為染色體	29
圖 5.5	PWP_matrix: $M$ 流程	31
圖 5.6	任務優先關係矩陣	31
圖 5.7	模擬 6 個任務 從 PWP_matrix: $M$ 標準化後的 $M$	31
圖 5.8	PWP_matrix 的硬性限制與柔性限制	32
圖 5.9	PWP 任務優先值去排序: $q_2 \rightarrow q_3 \rightarrow q_1 \rightarrow q_4$	32
圖 5.10	選擇參考任務 $q_r$	34
圖 5.11	異質專案排程實驗結果	36

## 表目錄

表 3.1	基板重工決策案例一之績效比較-----	13
表 3.2	基板重工決策案例二之績效比較-----	13
表 3.3	基板重工決策案例三之績效比較-----	14
表 4.1	基板配對排序機各種排序方法的績效比較-----	22
表 5.1	異質專案排程之染色體表達與解讀-----	28

# 第一章 緒論

## 1.1 研究動機

薄膜電晶體顯示器(TFT-LCD)已成為我國一個重要的產業，此產業在新製程或新產品剛導入時，通常良率不高，而且良率出現的位置和時點，都具有隨機的性質。由於市場變動很快，新產品新製程會不斷導入，TFT-LCD 產業在低良率情境時，應如何有效進行製造管理，對我國 TFT-LCD 產業實在很重要；但是過去的學術文獻甚少探討。

吾人乃針對此議題進行一個三年的研究計畫，研究 TFT-LCD 產業在低良率情境的製造管理議題，此計畫包括下列三個研究子題：(1)基板重工決策、(2)基板配對與重組決策、(3) 產品組合與投料決策。本研究成果豐碩，目前已經獲得四個國內外專利，對提升我國 TFT-LCD 產業的競爭力略有貢獻。

## 1.2 研究子題

子題一（基板重工決策）的研究問題與成果可簡述如下：在 TFT-LCD 廠面臨低良率情境時，TFT-Plate 製程及 CF-Plate 製程的各加工檢測站，應如何根據其品質產出狀況，決定是否繼續加工或是重工，以使總獲利最大。本研究建構一數學模式，考量範圍包含 TFT 製程、CF 製程及 Cell 製程的特性，利用基因演算法，可以在整體獲利最大的準則下，訂定個別產品在 TFT 及 CF 兩製程，各檢測站的最佳重工決策。

子題二（基板配對與重組決策）的研究問題與成果可簡述如下：TFT-Plate 和 CF-Plate 都是批次(Batch)加工作業，一個 TFT-Plate 和 CF-Plate 組合後，才能變成一個 TFT-LCD Plate。由於各基板的良率不同，為提升組合的良率，各批次的基板必須重組，基板重組的抽換是藉排序機(Sorter)進行。此研究情境有兩個決策問題，其一是基板重組決策，其二是排序機的派工決策。

子題三（產品組合與投料決策）的研究問題與成果可簡述如下：給定一產品組合和產出量，根據前二子題的研究，我們可求得最佳化的重工決策、重組決策、派工決策，進而求的其最大之獲利。若將各種產品組合與產出量的所有可能解，視為一個解空間，吾人可以基因演算法，求解出最佳的產品組合與產出量。

## 1.3 研究成果

本研究計畫的研究成果，到目前為止，共獲得四項專利，分述如下：

- 子題一：獲得一項中華民國專利 [1]
- 子題二：獲得一項中華民國專利 [2]、一項大韓民國專利 [3]
- 子題三：獲得一項中華民國專利 [4]

## 1.4 章節安排

本報告其餘章節安排如下：第二章是每個子題的文獻回顧。第三章說明子題一的問題、解法、和成果。第四章說明子題二的問題、解法和成果。第五章說明子題三的問題、解法、和成果。第六章是結論和建議。

## 第二章 文獻探討

本章回顧各主題的相關文獻。首先是討論基板重工的相關文獻，其次討論基板重組與排序的文獻，最後討論專案排程的文獻。

### 2.1 基板重工相關文獻

雖然針對 TFT-LCD 廠的重工決策問題，在過去的學術文獻上，經過吾人的檢索查閱，尚未發現有相關的研究，但隨著原物料成本的提高，以及回收再利用觀念的宣導，許多產業都會將不良品利用重工的過程重新加工[5]，所以有幾篇針對不同產業重工議題的研究文獻衍生而出。

過去有不少文獻針對印刷電路板組裝作業(Printed circuit board assembly, 簡稱 PCBA)的重工議題加以探討，但主要著重在 PCBA 的自動化重工技術(automated PCBA rework)的建立以及發展[6-7]；以及探討自動化重工實施情形的一些成本結構及績效衡量分析[8]；並在成本與生產週期(cycle time)的考量下，決定自動化重工系統進行的時點[9]以及實行重工的策略(定期或是不定期)[10]

上述文獻主要針對組裝產業，在製造加工業方面，也有一些針對重工現象加以探討的文獻，除了有針對重工策略面分析的研究[11-12]，針對不同績效考量，探討一批欲加工的物件，在進行批次加工作業時，若一部分物件為不良品時，原批是否應該等那些不良物件重工完後再併批加工，還是直接往下加工或與其他良品併批加工，或是發展其他重工策略；也有一些文獻依據不同的績效考量，決定當加工達到多少數量之後進行重工[13-14]為最佳，以及決定不良品個數達到某數量之後進行重工[15-16]為最佳。

此外還有幾篇是與本研究問題背景最為相似的半導體產業所作的重工或是報廢決策研究(報廢決策類似於重工決策，不同在於報廢決策為低於門檻值即丟棄，不加以回收重新利用)；Daigle 等人[17]建構一管理模式來減少半導體廠中晶圓(wafer)報廢的情形發生。Michael 等人[18]同時考量重工策略以及派工法則下，決定最佳的派工法則與重工策略組合。Maynard 等人[19]發展出一啟發性解法，在良率及成本的考量下，可求出半導體廠中，晶圓批的批量由於製程良率的問題，導致某些並非包含全良品晶圓的「小量晶圓批」(small wafer lot)，在哪一層(layer)進行報廢為最佳。

另外 Wu 等人[20-21]利用基因演算法，探討半導體廠中小量晶圓批的報廢決策問題，此文獻的問題架構最為接近本研究，在於此篇文獻與本研究相同也是利用門檻值的方法，針對其中幾個關鍵加工層(critical layer)設置報廢決策的決策點，考量單一產品下，以半導體廠的利潤最大化為目的，決定各關鍵加工層的報廢決策門檻值(threshold)，良好晶圓片數高於門檻值的晶圓批可繼續進行加工，反之該晶圓批則逕行報廢。

而雖然 TFT-LCD 廠的 Array 製程及 CF 製程類似半導體廠的製程，但是至 Cell 製程時，TFT-LCD 廠會有半導體廠所沒有的將兩片不同的基板(TFT 基板及 CF 基板)進行結合的組立製程，這也是 TFT-LCD 廠與半導體廠相當大的差別，因為組立製程的一些特性影響，所以兩者在重工(報廢)決策的考量上也將會有所

差異，這也是本研究加以延伸探討的主要部分。

半導體廠報廢決策往往是以批(lot)為單位，低於門檻值的晶圓批上的多片晶圓皆報廢，而 TFT-LCD 廠則是以基板為單位，低於門檻值的基板，該基板上的面板皆報廢(重工)，所以批相當於基板，晶圓相當於面板，這是兩種產業在報廢(重工)決策的類比之處。

## 2.2 基板配對與排序的相關文獻

過去 TFT-LCD 製造管理的相關文獻，主要可分為兩大類，第一類為基板配對決策方法之研究，而第二類則著重在排序機派工決策之研究。

在基板配對決策的研究中，主要有三篇主要的文獻，分別是楊毅臻 [22]、楊佳翰[23]、]及張東華[24]，在這部分的研究已有最佳化配對良率的演算法，因此不再進行探討。而在排序機派工決策方面，則有楊佳翰[23]、楊毅臻[22]及王君豪[25]三篇文獻，分別在以下進行簡介。

為縮短卡匣上下排序機的次數，楊佳翰[23]提出以基因演算法(Genetic algorithm, GA)及模擬退火法(Simulated Annealing, SA)發展良率配對的方法。該研究假設排序機有 1 個輸出埠、 $k$  個輸入埠，若有  $N$  組卡匣的基板需要配對，該研究先將卡匣分群，一群包含有  $k$  個 TFT 卡匣和  $k$  個 CF 卡匣，然後以線性規劃法(Linear Programming, LP)求解一群內基板配對的最佳解。該研究可有效縮短卡匣上下排序機的次數，但是所規劃的基板配對良率未必是最佳解。

為確保基板配對良率，進而縮短卡匣上下排序機的次數，楊毅臻[22]應用匈牙利指派法求解良率配對，然後發展啟發式方法(Heuristic Methods)來決定目標卡匣產出順序(簡稱產出順序)及會員卡匣上下排序機的順序(簡稱投入順序)，期以最小化排序機的總作業時間。該研究雖然可確保最佳配對良率，然而，在求解良率配對的線性規劃問題時，該研究假設該線性規劃問題只有一組最佳解，並據以求解排序機派工問題。

為配合 TFT-LCD 連續型生產的特性，王君豪[25]發展基因啟發式演算法來求解排序機派工問題，其研究方法利用啟發式解法進行求解投入順序問題(input sequencing problem)，下游組立壓合機台(assembly machine)利用率為績效衡量指標，並以此作為基因演算法的適應函數，來進行規劃目標卡匣的產出順序(output sequencing problem)，此基因啟發式解法同時整合排序機作業及下游組立機台作業，以整體性的觀點增進組立製程的生產效率。

針對上述文獻回顧，吾人可以發現王君豪雖以整體性的觀點來進行排序機派工決策，但在投入順序問題上仍有改善之空間，因此本研究的重點在發展一動態規劃法(Dynamic Programming, DP)以改善王君豪的研究中，使用啟發式解法求解投入順序決策問題不足的部分。

## 2.3 異質專案排程相關文獻

有關異質專案排程的文獻，根據 Kolisch [13]的研究，此等排程問題，用逐步求解法(serial method)會比用同步求解法(parallel method)好。以下分析同步與逐步兩種解題架構，並介紹在各解題架構下所發展的演算法。

同步求解法是同步求解指派問題和排序問題。此種求解法譬如 Wang[14]所提的基因演算法，此演算法的染色體是由兩個字串(string)組成，第一個為排程字串，第二個為指派字串，排程字串不得違反任務的順序限制。由於逐步求解法績效較佳，目前已成為求解順序限制(sequence constraint)排程問題的主流。

逐步求解法則是先求解指派問題，再求解排序問題；亦即給定一個任務順序，經過指派決策，可求得此任務順序的績效，因此指派決策基本上一個績效評估機制(performance evaluator)。根據此績效評估機制，排序問題就是在龐大解空間中，找出最佳的任務順序(task sequence)，所以排序決策基本上是一個搜尋機制(search mechanism)。因此一個逐步求解架構的演算法，是由兩個機制所構成：績效評估機制（以下簡稱 evaluator）、搜尋機制(以下簡稱 Search Engine)。

DHC 排程問題常見的評估機制有下列四種：LIT (Longest Idle Time), EST [23](earliest start time), EFT(earliest finish time), DS (dependent task scheduler)。

LIT 法把任務優先指派給閒置時間最長(longest idle machine)的機器，此評估機制相當著名而且簡單的。EST 是將任務指派給可以最早開始加工的機器。EFT 是將任務指派給可以最早完工的機器。DS 是 EFT 方法的修正，若能不增加目前總完工時間(make span)，優先將任務指派給加工時間(processing time)與傳輸時間相加最短的機器。若會增加目前總完工時間(make span)，用 EFT 指派。根據過去研究，上述的指派方法 DS 相對較佳。

搜尋機制的各種方法基本尚可分為：啟發式(Heuristic approach)和演化式(evolutionary approach)和非演化式 (non-evolutionary approach)三大類。啟發式一般皆為 LS(list scheduling)的延伸，LS 為先對任務給定優先值，根據優先值排程。Sih and Lee 學者提出一個 DLS(dynamic Level scheduling)演算法 [11]，此方法根據任務先後關係和各機器執行時間的平均組合等級，以任務根據等級去排程。Topcuoglu 等學者也提出類似概念的 CPOP (Critical-Path-on-a-Processor)啟發法 [25]去排程，根據為關鍵路徑的任務為優先排程。

Boyer 學者提出一個非演化式方法稱為 RS 演算法(random search algorithm) [15]，此方法以隨機方式產生一條「符合拓樸排序」(topological sort) [18]，所謂「符合拓樸排序」是指排序符合 DAG 的拓樸結構，亦即該順序不違反任務的順序限制。RS 法不斷遞迴去比較新的解和目前的最佳解，如果找到更好的解，就取代目前的最佳解。此法所需的計算時間非常快速，而記憶體の利用也比一般基因演算法少很多。

演化式一般多用基因演算法來求解，如 Shroff 等學者採用 genetic simulated annealing(GSA)，利用標準的 GA 去搜尋但用 SA 模擬退火法[22]來選擇染色體去更新母體。Dhodhi 等人提出 PSGA (problem-space genetic algorithm) 演算法[23]，他們提出 *b-value* 和 *t-value* 的概念，使起始母體(initial Population) 解的品質較佳。然後利用交配與突變等因子來更新母體。

過去基因演算法，都是利用交配運算子(crossover operator)[24] 和突變因子

(mutation operator) 來產生新染色體。在 DHC 的排程指派問題，因為任務有順序限制，交配因子和突變因子可能產生兩個問題：(1)很容易產生不合理的解，(2) 產生品質不佳的解。因此過去的方法在搜尋的效率上仍有改善的空間。



## 第三章 基板重工決策

### 3.1 研究動機

薄膜電晶體顯示器(Thin Film Transistor Liquid Crystal Display, TFT-LCD)產業,生產流程主要可分為三大階段製程,第一階段為陣列(Array)製程與彩色濾光片(Color Filter, 簡 CF)製程,第二階段為組立(Cell)製程,第三階段為模組(Module)製程。Array 製程類似半導體製程,主要製作薄膜電晶體基板(TFT plate),而 CF 製程則是製作彩色濾光片基板(CF plate),之後兩個基板經由 Cell 製程組立壓合,注入液晶後完成 TFT-LCD 基板的製作,再將此 TFT-LCD 基板切割成數個 TFT-LCD 面板(panel),最後經由 Module 製程將驅動晶體電路板、背光模組裝到面板上,即完成所有 TFT-LCD 面板的製作。

無論是 Array 製程所製作的薄膜電晶體基板或是 CF 製程所製作的彩色濾光片基板,通常皆被設計為可切割成數個面板,切割數依顯示器應用產品需求不同而有所不同。在 Array 製程及 CF 製程中,一個原物料基板的投入至最終基板的完成,必須經過多道加工作業(operation),因為製程良率(process yield)的問題,所以一個基板所包含的良好面板個數將會受到影響。假設一個基板可切割成  $n$  個面板,其製程良率假定為  $p$ ,則基板上所包含良好面板的期望個數將等於  $nxp$  個,小於或等於原先的  $n$  個良好面板數量。

一般而言,當有一新製程導入的時候,其製程良率往往不是很高,所以 TFT-LCD 廠會在 Array 製程及 CF 製程階段中,於某些作業與作業中間加設檢測站,並訂定一門檻值(threshold),依據基板上所包含的良好面板個數,來決定此基板是否可繼續加工,高於門檻值的基板可繼續投入下一作業加工;反之,若低於門檻值的基板,由於基板本身的裸基板成本昂貴,所以並非直接報廢,而是經由重工(rework)步驟,將這些基板磨平、去除缺陷後再重新運至投料點重新投料,繼續加工。

如何訂定門檻值,也就是如何訂定重工決策(rework rule),是目前 TFT-LCD 廠經常面臨的一個決策問題。若門檻值定的太低,優點是可以讓很多基板有繼續加工至完成的機會,增加收益的可能,但缺點是若包含良好面板個數很少的基板也能繼續加工時,其面板的單位加工成本就相對提高,且最終的 TFT-LCD 面板成品若來自面板個數較少的基板,也較容易被分類為較低的品質等級;若門檻值定的太高,優點是面板的單位加工成本較低,且最終 TFT-LCD 面板成品容易被分類為較高的品質等級,但缺點是由於基板通過門檻的機會較低,所以能繼續加工至完成的基板個數也相對較少,所以可能會導致最後 TFT-LCD 面板的完成品數量過少,減少收益。因為 TFT-LCD 屬於資本密集的產業,基板成本及其他物料成本皆相當高,所以如何在 Array 及 CF 製程階段中,訂定各檢測站的最佳重工決策,是 TFT-LCD 廠相當重要的問題。

過去在 TFT-LCD 產業已有許多研究專注於其生產過程中排程的問題。但是針對 TFT-LCD 廠的重工門檻值決策問題,在過去的學術文獻上,經過吾人的檢索查閱,尚未發現有相關的研究。現今的 TFT-LCD 廠重工決策的訂定往往都是直覺式

的思考，譬如業界目前大多採用的 1/2 重工決策，也就是經某一加工作業加工後，若基板上所包含的良好面板個數大於總面板數的一半以上，就可投入下一加工作業繼續加工；反之，則逕行重工。有鑑於過去研究的不足，以及工業界直覺性的思考所訂定的重工決策，並未考量到整體績效，且並非站在整個生產系統的觀點來訂定，所以本研究將針對此問題，發展出一較佳的重工決策。

### 3.2 問題描述

根據上述的研究動機，本研究將考量整個 TFT-LCD 廠生產流程中各階段主要製程的特性，針對不同的產品，訂定個別產品在 Array 製程及 CF 製程階段中，各檢測站的最佳重工決策，使 TFT-LCD 廠的獲利最佳化，

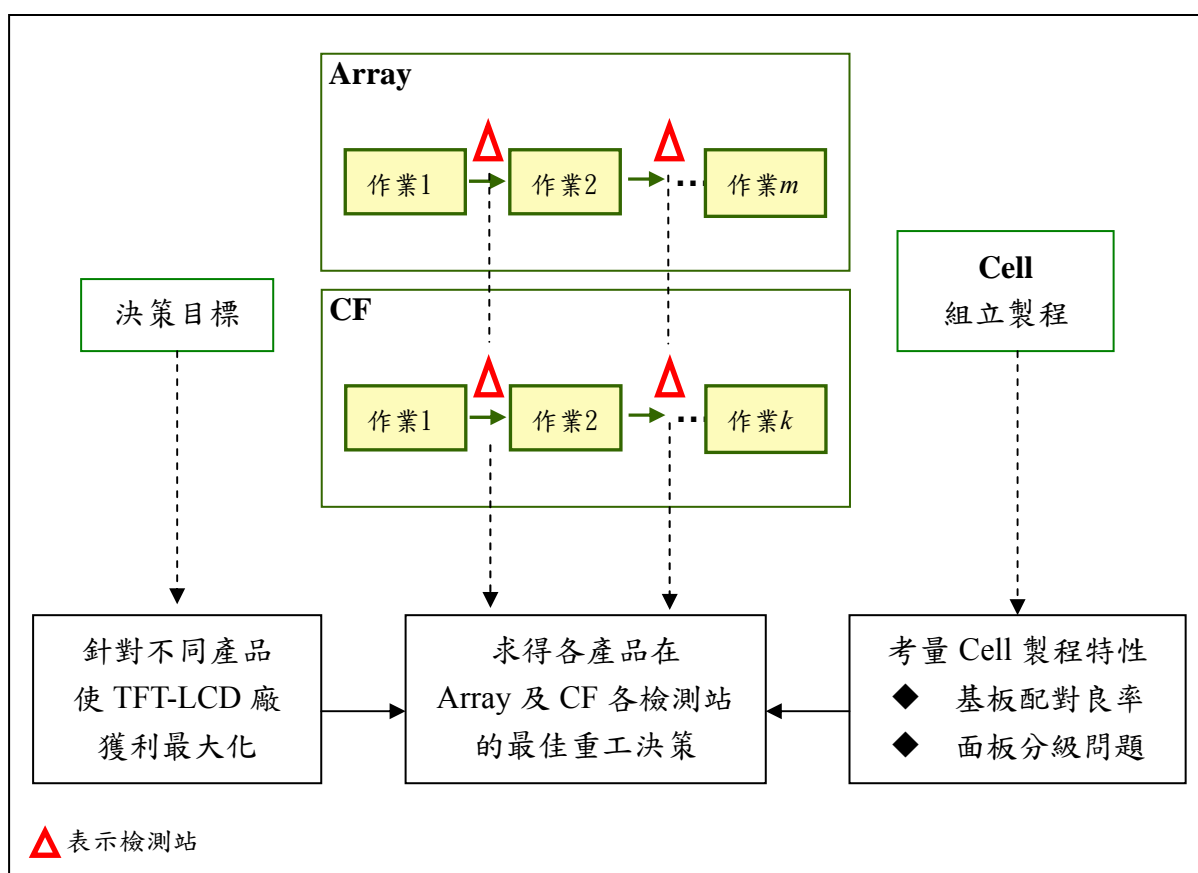


圖 3.1 基板重工問題描述

圖 3.1 為問題描述說明，其中 Cell 製程的主要特性有以下兩點：(1)基板配對良率問題、(2)面板分級比率問題。基板配對良率問題為除了 TFT 基板與 CF 基板本身良率(基板中包含良好面板個數的比率)，會影響 Cell 製程組立後 TFT-LCD 基板的配對良率(mapping yield)外， TFT 基板或是 CF 基板中良好面板的位置，也會影響組立後的 TFT-LCD 基板的配對良率，如圖 3.2 所示，雖然 3.2(a)與 3.2(b)在組立前兩者的 TFT 基板與 CF 基板的良率相同，但因為良好面板的位置不同，導致經由組立製程後的基板配對良率不同。

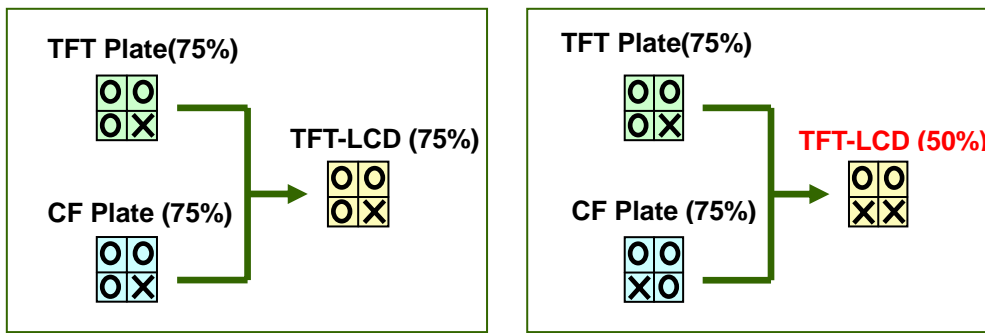


圖 3.2(a)

圖 3.2(b)

面板 = O，表示檢測後為良品 面板=X，表示檢測後為不良品

圖 3.2 TFT 基板與 CF 基板良率影響配對良率說明

面板分級比率問題為面板在 Cell 製程完成後，會依照面板的品質，加以區分等級(grade)，等級越高者，市場售價越高。而組立後 TFT-LCD 基板的配對良率是影響等級區分的重要依據，通常 TFT-LCD 基板的配對良率越高，其切割後的面板被分類為高品質等級的機率越高。譬如說，今假定每個面板可分為 A、B、C 三種等級，則如圖 3.3 所示，面板若來自配對良率 75%的基板，將會有 0.9 的機率被分類為最高等級 A，高於來自配對良率 50%的基板上之面板 (0.85 的機率被分類為最高等級 A)。

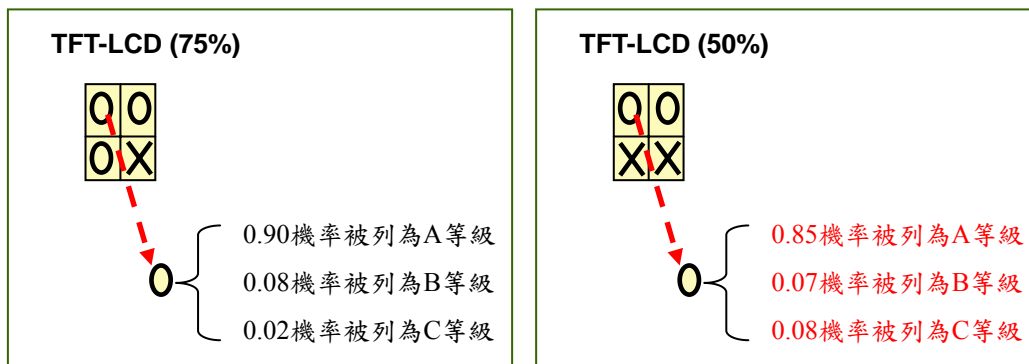


圖 3.3 (a)

圖 3.3 (b)

面板 = O，表示檢測後為良品 面板=X，表示檢測後為不良品

圖 3.3 TFT-LCD 基板等級分類

本研究情境有以下幾點的假設。針對此等情境假設，本論文建構出一數學模式，並利用基因演算法進行運算，針對不同的產品，求得使獲利最佳的檢測站重工決策組合。

- ◆ TFT/CF 基板上良好面板個數分配已知(皆假設為二項式分配)。
- ◆ 假設 Array/CF 製程中皆僅有單一瓶頸，無瓶頸飄移現象，且瓶頸機台為序列(series type)之生產型態機台，。
- ◆ 重工次數上限已知
- ◆ Array/CF 製程皆等待所有重工完成後，蒐集所有 TFT/CF 基板，一次投

入 Cell 製程進行組立作業。

- ◆ Cell 製程的產能大於 Array 製程及 CF 製程的產能，可確保 Array 製程及 CF 製程所製作之基板，皆可進行組立製程作業。
- ◆ 假設 Cell 製程在進行組立作業的過程，是隨機自 Array 製程階段的最終產出中取一片 TFT 基板，同時也隨機自 CF 製程階段的最終產出中取一片 CF 基板進行組立作業，並無事先將 Array 製程與 CF 製程的基板最終產出進行排序或是篩選配對的過程。
- ◆ Cell 製程無良率問題，即進行組立製程時，必能確實完成作業。

### 3.3 解題模式

本研究的研究架構主要可分為兩大模組，模組一(Evaluator)為計算一重工門檻值組合下的績效，利用建構一數學模式，在給定一些產品參數後，計算一組重工門檻值組合下的績效；模組二(Searcher)為搜尋最佳重工組合，目的在於比較眾多重工門檻值組合在績效評估模組中的績效，選取一最佳的重工門檻組合。

#### 3.3.1 模組一

模組一如圖 3.3 所示可再細分為四個子運算模組，分別為：(1)Array/CF 製程運算模組，(2)Cell 組立製程運算模組，(3)面板分級運算模組，(4)績效評估模組。

Array/CF 製程運算模組的目的在於建構出 Array 及 CF 製程階段中，各作業道次的加工及重工決策的運算模式，依據各檢測站所選擇的重工決策，求得在該重工決策組合下，兩製程階段的最終基板期望產出(Array 製程階段產出的 TFT 面板以及 CF 製程階段產出的 CF 面板)，以及兩種基板的最終產出依據基板良率(所包含良好面板個數的比例)的分佈狀態。

Cell 組立製程運算模組建構出 Cell 製程階段中，進行組立製程的運算模式，考量 Cell 製程中進行組立作業的特性建構一數學模式加以分析，將 Array/CF 製程運算模組所求得的基板期望產出數據，經由運算後，求得合併而成的 TFT-LCD 基板的期望產出，以及依據基板配對良率的分佈情形。

面板分級運算模組建構出最終面板成品的等級分級運算模式，利用 Cell 組立製程運算模組所求得的各配對良率之 TFT-LCD 基板期望產出數據，經由本模組運算後，求得各等級面板的產出分佈情形。

績效評估模組是將前三個運算模組所求出的一些影響成本或收益的中間產物及最終產物數據，套入所建構的目標函式中，求得目標函數值，本研究的目的為針對不同產品，使產品利潤最大化。

#### 3.3.2 模組二

模組二最佳重工決策選擇模組用於選取使產品利潤最大化的檢測站重工決

策組合，本研究利用基因演算法(genetic algorithm)建構此模組。亦即經由模組一的運算，我們可求得某一產品在一檢測站重工決策組合下的目標函式值，但若要求出使產品目標函式值最大的檢測站重工決策組合，則必須考量所有可能情形並加以比較，以下先分析本研究問題的求解空間。

基本上若每個基板可包含  $N$  個面板時，一個檢測站的重工決策的門檻值 ( $h$  值) 訂定就有  $N$  種選擇 (1.2.3... $N$ )，但對於每個檢測站  $h$  值的訂定，還須滿足一條條件，那就是一檢測站的  $h$  值必須大於或等於其後檢測站的  $h$  值，因為若今假設某一檢測站的  $h$  值是 2，下一檢測站的  $h$  值是 5，那麼在前一站包含 3 片良好面板的基板會通過，可是該基板到下一站又一定會被重工，既然如此，為了省加工成本，應該當站就重工，所以考量各檢測站的重工決策組合時，必須加入此條件。

由於一檢測站的  $h$  值必須大於或等於其後檢測站的  $h$  值，所以其實可將此問題視為排列組合問題中的重複組合問題，重複組合問題的定義為：從  $n$  種不同物件中，取  $m$  個為一組，每種物件可以重複選取 2 次、3 次、...、至  $m$  次時，稱為  $n$  中取  $m$  的重複組合問題，其組合數公式為  $H_m^n = C_m^{n+m-1}$ ，以下敘述此公式的由來。

假定有  $M$  個檢測站，且依順序為 1、2、...、 $M$ ， $h_i$  為其門檻值， $1 \leq i \leq M$ ，可得知其門檻值限制為  $N \geq h_1 \geq h_2 \geq \dots \geq h_M \geq 1$ ，而此問題即可視為從 1 到  $N$  中取  $M$  個可重複的數字，且不限制重複次數，而取出來的數字組合，會與一組理解唯一對應，例如：若  $N=4$ ， $M=3$ ，門檻值組合為  $(h_1, h_2, h_3)$ ，若抽取的數字組合為 1、2、4，則與一組理解  $(4,2,1)$  唯一對應；若抽取的數字組合為 2、4、2，則與一組理解  $(4,2,2)$  唯一對應；以此類推，可得知此問題即為排列組合中的重複組合問題，所以本問題的合理解空間便可利用重複組合的公式求得  $= H_M^N = C_M^{N+M-1}$ 。

所以當 Array 製程有  $M_1$  個檢測站時，其可能決策組合就有  $H_{M_1}^N = C_{M_1}^{N+M_1-1}$  個；當 CF 製程有  $M_2$  個檢測站時，其可能決策組合就有  $H_{M_2}^N = C_{M_2}^{N+M_2-1}$  個，所以本研究的求解空間數為  $H_{M_1}^N \times H_{M_2}^N = C_{M_1}^{N+M_1-1} \times C_{M_2}^{N+M_2-1}$ ，可發現求解空間會隨著  $N$ 、 $M_1$ 、 $M_2$  的增加而遞增，為避免求解空間過大時，造成求解速度的緩慢，所以本研究採用基因演算法求解，以加速求解速度。

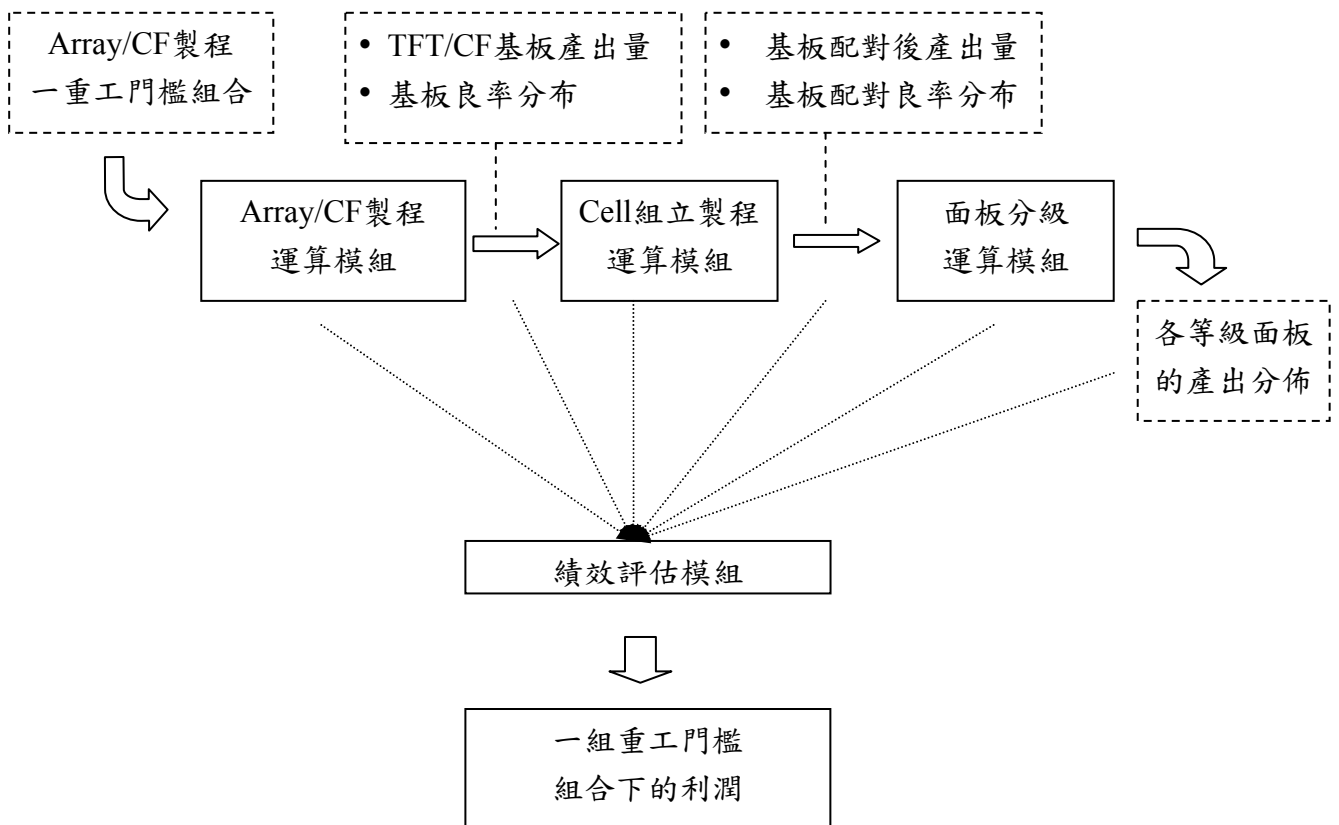


圖 3.4 模組一運算流程

### 3.4 實例驗證

本研究將利用模擬一 TFT-LCD 廠的生產環境，來驗證本研究所提出的方法是否可行，並與兩種決策方法比較，其一是不採用重工決策以，其二是目前工業界大多數直覺性思考所訂定的 1/2 重工決策(包含良好面板個數大於總數一半以上的基板，即可投至下一 stage 繼續進行加工)進行比較。我們用三個案例來比較此三種方法的決策績效。

在案例一，每片基板中所包含的面板數為 6 片，所以若採用 1/2 重工決策，即為經由一 stage 加工後，若一基板包含 3 個以上的好面板，則該基板可投入下一 stage 繼續加工。比較結果如表 3.1，本研究所發展之方法，較其他兩種方法獲利高約 9.30-11.61%。

在案例二，每片基板中所包含的面板數為 10 片，所以若採用 1/2 重工決策，即為經由一 stage 加工後，若一基板包含 5 個以上的好面板，則該基板可投入下一 Stage 繼續加工。比較結果如表 3.2，本研究所發展之方法，較其他兩種方法獲利高約 6.48-7.63%。

此案例的每片基板中所包含的面板數為 15 片，所以若採用 1/2 重工決策，即為經由一 stage 加工後，若一基板包含 8 個以上的好面板，則該基板可投入下一 Stage 繼續加工。比較結果如表 3.3，本研究所發展之方法，較其他兩種方法獲利高約 3.36%-3.82%。

經由以上分析比較，可發現本研究所提出的最佳重工決策方法，與無採用重工決策以及目前大部分工業界所採用的 1/2 重工決策比較，無論在各等級面板的期望產出，以及目標函式值的比較，皆有相當明顯的改善，證明本研究以系統觀的角度所求得的最佳重工決策，相較於以往採直覺性思考所訂定的重工決策，本研究確實有其價值性。

表 3.1 基板重工決策案例一之績效比較

	Z(片)	P(片)	N(片)	目標函數值(元)
(1)無重工決策	34,200	2,317	1,452	\$314,915,442
(2) 1/2 重工決策( $h=3$ )	34,336	2,319	1,452	\$321,556,164
(3) 本研究所求得之最佳重工決策	37,062	2,491	1,467	\$351,466,288
(4) 本研究改善無重工決策的比率 $=((3)-(1))/(1)$	8.37%	7.51%	1.03%	11.61%
(5)本研究改善 1/2 重工決策的比率 $=((3)-(2))/(2)$	7.94%	7.42%	1.03%	9.30%

表 3.2 基板重工決策案例二之績效比較

	Z(片)	P(片)	N(片)	目標函數值(元)
(1)無重工決策	55,225	4,080	2,897	\$223,299,419
(2) 1/2 重工決策( $h=5$ )	55,266	4,080	2,891	\$225,701,499
(3) 本研究所求得之最佳重工決策	57,251	4,178	2,929	\$240,335,878
(4) 本研究改善無重工決策的比率 $=((3)-(1))/(1)$	3.67%	2.40%	1.10%	7.63%
(5)本研究改善 1/2 重工決策的比率 $=((3)-(2))/(2)$	3.59%	2.40%	1.31%	6.48%

表 3.3 基板重工決策案例三之績效比較

	Z(片)	P(片)	N(片)	目標函數值(元)
(1)無重工決策	86,071	4,922	3,698	\$256,348,671
(2) 1/2 重工決策( $h=8$ )	86,049	4,911	3,689	\$257,472,097
(3) 本研究所求得之最佳重工決策	87,769	4,919	3,671	\$266,128,646
(4) 本研究改善無重工決策的比率 = $((3)-(1))/(1)$	1.97%	-0.06%	-0.73%	3.82%
(5)本研究改善 1/2 重工決策的比率 = $((3)-(2))/(2)$	2.00%	0.16%	-0.49%	3.36%



## 第四章 基板重組派工決策

### 4.1 研究問題

在組立製程中，TFT 基板和 CF 基板配對的結果會影響製程的良率。在組立製程中，一塊 TFT-LCD 基板(plate)可被分割成很多面板(panel)；若要得到一良品 TFT-LCD 面板，則在進行壓合時的 TFT 和 CF 面板都必須為良品才行；若 TFT 或 CF 面板其中有一為不良品，則壓合作業後的 TFT-LCD 面板便為不良品。如圖 4.1 所示，例 1 中 TFT 基板和 CF 基板的個別良率均為 75%，但在進行壓合作業後，因為配對關係，僅得到良率為 50%的 TFT-LCD 基板，而在例 2 中，TFT 基板和 CF 基板的個別良率同樣為 75%，因為較好的配對，經壓合後可得到良率為 75%的 TFT-LCD 基板。本研究稱此配對後的良率為配對良率(mapping yield)。

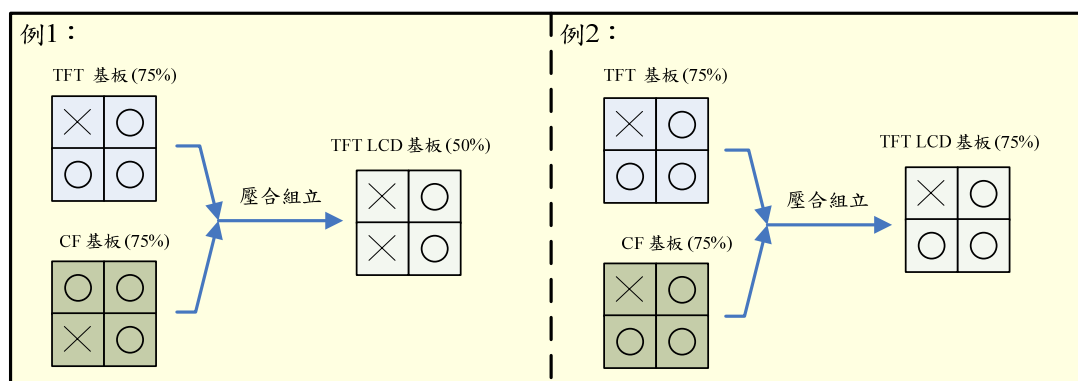


圖 4.1 TFT 基板和 CF 基板的配對良率

在生產現場，基板是以卡匣(cassette)為單位進行承載，通常一個卡匣可以承載約 10-20 片基板。基板製造的良率具有隨機特性，在一卡匣內，每一片基板的良率未必相同，即使良率相同，良品面板位置的分佈也未必一致。為了提高配對良率，工廠通常會將多個 CF 和 TFT 卡匣內的基板整合配對，一配對成組的 CF 和 TFT 基板互稱為配對基板(mapping plate)。假設一個卡匣可以承載  $n$  個基板，當 TFT 和 CF 各有  $N$  個卡匣時，則基板配對方式共有  $(N \times n)!$  種組合。在這些組合中，如何找到最佳良率的配對組合，我們稱之為基板配對決策 (plate mapping decision)，在過去的研究中，已有最佳演算法能找到最大良率之基板配對決策。

基板配對決策之後，一 CF 卡匣內的基板所對應的  $n$  個 TFT 基板可能雜散在不同的 TFT 卡匣內，為了方便 Cell 製程的組立作業，工廠通常會先進行基板抽換及重置作業(pick-and-replace)，亦即將一 CF 卡匣所配對的 TFT 基板從原卡匣內全部抽出，置放在一個新的 TFT 卡匣內，以便壓合組立，此新的 TFT 卡匣稱為目標卡匣(target cassette)。如圖 4.2 所示，CF 卡匣  $X$  內有三個基板( $x_1, x_2, x_3$ )，其配對基板( $q_1, r_2, s_3$ )分別存放在 TFT 卡匣  $Q, R, S$  內。抽換作業將 TFT 基板  $q_1, r_2,$

$s_3$  從其原卡匣抽出，置放在新的 TFT 卡匣  $Y$  內，使卡匣  $X$  內的所有基板，其配對基板均在卡匣  $Y$  內，抽換作業完成後的卡匣  $Y$  為目標卡匣。

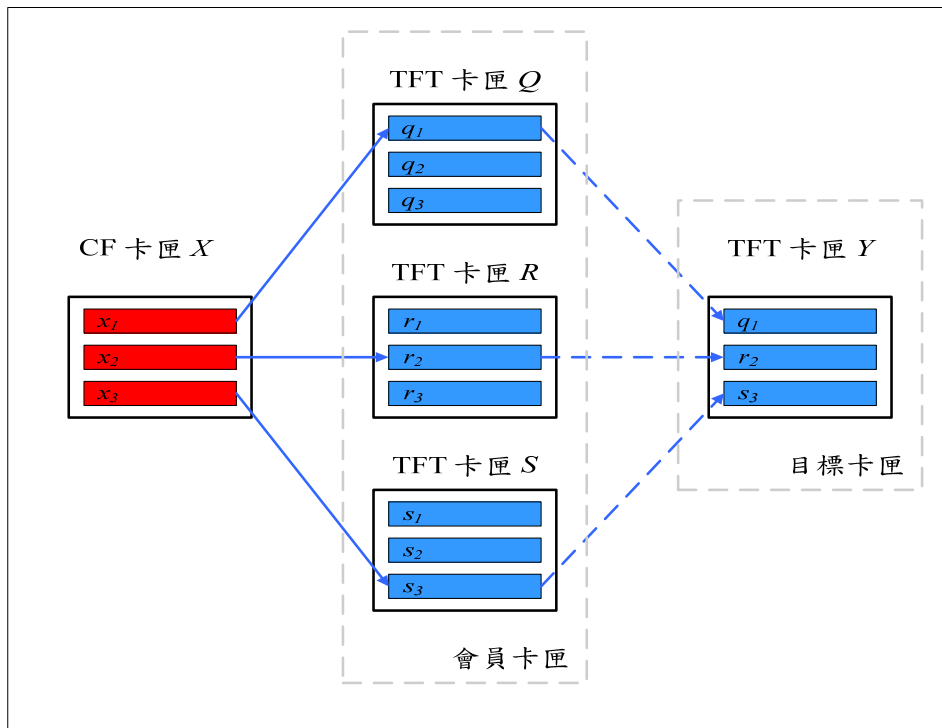


圖 4.2 基板抽換作業

進行抽換作業的設備稱為排序機(sorter)。如圖 4.3 所示，一個典型排序機的組成包括一個機器手臂(robot)，一個輸出埠(output port)，數個輸入埠(input port)。進行抽換作業時，輸出埠置放一空卡匣，機器手臂的功能是從輸入埠所置放的卡匣中抽出 TFT 基板，將其放在輸出埠的卡匣，以形成目標卡匣。

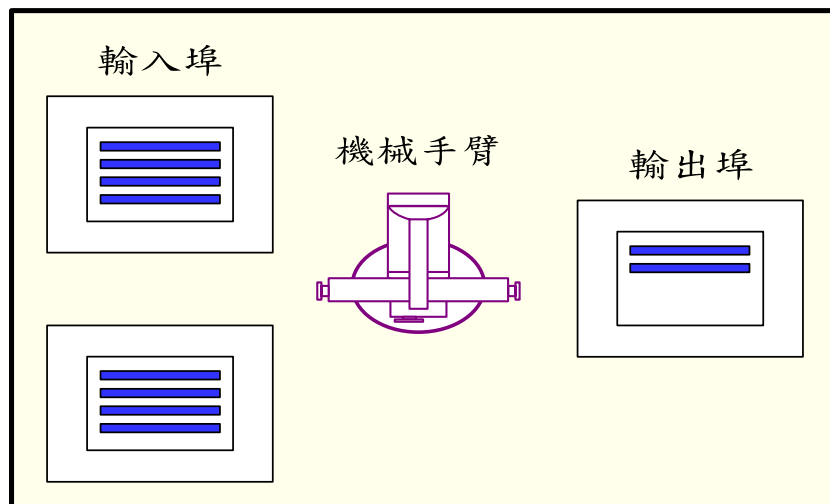


圖 4.3 排序機設備

經基板配對決策後，一 CF 卡匣內的基板所對應的  $n$  個 TFT 基板可能會分散在多個 TFT 卡匣中，這些 TFT 卡匣稱為該 CF 卡匣的會員卡匣(member cassette)，本研究利用卡匣配對矩陣(cassette mapping matrix, 簡稱配對矩陣)整合各 CF 卡匣與其會員卡匣間的卡匣對應關係(cassette-to-cassette relationship)，配對矩陣可用  $M_{N*N} = [m_{ij}]$ ,  $1 \leq i \leq N, 1 \leq j \leq N$  表示，若 TFT 卡匣  $j$  為 CF 卡匣  $i$  之會員卡匣，則  $m_{ij}=1$ ，反之則  $m_{ij}=0$ 。圖 4.4 為 4 個 CF 卡匣和 4 個 TFT 卡匣進行基板配對決策後所得的卡匣配對矩陣  $M_{4*4}$ ，如圖所示，CF 2 卡匣的配對基板分散在 TFT 1, TFT 2 及 TFT 3 卡匣中，因此 TFT1, TFT 2 及 TFT 3 三個卡匣為目標卡匣 CF2 的會員卡匣，抽換作業需將配對基板從三個會員卡匣中抽出，放置在一新的 TFT 卡匣內，才能完成目標卡匣的產出。

	<b>TFT 1</b>	<b>TFT 2</b>	<b>TFT 3</b>	<b>TFT 4</b>
<b>CF 3</b>	1	0	0	0
<b>CF 4</b>	0	1	0	1
<b>CF 2</b>	1	1	1	0
<b>CF 1</b>	1	1	1	1

圖 4.4 配對矩陣

在排序機的派工過程中，會衍生出兩個問題，分別是目標卡匣的產出順序決策(output sequence decision)及會員卡匣上下排序機的順序決策(input sequence decision)，這兩個問題為排序機派工的主要決策；決策一為目標卡匣的產出順序(output sequence, 簡稱產出順序)，如圖 4.5 中所示，產出順序可為  $C1 \rightarrow C2 \rightarrow C3 \rightarrow C4$ ，或  $C3 \rightarrow C4 \rightarrow C2 \rightarrow C1$  等，共  $4!$  種情形；決策二為會員卡匣上下排序機的順序(input sequence, 簡稱投入順序)，如圖 4.5 中所示，以 CF2 目標卡匣為例，其會員卡匣為 TFT1, TFT2, TFT3，在排序機只有一個輸入埠的情形下，投入順序為  $T1 \rightarrow T2 \rightarrow T3$  或  $T1 \rightarrow T3 \rightarrow T2$  等，共  $3!$  種情形。

組立製程為流線型生產(flow line)，排序機下游為組立壓合機台(assembly machine)，作業內容是將配對好的 TFT 基板和 CF 基板進行壓合。其生產特性為，在不缺料的情形下，每隔固定時間  $T$  可以產出一個壓合卡匣，反之在缺料的情形下，壓合機的產能則會閒置；因此壓合機對排序機的要求是每隔固定時間  $T$  需產出一目標卡匣，若目標卡匣的產出時間變異過大時，壓合機的產能可能會閒置；因此本研究著焦於排序機應該如何派工，以最大化壓合機台利用率，以期能提高 TFT-LCD 廠的最終產出。

綜上所述，本研究目的是以卡匣配對矩陣作為輸入，發展一演算法來進行排序機派工決策，妥善規劃產出順序和投入順序，以最大化下游組立壓合機台利用率。

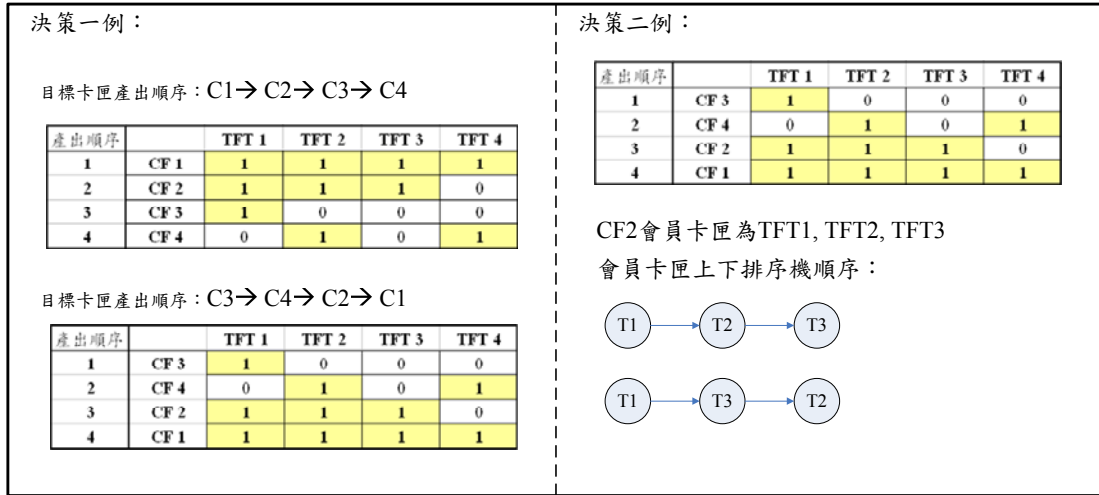


圖 4.5 排序機派工決策

## 4.2 解題方法

本研究主要的決策問題是規劃投入順序，因此本節重點在將投入順序決策建構成動態規劃模型(Dynamic Programming Model)，其內容包含有二，分別是將投入順序決策問題建構成完整的網路，再依動態規劃特有的系統化方法找尋最佳投入順序。欲將投入順序決策過程描述成動態規劃網路，其過程主要可分為兩個部分（參閱圖 4.6）。

第一部分是基本的網路模型的建立，此基本架構包含了階段(stage)，狀態(state)，批次(batch)，而基本架構的建立透過 Stage Modeling, State Modeling 和 Batch Modeling 三個模組來完成；第二部分為加強網路結構，利用 State Deployment 模組將原本不完整的狀態，重行部署成為一個或多個完整的狀態，目的是為了使網路架構更加完整。

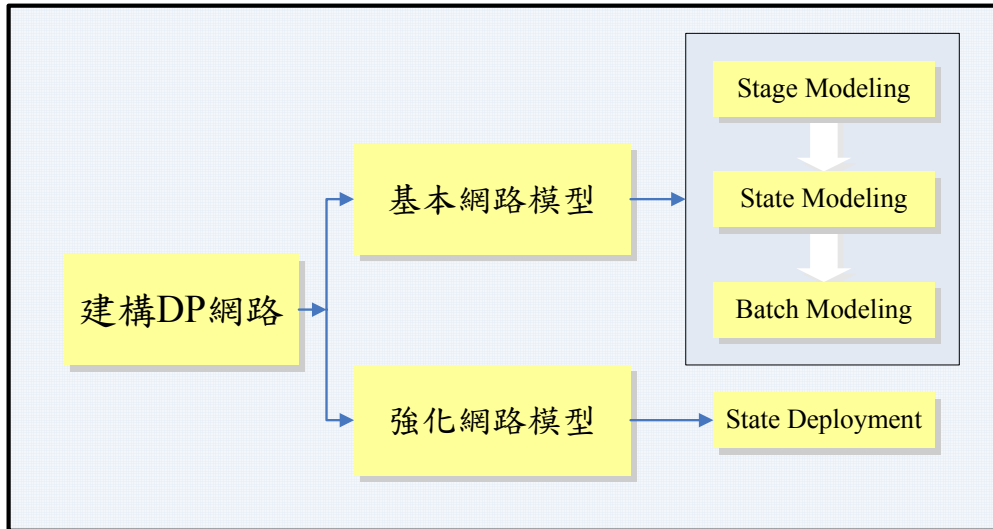


圖 4.6 建構 DP 網路流程

以下吾人依序對 Stage Modeling, State Modeling, Batch Modeling 和 State Deployment 四模組進行介紹。Stage Modeling 是建立基本網路架構的第一步驟，其目的是建立出網路中所有的階段  $Stage\_i$ ,  $1 \leq i \leq N$ ，在本研究方法中，階段  $i$  (stage  $i$ ) 是指能產出目標卡匣  $K_i$  的所有投入順序 (state) 所形成的集合，因此階段  $i$  可表示成  $Stage\_i = \{seq_{i,j} | 1 \leq j \leq s(i)\}$ 。

State Modeling 為建構基本網路的第二步驟，其目的是建立出網路中的所有狀態 (state)  $seq_{i,j}$ ,  $1 \leq i \leq N$ ,  $1 \leq j \leq s(i)$ ，在本研究中，在階段  $i$  中的狀態 (state) 是指能產生目標卡匣  $K_i$  的投入順序。

Batch Modeling 為建構基本網路的第三個步驟，目的是建立出網路中的所有狀態內所包含的批次  $B_{(i,j),k}$ ,  $1 \leq i \leq N$ ,  $1 \leq j \leq s(i)$ ,  $1 \leq k \leq b(i)$ ；Batch Modeling 主要透過重複不放回隨機選取的程序 (Random Select without Replacement Procedure) 來產生一個狀態內的所有批次序列。

經過 Stage Modeling, State Modeling 及 Batch Modeling 三個步驟，可得到基本的網路模型 (參閱圖 4.7)，同時可歸納出狀態具有三種不同的組態，如圖 4.8 所示，分別為：(1) 組態一：狀態均由完整的批次所組成，(2) 組態二：狀態由至少兩個批次組成，且最後一個批次為不完整批次，(3) 組態三：狀態只由一不完整的批次所組成；

在組態二及組態三中，狀態內的批序列均包含一個不完整的批次，在本研究方法中，帶有不完整批次的狀態 (state with unfilled batch) 意味著一個無法提供完整資訊的決策組合，因此本研究方法透過 State Deployment 模組來重組組態二及組態三的狀態，使原本屬於組態二或組態三的狀態能盡可能重組為組態一的狀態，

經過 State Deployment 的步驟之後，網路的架構更加完整，其產出結果如圖

4.9。吾人繼而發展動態規劃法(dynamic programming)求解此基板重組與排序問題。

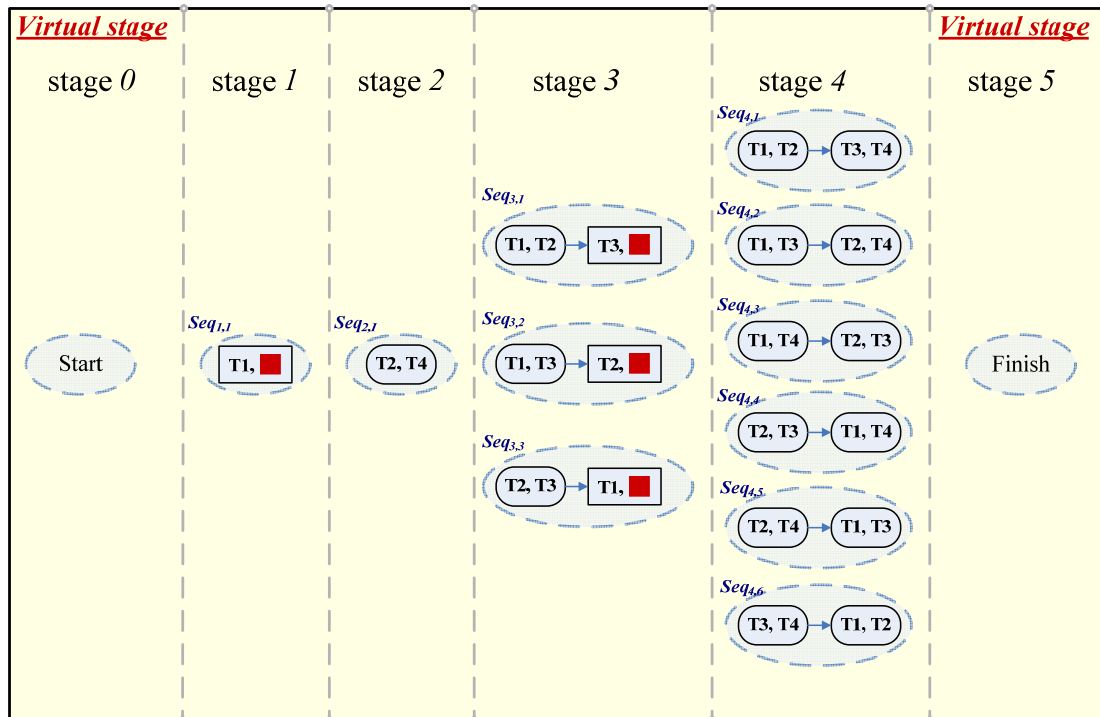


圖 4.7 基本網路架構

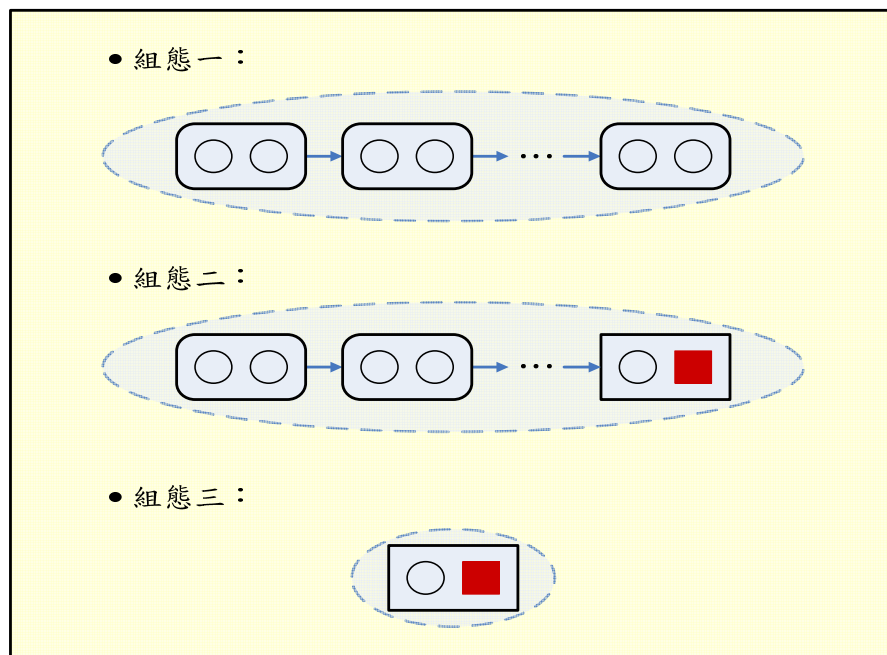


圖 4.8 三種狀態組態

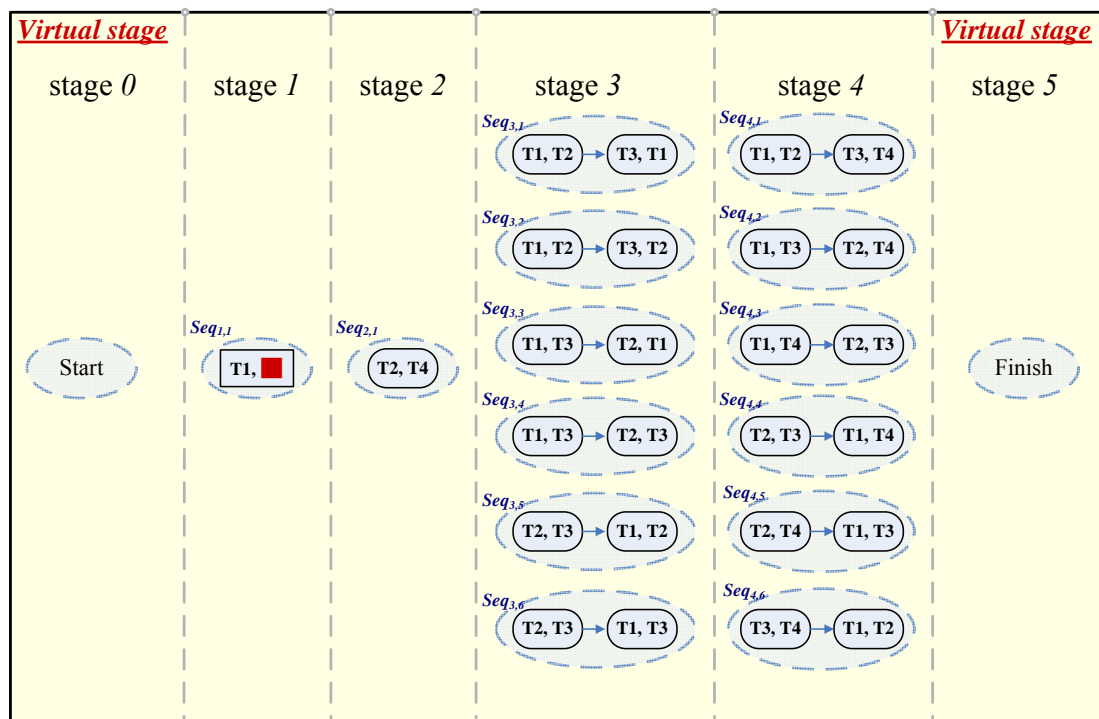


圖 4.9 完整的基本網路架構

### 4.3 實例驗證

本研究以90個工廠實際的案例來說明本研究所發展的方法及成果。本實例驗證採用張東華[24]基板配對決策所得到的結果，TFT及CF卡匣總數以 $N$ 表示，每個卡匣的基板個數是 $n$ ，每個基板的面板個數以 $panel$ 表示。本模擬比較的基準為目前業界使用的隨機完成順序及王君豪提出的基因啟發式解法。績效指標是下由的產能利用率。

本案例在驗證執行程序上，是先採用王君豪[25]所發展的基因啟發式解法求解15次，可得到15個較佳的產出順序，以這15個目標卡匣組產出順序為輸入資料，再以本研究方法求解投入順序決策問題，15個解中績效最高的者則為本演算法之最佳解。

由驗證的結果可得知，本研究方法在現今TFT-LCD生產情境中具有十分顯著的效益，壁排序機裝卸卡匣的時間和下游組立機台作業時間比例為0.05的情形下，本研究方法所得到的績效比王氏利用基因啟發式解法有更佳的表現，改進的績效如表4.1所示，最大有8.3%的改進幅度。

表 4.1 排序機各種排序方法的績效比較

生產情境	輸入埠數 $p$	隨機解	王氏	本研究	改進績效
$N=10$ $n=20$ $panel=6$	1	89.9%	96.2%	100.0%	3.8%
	2	91.2%	95.9%	100.0%	4.1%
	3	90.0%	95.5%	100.0%	4.5%
	4	89.7%	95.1%	100.0%	4.9%
	5	88.7%	94.5%	100.0%	5.5%
$N=10$ $n=20$ $panel=12$	1	90.9%	94.5%	100.0%	5.5%
	2	91.2%	95.0%	100.0%	5.0%
	3	89.7%	94.6%	100.0%	5.4%
	4	89.5%	94.1%	100.0%	5.9%
	5	90.1%	93.6%	100.0%	6.4%
$N=10$ $n=20$ $panel=30$	1	88.5%	92.8%	100.0%	7.2%
	2	89.1%	92.6%	100.0%	7.4%
	3	88.6%	92.7%	100.0%	7.3%
	4	86.3%	92.7%	100.0%	7.3%
	5	85.6%	91.7%	100.0%	8.3%
$N=15$ $n=20$ $panel=6$	1	93.4%	98.3%	100.0%	1.7%
	2	92.6%	98.2%	100.0%	1.8%
	3	93.6%	98.1%	100.0%	1.9%
	4	94.0%	98.0%	100.0%	2.0%
	5	92.6%	97.9%	100.0%	2.1%
$N=15$ $n=20$ $panel=12$	1	92.6%	95.1%	100.0%	4.9%
	2	92.5%	95.3%	100.0%	4.7%
	3	92.7%	95.5%	100.0%	4.5%
	4	93.1%	95.8%	100.0%	4.2%
	5	92.9%	95.5%	100.0%	4.5%
$N=15$ $n=20$ $panel=30$	1	93.1%	94.4%	100.0%	5.6%
	2	92.5%	94.2%	100.0%	5.8%
	3	91.6%	94.2%	100.0%	5.8%
	4	90.9%	94.3%	100.0%	5.7%
	5	91.4%	94.4%	100.0%	5.6%
$N=20$ $n=20$ $panel=6$	1	94.9%	98.5%	100.0%	1.5%
	2	95.1%	98.5%	100.0%	1.5%
	3	95.7%	98.4%	100.0%	1.6%
	4	95.5%	98.3%	100.0%	1.7%



	5	96.2%	98.3%	100.0%	1.7%
<i>N=20</i> <i>n=20</i> <i>panel=12</i>	1	94.8%	97.9%	100.0%	2.1%
	2	94.6%	98.2%	100.0%	1.8%
	3	95.1%	98.1%	100.0%	1.9%
	4	94.7%	98.0%	100.0%	2.0%
	5	94.7%	97.9%	100.0%	2.1%
<i>N=20</i> <i>n=20</i> <i>panel=30</i>	1	92.8%	95.2%	95.2%	0.0%
	2	93.1%	96.2%	98.2%	2.0%
	3	93.1%	96.9%	99.1%	2.2%
	4	93.3%	97.7%	99.5%	1.8%
	5	93.5%	97.6%	100.0%	2.4%

## 第五章 專案排程

產品組合和投料的決策問題，基本上是一個空間搜尋問題，只要基板重工決策、基板配對決策可以求解，我們就可求出最佳的投料組合和數量決策。為發展快速有效之搜尋演算法，本計畫在研究此問題，意外想到一個異質專案排程的解法，並發展出一個特殊的搜尋演算法，本章特別介紹此專案排程問題和解法。

### 5.1 研究問題

有限資源的專案排程(project scheduling)問題可能會非常複雜，因為具有下列二個特性。第一、任務(task)之間具有相依關 (dependent relationship)，此等相依關係通常可用一個有向非循環圖來描述 (directed acyclic graph, 簡稱 DAG)，如圖 5.1 所示，該專案具有 6 個任務 (task)，任務間有執行順序上的限制，譬如說，任務  $t_2$  需要任務  $t_1$  與任務  $t_6$  都執行完畢，才能執行。第二、任務的執行具有共用資源的特性，如圖 5.2 所示，該專案所有的資源是 3 部機器，每個任務都可用任一部機器加工。

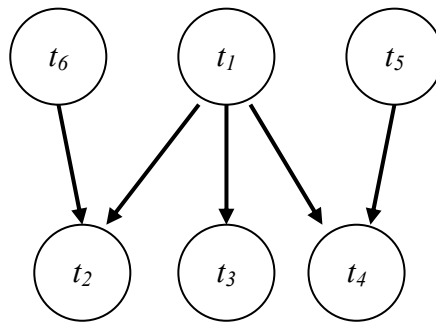


圖 5.1 DAG 任務

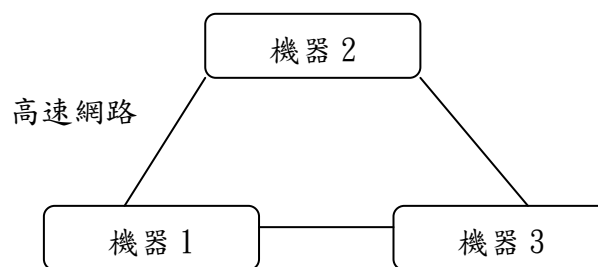


圖 5.2 3 個機器的異質系統

此種排程問題可分成兩個子問題：排序(sequencing)問題、指派問題(matching)。排序問題是決定任務執行的先後順序，指派問題是將任務指派給機器。這類排程問題的複雜度(complexity)很高，因為  $t$  個任務排序最多有  $t!$  組合，

若該專案有  $m$  個資源，則有  $m'$  個指派組合；結合排序與指派的決策，一專案的排程方案最多有  $t! \cdot m'$  種，解空間非常龐大。

上述專案排程問題可能發生在各種領域，譬如研發專案排程、生產系統排程、計算機系統排程。在計算機領域，一個典型的案例是異質運算系統(distributed heterogeneous computing; DHC)的排程。所謂 DHC 系統是指由多部異質機器(具有不同運算能力的計算機)、高速網路、通訊協定、作業系統，和程式環境所組成的一強大運算系統。此種計算系統執行一個程式時，先將程式切割成多個子程式，排定子程式的執行順序，並指派各子程式給不同機器處理，因為子程式間有相依關係，而且共有資源，因此 DHC 排程問題具有上述專案排程問題的特性，以下本研究稱此專案排程問題為 DHC 排程問題。

在 DHC 系統中，應用程式通常分割為許多任務(task)，這些任務的關係可以被描述為有向非循環圖(DAG; directed acyclic graph)，表示為  $G = (T, <, E)$ ， $T = \{t_i, i = 1, \dots, n\}$  代表所有任務的集合， $t_i$  為第  $i$  個任務， $n$  為任務的個數。 $<$  代表任兩節點的順序關係，例如：任兩個任務  $t_i, t_k \in T$ ，當  $t_i < t_k$  意指， $t_i$  為  $t_k$  的先行者(predecessor)， $t_k$  為  $t_i$  的後續者(successor)。所以  $t_i$  執行完畢後， $t_k$  才可以被執行。 $E$  代表整個圖形所有的邊(edge)或弧(arc)的集合，而每個邊或弧都附加傳輸資料量  $D_{i,k}$ ，代表從任務  $t_i$  到任務  $t_k$  所需的傳輸資料量(以位元 byte 為單位)。圖 5.3 是一個 DAG 的釋例，例如  $(1 < 3)$  代表任務  $t_1$  與任務  $t_3$  之間相連，且任務  $t_1$  執行完成之後，才可以執行任務  $t_3$ ，而任務  $t_1$  到任務  $t_3$  的傳輸資料量  $D_{1,3}$  為 5。該圖中也明顯說明任兩任務並不需要完全連接，例如任務  $t_5$  為一個獨立的任務。

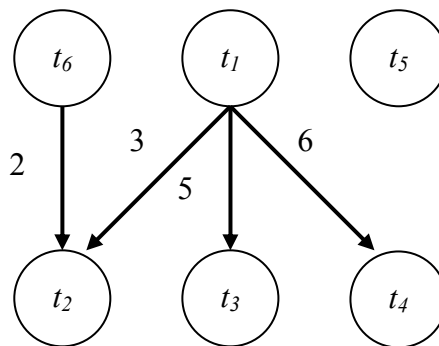


圖 5.3 有傳輸時間的 DAG 任務

DHC 運算系統是由一組獨立且不同型的機器由高速網路所整合的運算系統， $M$  為所有機器的集合，描述為  $M = \{m_j, j = 1, \dots, m\}$ ， $m$  為機器的個數， $m_j$  為第  $j$  個機器。在 DHC 系統中，不同機器之間的頻寬 (bandwidth) 跟據其網路 (network) 而有所不同，通常描述為  $m \times m$  矩陣  $R_{m \times m}$ 。任兩任務之間的傳輸成本 (communication time) 如下所示：

$$CommTime(t_i, t_k) = \frac{D_{i,k}}{R[H(i), H(k)]}$$

$$\text{if } H(i) = H(k), \text{ then } CommTime(t_i, t_k) = 0$$

上式中， $H(i)$ 為任務  $i$  被指派的機器， $R[H(i), H(k)]$ 為頻寬， $D_{i,k}$ 為從任務  $t_i$  到任務  $t_k$  所需的傳輸資料量，當任兩任務  $t_i, t_k$  指派給同一個機器時，傳輸成本為 0，代表並不需要傳輸資料。為了簡化問題，我們假設任兩機器的頻寬皆為 1.0，因此任兩任務之間的傳輸成本相等於其資料傳輸量，指派給不同機器不會影響傳輸時間。

因為是異質機器，同一任務在不同機器的期望執行時間(expected execution time)會不同。再者任務不能再拆解，亦即設每一個任務只能指派給一部機器執行。每一個任務根據其所執行機器可以精確的估計其期望執行時間，期望執行時間可以一個  $n \times m$  的矩陣  $E$  表達，其中  $E_{ij}$  代表任務  $i$  在機器  $j$  的期望執行時間

本排程部問題績效指標是總完工時間(makespan of the program)，目標函數如下所示：

$$T = \max (F_1, \dots, F_m)$$

上式中， $F_j$  為機器  $j$  的全部完成時間(makespan)， $j=1, \dots, m$ 。完成時間包括執行時間(processing time)和因有優先關係而產生的等待時間(waiting time)，其中等待時間包括了機器實際閒置時間(idle time)和資料傳輸時間(communication time)。

## 5.2 指派的解法

本研究求解 DHC 的排程與指派問題，也是採用逐步求解(serial methodology)的解題架構。亦即先求解指派問題，再求解排序問題。如前所述，指派的決策是一績效評估機制，排序問題是一最佳解的搜尋機制。本研究是採用 Boyer 所提出的 DS 法則[30]當指派的決策，本章擬說明此 DS 法則，以下先說明符號和演算法流程，其次舉例說明此演算法。

### 指標與變數:

任務指標:  $i$  ( $1 \leq i \leq n$ )

機器指標:  $j$  ( $1 \leq j \leq m$ )

$F_j$ : 就目前已經指派在機器  $j$  上的任務，預估機器  $j$  的總完成時間

$F_{ij} = T_j$ : 若將新任務  $i$  指派在機器  $j$  上，預估機器  $j$  的總完成時間

$T = \max(F_1, \dots, F_m)$ : 指派新任務後，預估程式總完工時間

$\Delta_{ij} = F_{ij} - F_j$ , 若新任務  $i$  指派給機器  $j$ ，機器  $j$  預估完成時間的增量

茲將 Boyer 文獻中 DS 法則的演算法敘述於下。此演算法輸入參數包括一給

定的任務順序(ordered list of tasks)、計算機器執行各任務的時間(machine list)、任務的 DAG 圖。輸出結果是將每個任務指派給機器(matching decision)，並計算出總完工時間(makespan)。首先初始化  $T=0$ ，目前排程解為空集合。依序把任務預估計算每個機器  $j$  的  $\Delta_{ij}$  和  $T_j$ ，根據是否有無超過目前的 makespan 為判定。大於目前的 makespan，把任務指派給最小  $T_j$  的機器  $y$ ，更新目前的 makespan 和  $F_j$ ；小於等於目前的 makespan，把任務指派給最小  $\Delta_{ij}$  的機器  $x$ ，更新  $F_j$ 。直到全部任務指派完停止。流程如下：

Procedure DS

Input: ordered Task\_list, Machine\_list, DAG

Initialization:  $T=0$ ,  $Solution\_schedule = \{\text{empty}\}$

**While**(Task\_list not empty)

    Remove task  $i$  from head of Task\_List

    For each machine  $j$  in Machine\_List

        Compute  $\Delta_{ij}, T_j$

$x = \arg \min_j (\Delta_{ij})$

$y = \arg \min_j (T_j)$

**if** ( $T_y > T$ ) **then**,

$T = T_y$ , and  $F_y = T_y$

$Machine\_Assign(i) = y$

**else**

$F_x = T_x$

$Machine\_Assign(i) = x$

    Endfor

**Endwhile**;

Return  $Solution\_Schedule = \{Assign(i), \text{for each task in Task\_List}\}$

## 5.3 排序的解法

如前所述，本研究求解 DHC 的排程與指派問題，是採用逐步求解(serial methodology)的解題架構。指派的決策是一績效評估機制，排序問題是一最佳解的搜尋機制。本研究提出一以共識因子為基礎的基因演算法為搜尋機制。此基因演算法分為六個模組：染色體表達法、適應函數、初始母體產生法、產生新染色體的共識因子、選擇染色體形成新母體的方法、終止條件。茲將各模組分別敘述如下：

### 5.3.1 染色體與適應函數

每條染色體代表為任務執行順序(task sequencing)，可表示為  $X = \{x_1, x_2, \dots, x_n\}$ ， $X$  代表一條染色體， $n$  代表所有任務的個數。染色體基因值(gene value)表示為執行的任務  $t_i$ ，染色體的索引(index)代表任務  $t_i$  的先後。而一個有效的染色體不可以違反任務之間的執行順序限制。例如在 DAG 中，6 個任務可能產生如表 5.1 的染色體  $t_1 \rightarrow t_6 \rightarrow t_5 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4$ ，其中  $t_6$  在  $t_2$  之前，因此  $t_6$  的 index 小於  $t_2$  的 index。

表 5.1 異質專案排程之染色體表達與解讀

索引(index)	1	2	3	4	5	6
基因(gene)	$t_1$	$t_6$	$t_5$	$t_2$	$t_4$	$t_3$

適應函數是用於表達各染色體解的品質，以決定其下一世代的存活率。解的品質越好，染色體存活率越高。產生的染色體經由派工方法 DS 計算出的程式總完工時間  $T$ ，作為適應值(fitness value)。最小化目標式，其適應函數如下：

$$\text{適應函數: } T = \max (F_1, \dots, F_m)$$

### 5.3.2 初始母體產生法

因為任務之間有執行順序限制，染色體基因不可以隨機的指派。為了產生好的染色體，我們採用 PSGA 的優點，根據問題特徵去求得  $b$ -level 和  $t$ -level 等數值，以產生任務的優先值(task priority value)，敘述如下。

(A)計算  $b$ -level:

$b$ -level 為任務  $t_i$  到最後一個任務的最長關鍵路徑長度，因此每個任務  $t_i$  的  $b$ -level 被 DAG 的所限制，因此越前面的任務所需的旅行路徑越長，其  $b$ -level 通常較高。每個任務  $t_i$  的  $b$ -level 計算為在各機器平均執行時間  $AvgE(t_i)$  加上與最長 (longest) 的子任務  $t_j$  (child task) 的  $b$ -level 和傳輸時間  $CommTime(t_i, t_j)$ 。根據 DAG 由下往上計算每個任務  $t_i$  的  $b$ -level，若任務  $t_i$  沒有子任務  $t_j$  其  $b$ -level 為其平均執行時間  $AvgE(t_i)$ 。 $b$ -level 的計算流程如下：

**Procedure  $b$ -levels( ):**

Construct a reverse topological order list (*RevTopOrdList*) of tasks.

**for** each task  $t_i$  in the *RevTopOrdList* **do begin**

$max = 0$ ;

**for** each child task  $t_j$  of task  $t_i$  **do begin**

**if** (  $CommTime(t_i, t_j) + b\text{-levels}(t_j) > max$  ) **then**

$max = CommTime(t_i, t_j) + b\text{-levels}(t_j)$

**endif**;

**endfor**;

$b\text{-levels}(t_i) = AvgE(t_i) + max$

**endfor**;

(B)計算  $t$ -level:

另外我們還要計算每個任務的  $t$ -level， $t\text{-level}(t_i)$  為 entry node 到達任務  $t_i$  的最長距離，因此可以判斷任務  $t_i$  的最早開始時間。 $t\text{-level}(t_i)$  為父任務(parent task) 的  $t\text{-level}(t_k)$  與父任務的傳輸時間  $CommTime(t_k, t_i)$  加上父任務的執行時間  $AvgE(t_k)$ 。

根據 DAG 由上往下計算每個任務  $t_i$  的  $t$ -level，若任務沒有父任務的話，其  $t$ -level 為 0。  $t$ -level 的計算流程如下：

**Procedure  $t$ -levels( ):**

```

Construct a topological order list (TopOrdList) of tasks.
for each task  $t_i$  in the TopOrdList do begin
     $max = 0$ ;
    for each parent task  $t_k$  of task  $t_i$  do begin
        if  $t$ -level ( $t_k$ ) + AvgE ( $t_k$ ) + CommTime( $t_k, t_i$ ) >  $max$  then
             $max = t$ -level ( $t_k$ ) + AvgE( $t_k$ ) + CommTime( $t_k, t_i$ );
        endif;
    endfor;
     $t$ -level ( $t_i$ ) =  $max$ ;
endfor;

```

(C)計算任務的優先值:

任務優先值  $Task\_priority(t_i)$  以  $b$ -level 當作任務的優先基準， $t$ -level 當作變異，每個任務的優先值  $Task\_priority$  如下所示：

$$Task\_priority(t_i) = b\text{-level}(t_i) + \text{uniform}(t\text{-level}(t_i)/2, -t\text{-level}(t_i)/2)$$

(D)產生染色體:

任務的優先值  $Task\_priority$  越大，代表其優先順序越前面，因此可以根據每個任務的優先值排出任務的執行先後順序(task sequencing)，產生的染色體，而且根據  $t$ -level 變異影響，每次產生的任務優先值皆為不同，因此可以產生不同的染色體，如圖 5.4 所示任務的優先值轉換為染色體。因為經過問題結構去分析過而產生，染色體比隨機產生的染色體，有較佳的求解品質。

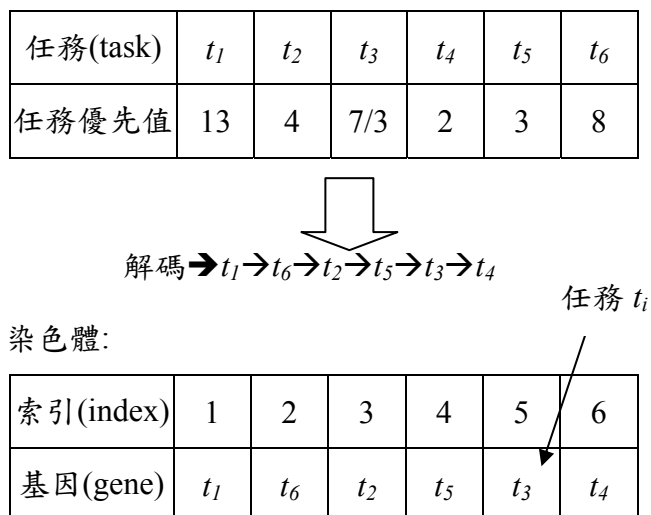


圖 5.4 由任務的優先值轉換為染色體

(E)初始母體:我們利用 PSGA 的方式產生了  $N_i:1000$  個染色體，根據其適應值，抽取前  $N_p:60$  條染色體當作我們初始染色體  $P(0)$ 。

### 5.3.3 產生新染色體的共識因子

傳統的基因演算法對於產生新的染色體一般採用的演化技術(evolutionary techniques)例如:單點交配(one-cut crossover),通常採用隨機兩個染色體區段交配,在這有相依關係的問題中,有時反而會產生不好的解,甚至會有不合理的解,交配的過程中並不考慮上一代好的基因,而盲目搜尋(blind search)。我們主要提出這共識因子(consensus operator),希望可以根據上一代好的染色體中尋找資訊,建立一個導引(guided)機制,根據這引導機制讓好基因順序保留,不好的基因順序則自然淘汰,以產生好的染色體,敘述如下。

#### (A)共識因子產生方式

共識因子產生方式主要分為三個步驟:

Step 1: 建立一個共識池(consensus pool)  $S$

為了讓取得的資訊不會陷入局部解(local trap),我們從母體  $p(t)$  隨機抽取一半的染色體放進我們建立的共識池  $S$  裡。

Step 2: 建立引導機制:對偶優先矩陣 PWP\_matrix(pair-wise priority\_matrix)

本研究提出這引導機制:對偶優先矩陣 PWP\_matrix;  $M$ , 初始為一個 0 矩陣( $n \times n$  matrix,  $n$  為任務的個數), 從共識池  $S$  中每條染色體, 藉由 PWP\_Chromosome 流程去計算任務之間的優先關係矩陣  $A$ , 形成新矩陣  $M (M=M+A)$ , 如圖 5.5 所示。其中 PWP\_Chromosome( $X_k$ ) 流程,  $X_k$  代表輸入的染色體, 我們初始一個任務優先關係矩陣  $A=[a_{ij}]$ , ( $1 \leq i, j \leq n$ ) 也為 0 矩陣( $n \times n$  matrix) 只要染色體  $X_i$  中, 任兩任務  $t_i, t_j$  任務  $t_i$  在任務  $t_j$  之前,  $a_{ij}$  就加 1, 流程如下(其中  $\text{rank}(t_i)$  為染色體的 index, 也為任務的優先順序)。

#### Procedure PWP\_Chromosome( $X$ ):

Initialization: matrix  $A=[a_{ij}] = \mathbf{0}$  ( $n \times n$  matrix)

For

any two tasks  $t_i$  and  $t_j$  in chromosome  $X$ ,  $t_i \neq t_j$

If  $\text{rank}(t_i) < \text{rank}(t_j)$ , then

$a_{ij} = a_{ij} + 1$

endif

Endfor;



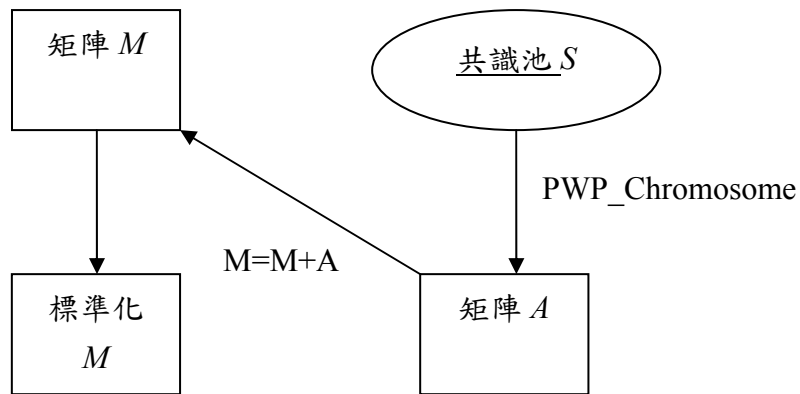


圖 5.5 PWP\_matrix:M 流程

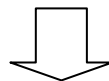
釋例，給定一染色體  $t1 \rightarrow t6 \rightarrow t5 \rightarrow t2 \rightarrow t4 \rightarrow t3$ ，藉由 PWP\_Chromosome 流程，產生如圖 5.6 的任務優先關係矩陣  $A$ 。

任務	1	2	3	4	5	6
1	X		1	1	1	1
2	0	X	1	1	0	0
3	0	0	X	0	0	0
4	0	0	1	X	0	0
5	0	1	1	1	X	0
6	0	1	1	1	1	X

圖 5.6 任務優先關係矩陣

我們還需要把對偶優先矩陣  $M$  標準化為  $M = M/N(S)$ ， $N(S)$ 代表共識池  $S$  所有的染色體個數。標準化的對偶優先矩陣  $M$  特性為互補性  $M_{ij} + M_{ji} = 1$ (圖 5.7)，這矩陣提供資訊，說明在  $S$  集合中，百分 80 的染色體，認為任務  $t_3$  在任務  $t_1$  前執行比較好。因此我們可以知道任兩任務之間  $t_i$  應該在  $t_j$  之前執行的共識比率，因此對偶優先矩陣  $M$  我們也稱為共識矩陣(consensus Matrix)。

任務	1	2	3	4	5	6
1	0	30	6	30	15	21
2	0	0	15	18	3	0
3	24	15	0	21	0	12
4	0	12	9	0	0	6
5	15	27	30	30	0	18
6	9	30	18	24	12	0



標準化:除以  $N(S)=30$

任務	1	2	3	4	5	6
1	0	1	0.2	1	0.5	0.7
2	0	0	0.5	0.6	0.1	0
3	0.8	0.5	0	0.7	0	0.4
4	0	0.4	0.3	0	0	0.2
5	0.5	0.9	1	1	0	0.6
6	0.3	1	0.6	0.8	0.4	0

圖 5.7 模擬 6 個任務 從 PWP\_matrix:  $M$  標準化後的  $M$

根據 PWP\_matrix 的共識建議，我們可以很清楚任務  $t_i$  在任務  $t_j$  之前執行的共識比率。然而機率 1 或 0 為必然任務  $t_i$  在任務  $t_j$  之前執行或之後執行的比例，而在母體  $p(t)$  不斷的演化之下，如圖 5.8，我們很清楚可以看出有些 1 或 0 的比例，經過世代的演化，還是沒有影響，有些卻漸漸演化成 1 或 0。那些不會變化的 1 或 0 其實就是問題本身的優先執行順序限制，因此為**硬性限制(hard constraint)**。而那些慢慢演化為 1 或 0，都是經過共識因子不斷演化，產生較好的共識建議，而演化的**柔性限制(soft constraint)**。因此，我們把這兩種都當作限制，不能去違背，之後共識因子產生的方式，在有 1 或 0 的限制下，要先判定沒有 1 或 0 的限制下的合理區塊(feasible region)，在去選取其共識建議。

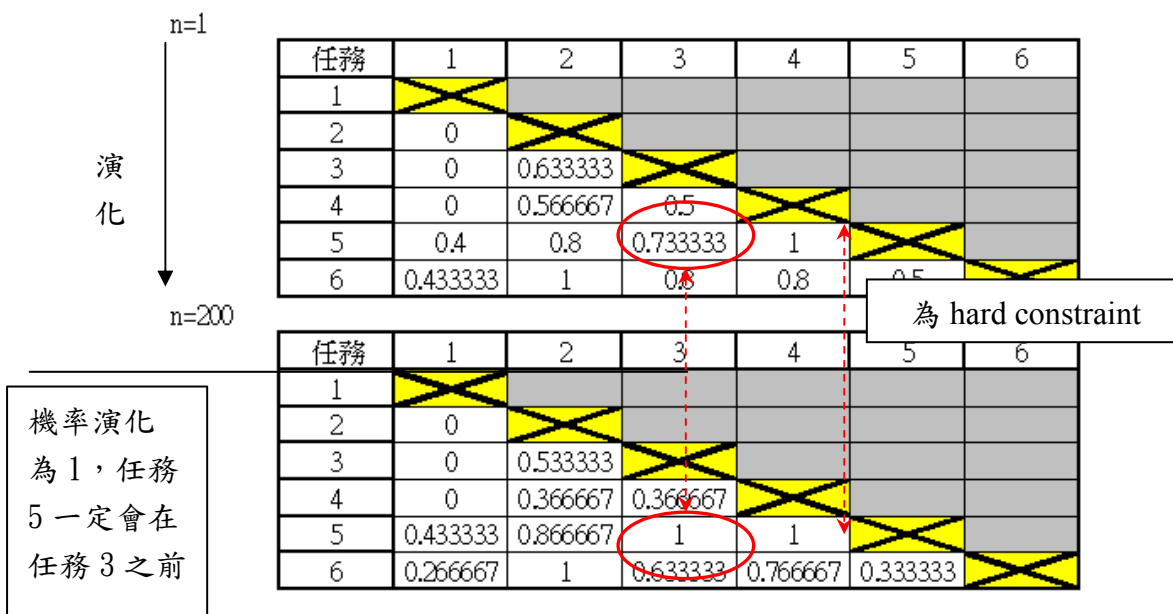


圖 5.8 PWP\_matrix 的硬性限制與柔性限制

Step 3: 根據引導機制 PWP\_matrix 產生一個新的染色體 Y

首先建立一個空集合  $Q = \phi$ ，與所有任務的集合  $X = \{t_i; 1 \leq i \leq n\}$ ， $Z$  為一個很大的數(例如: $Z=100$ )，隨機抽取任務  $q_i$  從  $X$  中，因此  $X = X - \{q_i\}$ ，而根據 PWP\_Matrix 去決定  $q_i$  的優先值  $h(q_i)$ (稱為 PWP 任務優先值)，把已經有 PWP 任務優先值的  $q_i$  放進  $Q$  集合中，為  $Q = Q \cup \{q_i\}$ ，直到抽取到  $X$  為空集合才停止。而新的染色體可以根據每個任務  $q_i$  的 PWP 任務優先值去排序產生，PWP 任務優先值越小任務的順序越前面，如圖 5.9 所示的例子中，排序為  $q_2 \rightarrow q_3 \rightarrow q_1 \rightarrow q_4$ 。

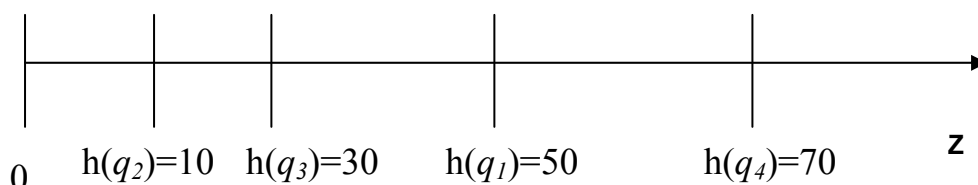


圖 5.9 PWP 任務優先值去排序:  $q_2 \rightarrow q_3 \rightarrow q_1 \rightarrow q_4$

我們已知 PWP\_matrix :M 還有在  $Q$  集合中的所有任務  $Q = \{q_1, \dots, q_{i-1}\}$  和在  $Q$  中已計算的 PWP 任務優先值  $h(q_k)$  ,  $q_k \in Q$  , 如何計算任務  $q_i$  的 PWP 任務優先值, 我們可以分為下列五個步驟:

(1) 決定  $h(q_i)$  合理區域:

因任務有優先執行順序的限制, 我們首先要判定不違反優先執行順序的限制出  $h(q_i)$  的合理區域, 由之前例子中我們可以在  $M$  中尋找出強烈限制與柔性限制為 0 or 1, 設定初始合理區域的開始點為  $h_s = 0$ , 結束點為  $h_e = Z$ 。對於每一個  $q_k$ ,  $q_k \in Q$ , 如果  $m_{ik} = 0$  和  $h(q_k) > h_s$  則  $h_s = h(q_k)$ , 和如果  $m_{ik} = 1$  和  $h(q_k) < h_e$  則  $h_e = h(q_k)$ , 更新後的合理區域, 表示新的  $q_i$  在這區域中都不會有違反優先執行順序的限制。式子如下所示:

```

For k=1 to i-1, /*for each task in Q*/
    If  $m_{ik} = 0$  and  $h(q_k) > h_s$ , then  $h_s = h(q_k)$ 
    If  $m_{ik} = 1$  and  $h(q_k) < h_e$ , then  $h_e = h(q_k)$ 
Endfor

```

(2) 在合理區域中選擇參考任務  $q_r$ :

已知合理區域所有的  $Q_h$ ,  $Q_h = \{q_k | h_s \leq h(q_k) \leq h_e, q_k \in Q\}$ , 為了尋找合理區域中的參考任務(reference task)  $q_r$ , 我們希望尋找共識比例最大的, 因此如下式(1)為  $q_i$  在  $q_k$  之後機率最大, 和式(2)  $q_i$  為在  $q_r$  之前機率最大, 兩者相比機率最大如式(3), 為我們要尋找的參考任務  $q_r$ , 也代表任務  $q_i$  在任務  $q_r$  之前的共識比例為  $p = M(q_i, q_r)$

$$k^* = \text{Arg Max} \{ M(q_k, q_i) \}, q_k \in Q_h \quad (1)$$

$$r^* = \text{Arg Max} \{ M(q_i, q_r) \}, q_r \in Q_h \quad (2)$$

$$r = \text{Arg Max} \{ M(q_i, q_{r^*}), M(q_{k^*}, q_i) \} \quad (3)$$

(3) 根據 PWP\_matrix, 執行伯努力(Bernoulli)試驗:

為了讓任務執行順序的產生變異, 我們執行一次伯努力(Bernoulli)試驗, 根據  $q_i$  與  $q_r$  的共識比例, 當作機率, 實驗成功代表任務  $q_i$  在任務  $q_r$  之前執行, 因此  $h(q_i) < h(q_r)$ , 合理區域更新為  $h_e = h(q_r)$ ; 或是實驗失敗代表任務  $q_i$  在任務  $q_r$  之後執行, 因此  $h(q_i) > h(q_r)$ , 合理區域更新為  $h_s = h(q_r)$ 。

(4) 更新合理區域:

實驗成功: 合理區域更新為  $h_e = h(q_r)$

實驗失敗: 合理區域更新為  $h_s = h(q_r)$

(5) 停止條件:

我們尋找到當合理區域中可插入的任務數  $n(Q_h)$  少於兩個, 便可決定其  $h(q_i)$  值。因此  $n(Q_h) > 2$  我們繼續尋找新的參考任務  $q_r$ , 回到步驟(2), 直到  $n(Q_h) \leq 2$ , 則決定  $h(q_i) = (h_s + h_e)/2$ 。

茲以一釋例說明如何決定  $h(q_i)$ ，如圖 5.10 所示，在已知  $Q=\{q_1, q_2, q_3, q_4\}$ ， $Q_h=\{q_k | h_s \leq h(q_k) \leq h_e, q_k \in Q\}$  下，若要插入任務  $q_5$ ，決定  $h(q_5)$ 。第一步驟:先決定  $h(q_i)$  合理區域為  $h_s = h(q_1)$ ,  $h_e = h(q_4)$ 。第二步驟:為了尋找參考任務  $q_r$ ，根據(1)(2)(3)式，可以求得  $r=q_2$ 。第三步驟:根據  $q_2$  做伯努力(Bernoulli)試驗，如果失敗， $h(q_5) > h(q_2)$ 。第四步驟:合理區域更新為  $h_s = h(q_r) = h(q_2)$ 。第五步驟:  $n(Q_h) = 2$ ，停止搜尋， $h(q_5) = (h(q_2) + h(q_4)) / 2$ ， $q_5$  插入在  $q_4$  和  $q_2$  之間。

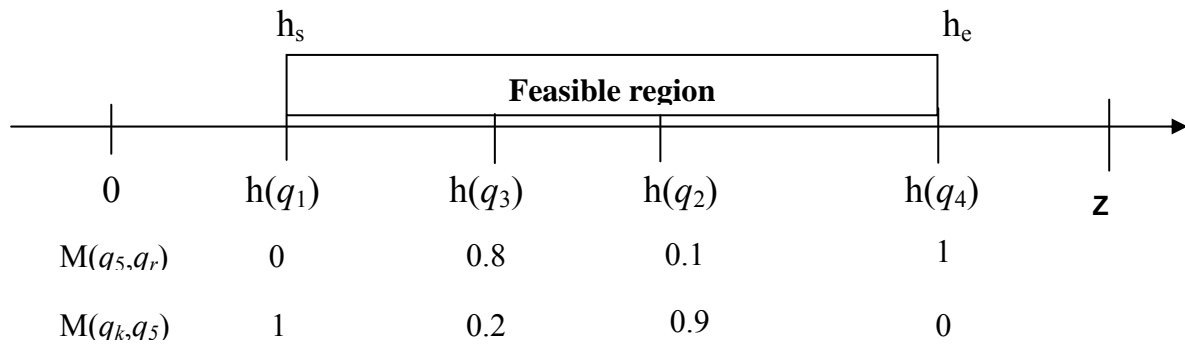


圖 5.10 選擇參考任務  $q_r$

### (B)調整共識因子影響力

從實驗發現，由共識因子所產生的染色體，強烈的被 PWP\_matrix 所主導，導致所產生的染色體容易陷入局部解當中，為了改進這個缺點，我們希望任務順序可插入的地方可以大幅跳躍，因此我們針對任務  $q_i$  對於參考任務  $q_r$  的共識比例  $p$ ，乘以一個降級因子  $r$  (degrading factor),  $0 < r < 1$ ，讓任務  $q_i$  比較不會照原本抽取的染色體排列順序排列，因此我們在共識因子中設計了兩種方式去產生新的染色體，一種是原本的共識因子  $P1=p$ ，產生母體  $P(t) * C_1$  條染色體;另一種為降級的共識因子  $P2=p*r$ ，產生母體  $P(t) * C_2$  條染色體。

### 5.3.4 選擇染色體形成新母體的方法

母體  $P(t)$  之染色體數目  $N_p$  加上經由共識和降級共識所產生的新染色體數目後，共有  $f = P(t) * (1 + C_1 + C_2)$  條染色體，此  $f$  條染色體放在一集合  $S_p$ ，選擇策略就是從此集合  $S_p$  選出  $N_p$  條的染色體放置於母體  $P(t+1)$  中。本文將以 rank-space 演算法。求解，其步驟如下：

步驟一：根據染色體之適應函數值遞減排序，依照排序結果給定一個  $Z_i$  品質序 (quality-ranking)，表示為  $R_q(Z_i)$ 。假設排序後結果為  $Z_1, Z_2, \dots, Z_f$ 。

步驟二：將品質序第一的染色體從  $S_p$  移到  $P(t+1)$

$$S_p = S_p - \{Z_1\};$$

$$P(t+1) \leftarrow \{Z_1\};$$

$$Y_1 = Z_1; /* 將放置在 P(t+1) 的染色體命名 */$$

$$N = 1; /* 計算放置在 P(t+1) 的染色體總數 */$$

步驟三：對  $S_p$  中的染色體，計算差異指標  $D(Z_i)$  (diversity index)

$$D(Z_i) = \sum_{k=1}^N \frac{1}{|Z_i - Y_k|}$$

步驟四：根據  $D(Z_i)$  對  $S_p$  中所有剩下的染色體遞增排序，依照排序結果給定  $Z_i$  一個差異序(diversity-ranking)，表示為  $R_d(Z_i)$ 。

步驟五：計算  $S_p$  中所有染色體之品質序和差異序的總合

$$T(Z_i) = R_q(Z_i) + R_d(Z_i)$$

步驟六：根據  $T(Z_i)$  對  $S_p$  中所有染色體遞增排序，排序結果表示為  $R_c(Z_i)$ 。

步驟七：對  $S_p$  中所有染色體計算可能存活到  $P(t+1)$  的機率

$$r = R_c(Z_i);$$

$$Prob(Z_i) = p \cdot (1-p)^{r-1}; \quad /* p 為事先給定的機率，一般訂為 0.667 */$$

步驟八：產生一隨機值，根據上述的機率分佈從  $S_p$  中隨機挑選一條染色體至下一代母體  $P(t+1)$ ，假設  $Z_m$  為被挑選之染色體，則移動  $Z_m$  至下一代。

$$S_p = S_p - \{Z_m\};$$

$$P(t+1) \leftarrow \{Z_m\};$$

$$Y_m = Z_m; \quad /* P(t+1) 的染色體重新命名 */$$

$$N = N+1; \quad /* 更新 P(t+1) 的染色體數量 */$$

步驟九：確認是否選滿  $N_p$  個染色體

if  $N < N_p$ ，則回步驟三

else 停止

基因演算法終止條件有兩種，即當近似最佳解持續維持達  $N_c$  個世代；或者演化世代  $t$  達到  $N_g$  時，則停止演化，並且認定最終近似解為最佳解。因為實際問題最佳解無法求證得知，我們只採用演化世代  $t$  達到  $N_g$  時，則停止演化， $N_g=100$ 。

總之，演算法可以被分為兩個階段。第一階段為一個初始化的階段。對於 DHC 問題輸入 DAG、期望執行時間矩陣  $E(\text{machine list})$ 。還有輸入使用者基因演算法參數： $N_i$  表示用 PSGA 方式產生的染色體個數， $N_p$  表示在母體  $P(t)$  的染色體總數， $t$  表示所經過之世代， $N_g$  為演化的世代個數， $C_1$  為共識因子產生比例， $C_2$  為降級共識因子產生比例， $r$  為降級因子。首先利用 PSGA 方式去計算在 DAG 中每個任務的優先值，根據每個任務的優先值去轉換成任務執行順序的染色體，產生個數為  $N_i$  個，抽取前  $N_p$  個，作為初始母體  $P(0)$ 。

第二階段為一個搜尋階段，母體  $P(t)$  的染色體經過一些基因運算子的處理，並根據適應值 (fitness value) 決定其存活率，再被挑選到下一代母體  $P(t+1)$ 。基因運算子主要有三種，複製 (reproduction)、共識 (consensus)、降級共識 (degrading consensus)，而當符合終止條件 (termination conditions) 時，則不再進行世代的演化，取得求解答案。

## 5.4 實驗結果

在 DHC 問題環境中由於沒有標準的測試案例，我們採用 Boyer[30] 的方式，

產生 12 種模擬情境根據 100 任務 20 個機器，每個模擬情境根據 DAG 與 machine list 組成。DAG 分為不同相依關係程度；任務的異質度程度、機器的異質度程度和機器是否有一致性(consistent)特性所產生的期望執行時間矩陣  $E(\text{machine list})$ 。

本研究把我們提出的演算法和最近幾年提出的演算法 RS、RSGA 和 PSGA 做實驗比較，在小 case 的題目中，幾乎所有方法都可以求解到最佳解，因此我們把焦點放在大 case 的題目中，在十二個模擬情境，根據 20 個機器組成的 DHC 和 100 個任務的 DAG，其結果見圖 5.11，從結果中，共識因子為基礎的演算法比其他兩者演算法有顯著差異。

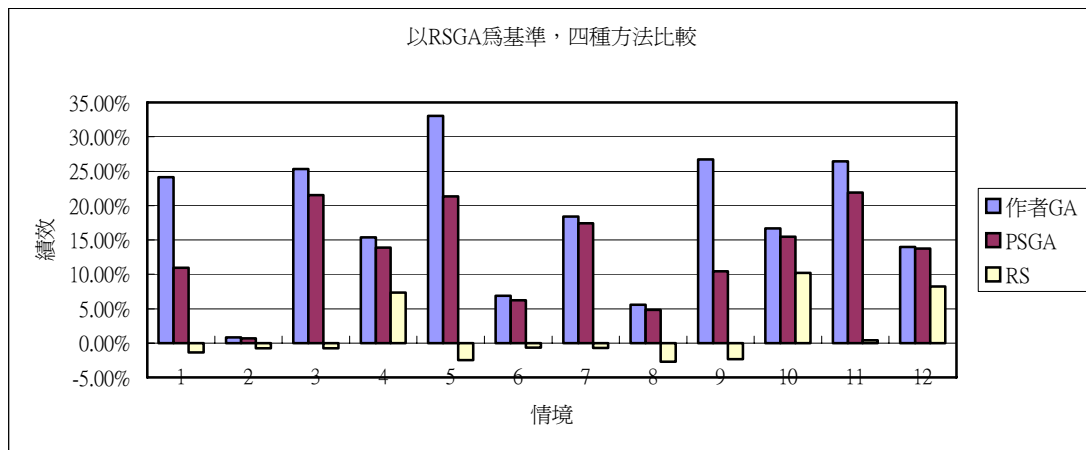


圖 5.11 實驗結果

## 第六章 結論與建議

### 6.1 研究結論

本研究針對 TFT-LCD 產業，在低良率情境下，進行三個製造管理議題的研究，各子題的重要結論彙述如下。

子題一是研究基板的重工決策。此研究在 TFT-LCD 廠面臨低良率情境時，可有效決定 TFT-Plate 製程及 CF-Plate 各檢測站的最佳重工決策，與目前業界所使用的方法相比較，可顯著提高 TFT-LCD 的產出數量。此研究對產業界和學術界都有貢獻，目前研究方法已經獲得一項中華民國專利。

子題二是研究基板配對與重組決策，此研究的成果可將 TFT-Plate 和 CF-Plate 都是的承載批，有效重新配對，並且決定排序機的派工，使產出流數更均勻，使得下游的生產線不會因缺料而停機。此研究成果，對產業界的貢獻相當具體明確，目前經獲得兩項專利，其中一項是中華民國專利，一項是大韓民國專利。

子題三是研究產品組合與投料決策，此子題的本質是一個解空間的搜尋問題，研究成果的貢獻不是很顯著。但是在研究過程中，為了研究解空間搜尋，我們意外發展出一個新的搜尋演算法，用異質的專案排程效果相當良好。此研究成果，已經獲得一項中華民國專利。

### 1.2 未來研究建議

進行上述三子題的研究時，我們也意外發現一個低良率情境的步進機派工問題。此問題是出現在半導體廠的步進機，當新製程或新產品導入時，良率過低，可能造成步進機的利用率偏低，但有效派工可舒緩此產能閒置問題。我們未來將針對此問題繼續研究。

## 參考文獻

- [1] 巫木誠、盧威豪、吳佳恩「物料報廢架構及物料報廢準則產生方法」，中華民國發明專利（證書號數：I323427）
- [2] 巫木誠、陳德珊：「基板排序派工之方法」，大韓民國專利，（證書號數：10-0816624）
- [3] 巫木誠、陳德珊：「基板排序派工之方法」，中華民國專利，（證書號數：I315476）
- [4] 巫木誠、吳政翰：「相依排程之派工方法」，中華民國發明專利（證書號數：I313437）
- [5] S.D.P Flapper, J.C. Fransoo, R.A.C.M. Broekmeulen, K. Inderfurth, “Planning and control of rework in the process industries: A review,” *Production Planning and Control* 13 (1), pp. 26–34, 2002.
- [6] N. Geren and E. K. Lo, “Equipment selection, integration and interfacing of a robotic PCBA component rework cell,” *Comput. Integr. Manufact. Syst.*, vol 11, no. 1-2, pp. 77-89, 1998.
- [7] N. Geren and A. Redford, “Automated rework of printed circuit board assemblies: Methods and procedures,” *Int. J. Comput. Integr. Manufact.*, vol. 9, no. 1, pp. 48–60, 1996.
- [8] N. Geren and A. Redford, “Cost and performance analysis of a robotic rework cell,” *Int. J. Prod. Econ.*, vol. 58, pp. 159–172, 1999.
- [9] N. Geren, “Determination and Comparison of Surface Mount Component Rework Cycle Times of Manual and Fully Automated Robotic Rework Stations for Rework Technology Selection,” *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 26, No.2, April, 2003.
- [10] M. R. Driels and J. S. Klegka., “Analysis of alternative rework strategies for printed wiring assembly manufacturing systems,” *IEEE Trans. Comp., Hybrids, Manufact. Technol.*, vol. 14, pp. 637–644, Sept, 1991
- [11] S.D.P. Flapper, R. H. Teunter, “Logistic planning of rework with deteriorating work-in-process,” *Int. J. Economics* 88 pp. 51-59, 2004.
- [12] A. M. Zargar, “Effect of rework strategies on cycle times,” *Computers and Industrial Engineering* 29, pp. 239-243, 1995.
- [13] J. J. Liu, P. Yang, “Optimal lot sizing in an imperfect production system with homogeneous reworkable jobs,” *European Journal Operational Research* 91 (3), pp. 517–527, 1996
- [14] G. K. Tay, D. P. Ballou, “An integrated production-inventory model with reprocessing and inspection,” *International Journal of Production Research* 26 (8), pp. 1299-1315, 1988.
- [15] C.-C. Chern, P. Yang, “Determining a threshold control policy for an imperfect production system with rework jobs,” *Naval Research Logistics* 46, pp. 273–301, 1999.
- [16] K.C. So, C. S. Tang, “Optimal batch sizing and repair strategies for operations with repairable jobs,” *Management Science* 41 (5), pp. 894-908, 1995.
- [17] Daigle, P. Powell, “Manufacturing scrap reduction team,” *IEEE/SEMI Advanced Manufacturing Conf.*, pp.230-231, 1996.
- [18] M. E. Kuhl, G. R. Laubisch, “A Simulation Study of Dispatching Rules and



- Rework Strategies in Semiconductor Manufacturing,” *IEEE/SEMI Advanced Manufacturing Conf.*, pp.325-329, 2004.
- [19] D. N. Maynard, D. S. Kerr, and C. Whiteside, “Determining cost of yield to monitor fab manufacturing processes” (in <http://www.micromagazine.com/archive/03/06/maynard.html>), *Micromagazine.com*, pp.63-69, June 2003.
- [20] M. C. Wu, C. W. Chiou, and H. M. Hsu, “Scrap rules for small lots in wafer fabrication,” *2002 Semiconductor Manufacturing Technology Workshop*, Hsin-Chu, Taiwan, R.O.C., pp. 181-184, 2002.
- [21] M. C. Wu, C. W. Chiou, and H. M. Hsu, “Scrapping small lots in a low-yield and high-price scenario,” *IEEE Transactions on semiconductor manufacturing*, vol. 17, no. 1, Feb, pp. 55-67, 2004.
- [22] 楊毅臻 (2003) 以基板配對為基礎的 LCD 廠排序系統規劃之研究，華梵大學工業管理研究所碩士論文
- [23] 楊佳翰 (2004) 應用改良型巨集式啟發式方法於 TFT-LCD 良率控制最佳化之研究，交通大學工業工程與管理研究所碩士論文
- [24] 張東華 (2005) TFT-LCD 組立製程配對作業之研究，交通大學工業工程與管理研究所碩士論文
- [25] 王君豪 (2005) TFT-LCD 排序機派工之研究，交通大學工業工程與管理研究所碩士論文
- [26] R. Kolisch, “Serial and parallel Resource-constrained Project Scheduling Methods Revisited: Theory and computation,” *European Journal of Operation Research*, vol. 90, pp.320-333, 1996.
- [27] L. Wang, H. J. Siegel, V. P. Roychowdhury, A. A. Maciejewski, “Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach,” *Journal Parallel Distributed Computing*, vol 47, pp. 8–22, 1997.
- [28] G. C. Sih and E. A. Lee, “A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, pp.175-186, 1993.
- [29] H. Topcuoglu, S. Hariri, and M. Y. Wu, ”Performance-effective and low-complexity task scheduling for heterogeneous computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp.260-274, 2002.
- [30] W. F. Boyer and G. S. Hurab, “Non-evolutionary algorithm for scheduling dependent tasks in distributed heterogeneous computing environments,” *Journal Parallel Distributed Computing*, vol. 65, pp.1035 – 1046, 2005
- [31] P. Shroff, D. W. Watson, N. S. Flann and R. F. Freund, “Genetic simulated annealing for scheduling data-dependent tasks in heterogeneous environments,” *Proceedings of the Heterogeneous Computing Workshop*, pp.98-104, 1996.
- [32] M. Dhodhi, I. Ahmad, A. Yatama, I. Ahmad, “An integrated technique for task matching and scheduling onto distributed heterogeneous computing systems”, *Journal Parallel Distributed Computing*, pp. 1338-1361, 2002.
- [33] Y. K. Kwok, I. Ahmad, “Static scheduling algorithms for allocating directed task graphs to multiprocessors,” *ACM Computing Surveys*, pp 406–471, 1999.