

國立交通大學
工業工程與管理學系

碩士論文

TFT-LCD 組立製程之基板配對演算法

A Plate-Mapping Algorithm for the Cell Process in
TFT-LCD Manufacturing



研究生：張東華

指導教授：巫木誠 博士

中華民國九十四年六月

TFT-LCD 組立製程之基板配對演算法

研究生：張東華

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

在薄膜液晶顯示器(TFT-LCD)製造過程中，包含一個組立製程 (Cell Process)，其中有一個主要的作業是將薄膜電晶體基板 (TFT plate) 與彩色濾片基板 (CF plate) 進行配對。一個典型的 TFT 基板約可包含 6-30 面板 (panel)，當 TFT 面板與 CF 面板二者中若有一個面板為不良品 (defective)時，則組立後的 TFT-LCD 也為不良品。如何將基板適當配對來提高製程良率，稱之為基板配對問題 (plate mapping problem)，對 TFTLCD 產業而言是一個非常重要的問題。在過去文獻對基板配對問題描述成一整數規劃(integer linear programming)，並利用匈牙利指派法(Hungarian method)來求解。經過實證資料測試，整數規劃求解存在多組解，每一組解有相同的良率，但對卡匣上下排序機的次數卻未必相同。本研究發展出發展一整數規劃模型簡稱為 CIP(comprehensive integer programming)，期望在多組解中找出最有利於減少卡匣上下排序機的次數的基板配對解。在某些情境下，CIP 求解時間可能過長，因此本論文另發展一啟發式法 (heuristic method)，稱為 Tune_Matrix，來配合該模型求解，以兼顧求解的品質及求解時間。

關鍵詞：TFT-LCD、整數規劃、基板配對、Cell 製程

A Plate-Mapping Algorithm for the Cell Process in TFT-LCD Manufacturing

Student : Dung-Hua Chang

Advisor : Dr. Muh-Cherng Wu

Institute of Industrial Engineering

National Chiao Tung University

ABSTRACT

The manufacturing of TFT-LCD involves a cell process, one major operation of which is to mate a TFT plate with a CF (color filter) plate. A typical TFT/CF plate comprises 6-30 panels. Such a mated panel is defective in quality when any one of the two panels is defective. How to mate the TFT/LCD plates, also called the *plate-mapping problem*, is thus very important to achieve a high yield. The plate-mapping problem was modeled in literature as an integer program and solved by the Hungarian method. By simulation, we find that there exist a great multiple solutions in the integer problem. Each of these solutions, though with the same yield, requires different number of loading/unloading operations. This research proposes an more comprehensive integer program (CIP) in order to identify a solution with highest mapping yield as well as with fewest number of loading/unloading requirements. The computation time required to solve the CIP may be quite lengthy. A heuristic procedure, called Tune_Martix, is developed to enhance the CIP in order to quickly identify a satisfactory one from the multiple solutions.

Keywords : TFT-LCD, integer program, plate-mapping, cell process

目錄

中文摘要	i
英文摘要	ii
目錄	iii
圖目錄	v
表目錄	vi
第一章 緒論	1
1.1 研究背景	1
1.2 研究問題	3
1.3 研究目的	4
1.4 論文章的安排	4
第二章 TFT-LCD 產業文獻探討	5
2.1 TFT-LCD 製程簡介	5
2.2 TFT-LCD 製造管理相關文獻	6
第三章 模式建構	7
3.1 符號定義	7
3.2 整數規劃模式	8
3.3 模式分析比較	9
3.4 設定值 ω 和 r 的推估	9
3.5 啟發式演算法	10
第四章 實例驗證	13

4.1 軟體與硬體	14
4.2 測試情境設計	15
4.3 CIP/SIP 求解品質與效率	15
4.4 Tune_Matrix 的求解品質與效率	16
第五章 結論與未來研究方向	18
5.1 結論	18
5.2 未來研究方向	18
參考文獻	19



圖目錄

圖 1.1 TFT 基板與 CF 基板組立作業	2
圖 1.2 排序機系統	3
圖 2.1 TFT-LCD 製作流程	6
圖 3.1 CIP 求解後基板配對關係矩陣	12



表目錄

表 4.1 卡匣、相對最佳容許值設定	15
表 4.2 CIP/SIP 規劃結果彙整	17
表 4.3 Tune_Matrix 結果分析	16



第一章 緒論

1.1 研究背景

薄膜電晶體液晶顯示器 (Thin Film Transistor Liquid Crystal Display, TFT-LCD) 是一個前景看好的產業，目前被廣泛的應用到電腦監視器、筆記型電腦、電視、手機及遊戲機等產品，大部份的視覺顯示裝置逐漸已被 TFT-LCD 所取代。然而 TFT-LCD 產業具有資本密集度高，價格波動大等特性 [1]，如何提高製程良率以降低成本，如何縮短生產週期時間以掌握高價賣點，都是 TFT-LCD 產業非常重要的議題。

TFT-LCD 的製作包含下列三個主要製程：陣列 (array) 製程、組立 (cell) 製程、模組 (module) 製程。陣列製程類似半導體製程，是用以製作薄膜電晶體基板 (TFT plate)，組立製程是將 TFT 基板與彩色濾光片基板 (color filter plate, CF plate) 進行組立壓合，注入液晶形成 TFT-LCD 基板，然後將 TFT-LCD 基板切割成 6-30 個 TFT-LCD 面板 (panel)。模組製程是將驅動電路板、背光模組等其他零件安裝到面板上，以完成 TFT-LCD 的最終成品。

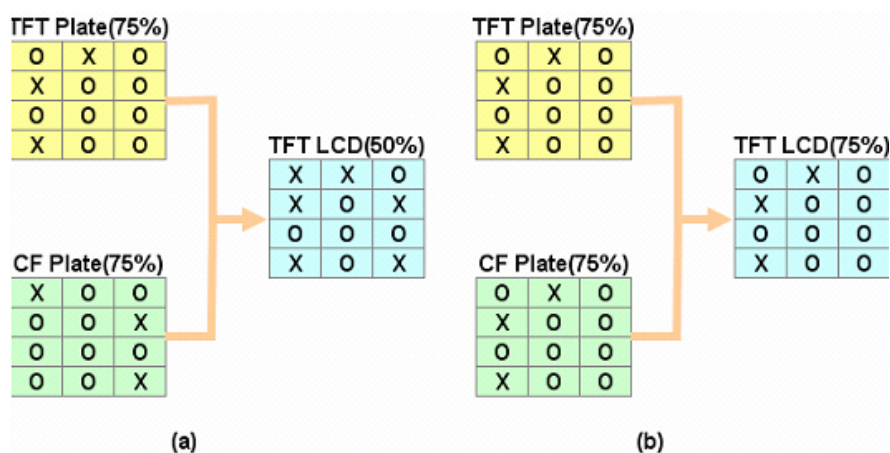
基板 (plate) 依應用目的，可分割成不同尺寸的面板 (panel)，基板良率是以面板良品數除於總面板數計算。生產一個良品的 TFT-LCD 面板，必須 TFT 面板與 CF 面板二者都為良品，兩者中只要有一個面板為不良品，則組立後的 TFT-LCD 面板就是不良品。因此 TFT 基板和 CF 基板組合後的良率 (簡稱配對良率, mapping yield)，會依基板上面板的良品位置分布而定。如圖 1.1 所示的兩釋例，各例的 TFT 基板和 CF 基板良率皆相同，但組合後的配對良率卻有差異，因此基板配對決策非常重要。

TFT 基板和 CF 基板是用卡匣 (cassette) 來承載運輸，一個卡匣約裝 10-20 個基板。假設一製程的卡匣可以承載 n 片基板，則一組 TFT 卡匣和 CF 卡匣的基板

配對，將有 $n!$ 組合，當有 N 組卡匣時，則基板配對有 $(N \times n)!$ 個組合，因此如何將基板適當配對 (plate mapping) 以最大化 N 組卡匣基板的配對良率，對 TFT-LCD 產業是一個非常重要問題。

N 組卡匣的基板完成配對決策(mapping decision)之後，須將卡匣內的基板抽換，使 TFT 卡匣和 CF 卡匣可以匹配成對。亦即，在抽換基板之後，每一 CF 卡匣皆有一匹配的 TFT 卡匣，此兩卡匣內的基板完全符合既定的配對決策。此基板抽換的程序可進而了解如下，固定所有 CF 卡匣內的基板，只抽換 TFT 卡匣的基板，以逐一產生與每一個 CF 卡匣匹配的 TFT 卡匣。本研究以下將抽換完成的卡匣稱為 TFT 目標卡匣(target cassette)，抽換前的卡匣稱之為 TFT 原始卡匣(source cassette)。

基板抽換的機器稱為排序機 (sorter)，排序機的組成結構如圖 1.2 所示，包含一個機器手臂 (robot)、數個終端埠 (destination ports)、數個來源埠 (source ports)。機器手臂是用來移動基板，終端埠是用來放置空卡匣，來源埠是用來放置原始卡匣。機器手臂將基板從來源埠移到終端埠，使空卡匣成為一目標卡匣。由於 TFT 基板的面積很大，因此 TFT 卡匣上下排序機的時間頗長。如何減少卡匣上下排序機的次數，對於縮短生產週期時間實在非常重要。



面板 = O，表示檢測後為良品 面板=X，表示檢測後為不良品

圖 1-1 TFT 基板與 CF 基板組立作業

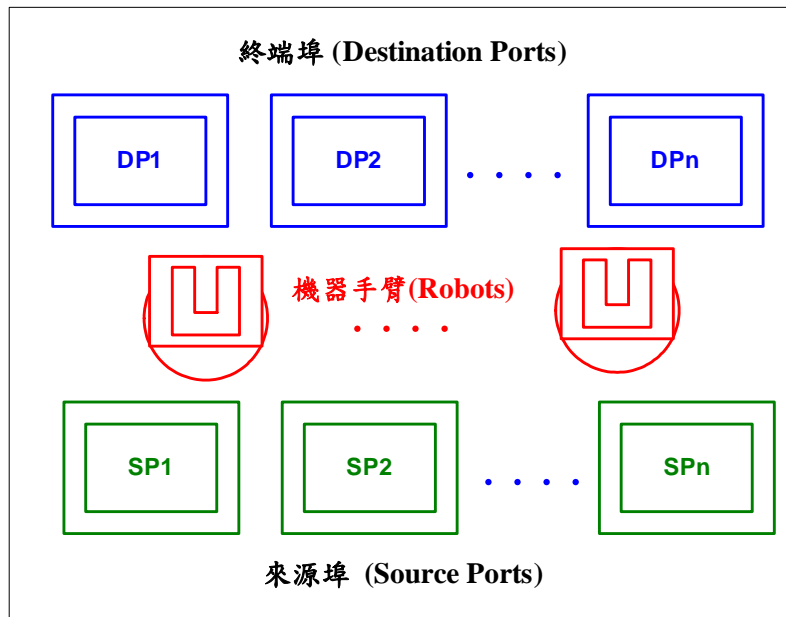


圖 1-2 排序機系統

1.2 研究問題

根據上述分析，吾人可知基板良率配對與減少卡匣上下排序機的次數，在 TFT-LCD 製程非常重要。有關基板良率配對的決策，過去已經有研究將其描述成一整數線性規劃(integer linear programming)問題，並且利用匈牙利指派法 (Hungarian method) 來求解[3]，以縮短求解時間。

為縮短卡匣上下排序機的次數，楊佳翰 [4] 提出以基因演算法 (genetic algorithm, GA)，發展良率配對的方法。該研究假設排序機有 k 個終端埠、 k 個來源埠，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)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[3]應用上述的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

定目標卡匣和來源卡匣上下排序機的優序，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，該研究隱含假設該線性規劃問題只有一組最佳解，並據以求解排序機的派工問題。

根據吾人以實證資料測試發現，此良率配對的整數線性規劃問題其實有多組解，雖然這些解的基板配對良率都一樣；然而，就卡匣上下排序機的次數而言，這些解的品質卻未必相同。

1.3 研究目的

為克服基板配對良率決策多組解的問題，本研究發展一整數線性規劃模型，期以在多組解中找出最有利於排序機作業的基板配對解。由於此模型在某些情境下，求解時間可能過長，因此本論文另發展一啟發式法(heuristic method)來配合該模型求解，以兼顧求解的品質和求解的時間。

1.4 論文章節安排

本論文章節安排如下。第二章稍加詳細說明 TFT 製程，並探討與 TFT-LCD 製造管理相關的文獻。第三章說明本研究所發展的混合整數線性規劃模型，和縮短該模型計算時間的啟發式解法。第四章討論用以驗證求解方法的模擬案例。第五章是結論與建議。

第二章 文獻回顧

2.1 TFT-LCD 製程簡介

薄膜液晶顯示器 (TFT-LCD) 是利用液晶材料的特性，透過外加電路驅動液晶轉向，以控制外部光源穿透與否，進而達到明暗不同的效果，若在結構中加上彩色濾光片，便能顯現色彩化的畫面。

TFT-LCD 之生產製造技術結合半導體、化學、材料、光電等產業之製造技術，主要製程可分為三個部份：(1) 陣列製程 (array process)，(2) 組立製程 (cell process)，(3) 模組製程 (module process)，如圖 2-1 所示。

- 陣列製造技術主要是將玻璃基板透過類似半導體製造技術(鍍膜、曝光、顯影、蝕刻等技術)，在基板上形成為數眾多的電晶體之陣列基板。
- 組立製造技術主要是利用滴注式液晶灌注製程(ODF, one-drop fill)，將 Array 製程完成的基板與彩色濾光片基板分別作配對對準，再進行框膠燒成，並注入液晶，並透過機械對位完成組立，最後切割成預定尺寸的面板 (panel)，並將偏光板貼付，最後做檢測工作。
- 模組製程，主要是將切割完成的面板與驅動 IC、電路板、背光板等外部零組件組立起來，之後再做最後的檢查。

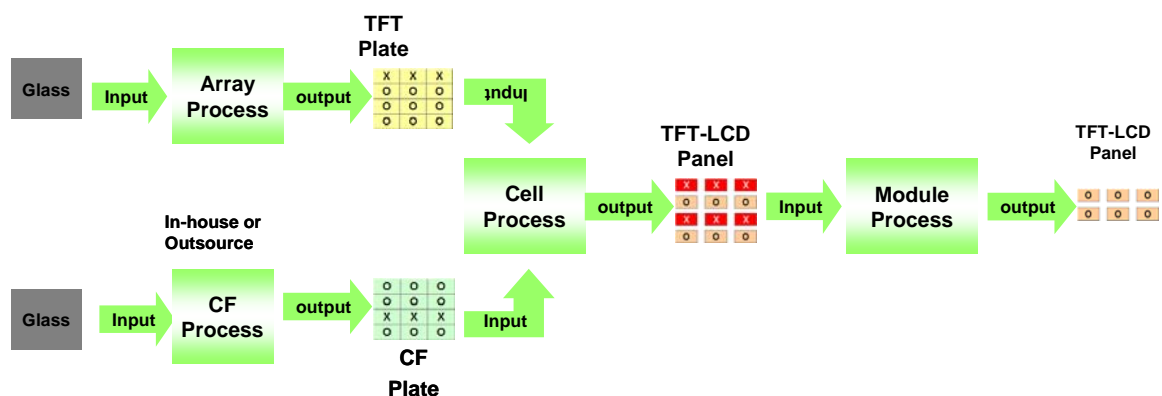


圖 2.1 TFT-LCD 製作流程

2.2 TFT-LCD 製造管理相關文獻

過去 TFT-LCD 製造管理相關的文獻，主要著重在研究生產排程 (production scheduling)。如李氏[2] 利用限制理論 (TOC) 之觀念，發展一同步化集批法則 (synchronized batching heuristic)，來解決 Cell 製程中批量生產派工方法，目的減少產能損失及縮短生產週期時間。謝氏[5] 利用先進規劃與排程系統 (APS) 來解決 TFT-LCD 產業在多廠區 (FAB) 的各自不同限制下，同步協調各廠的排程計劃，以提昇各廠的產能利用率。Jenog 等人[7] 利用啟發方法 (heuristics) 來解決 Cell 製程的排程規劃，目的求出最小的平均流程時間。Shin and Leon [9] 利用 MULTIFIT 和禁忌搜尋 (tabu search) 來發展出平行機台排程方法，解決 TFT-LCD 模組製程機台換線的問題。Toba[10] 利用模擬系統針對線上每一個批 (lot) 的在製品 (work in process, WIP) 進行排程控制，目的在不影響生產產能下，減少 WIP 的等待時間。

本研究所探討的基板良率配對和排序機派工問題，就吾人所知過去只有兩篇文獻[3-4]，此兩文獻已經於第一章介紹。

第三章 模式建構

本研究擬發展一整數線性規劃模型，來求解 TFT-LCD 的基板配對問題，期能在多組配對良率最佳解中，找到一組解能使 TFT 來源卡匣上下雙埠排序機 (two-port sorter) 的次數最少。本章首先描述此整數線性規劃模型，然後討論此模型的求解方法。

3.1 符號定義

■ 註標

k : TFT 卡匣 k 之編號;

m : CF 卡匣 m 之編號;

i : TFT 基板 i 之編號;

j : CF 基板 j 之編號;

■ 參數

N : 每類卡匣的總數量，單位：卡匣 (cassette)

n : 每個卡匣的基板數，單位：基板 (plate)

p : 每個基板的面板數，單位：面板 (panel)

H : 為一個極大的正數

ω : 為一個極小的正數

a_{kimj} = 第 k 個 TFT 卡匣中的第 i 個基板配對到第 m 個 CF 卡匣中第 j 個基

板的良率，單位%， $1 \leq i \leq n$ ， $1 \leq j \leq n$ ， $1 \leq k \leq N$ ， $1 \leq m \leq N$

■ 變數

$x_{kimj} = 1$ ，若第 k 個 TFT 卡匣中的第 i 個基板被配對到第 m 個 CF 卡匣中第

j 個基板，否則 $x_{kimj} = 0$ ， $1 \leq i \leq n$ ， $1 \leq j \leq n$ ， $1 \leq k \leq N$ ， $1 \leq m \leq N$

$q_{km} = 1$ ，若 k 個 TFT 卡匣中有基板被配對到第 m 個 CF 卡匣中的基板，

否則 $q_{km} = 0, 1 \leq k \leq N, 1 \leq m \leq N$ 。亦即，若 $\sum_{j=1}^n \sum_{i=1}^n X_{kimj} \geq 1$ ，則 $q_{km} = 1$ ；若 $\sum_{j=1}^n \sum_{i=1}^n X_{kimj} = 0$ ，則 $q_{km} = 0$ 。

$M = \sum_{m=1}^N \sum_{k=1}^N q_{km}$ ，TFT 來源卡匣的總移動次數，

$Y = \sum_{k=1}^N \sum_{m=1}^N \sum_{i=1}^n \sum_{j=1}^n a_{kimj} x_{kimj} / N \times n$ ，卡匣配對的平均良率

3.2 整數規劃模式

$$\text{Max } Z = Y - \omega M = \frac{\left(\sum_{k=1}^N \sum_{m=1}^N \sum_{i=1}^n \sum_{j=1}^n a_{kimj} x_{kimj} \right)}{N \times n} - \omega \left(\sum_{k=1}^N \sum_{m=1}^N q_{km} \right) \quad (1)$$

限制式：

$$\sum_{m=1}^N \sum_{j=1}^n x_{kimj} = 1, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, N \quad (2)$$

$$\sum_{k=1}^N \sum_{i=1}^n x_{kimj} = 1, \quad j = 1, 2, \dots, n, \quad m = 1, 2, \dots, N \quad (3)$$

$$H \times q_{km} \geq \sum_{i=1}^n \sum_{j=1}^n x_{kimj}, \quad k = 1, 2, \dots, N, \quad m = 1, 2, \dots, N \quad (4)$$

x_{kimj}, q_{km} : binary variables, $\forall k, \forall i, \forall m, \forall j$

公式 (1) 為目標函數，其中包含兩個子目標，子目標一是最大化基板配對的平均良率 Y ，子目標二是最小化 TFT 來源卡匣的移動總次數 M ， ω 代表子目標二的權重，是一個固定且極小的值，目的是使最佳解會一定滿足子目標一，然後從滿足子目標一的多組解中，找到滿足子目標二的最佳解。

公式 (2) 限制每一 TFT 基板只能配對給一片 CF 基板。公式 (3) 限制每一 CF 基板只能配對給一片 TFT 基板。公式 (4) 是用以間接求算 TFT 來源卡匣的移動次數。茲說明如下，若第 k 個 TFT 卡匣和第 m 個 CF 卡匣至少有一基板被配成對，則 $\sum_{i=1}^n \sum_{j=1}^n x_{kimj} > 0$ ，因為 H 為一極大的值，目標函數要最小化 q_{km} 的值，

q_{km} 又只有 0 和 1 兩種選擇，因此可確保 $q_{km}=1$ 。反之，若第 k 個 TFT 卡匣和第 m 個 CF 卡匣沒有任何基板被配成對，則 $\sum_{i=1}^n \sum_{j=1}^n x_{kimj} = 0$ ，同理可推論 $q_{km} = 0$ 。

3.3 模式分析比較

過去文獻[3]的基板良率配對在此簡稱為 SIP(simple integer programming)，本研究所發展的整數線性規劃模式稱為簡稱為 CIP (comprehensive integer programming)。與 SIP 相比較，CIP 的特點有二。第一是在目標函數加入子目標二，並且給予極小的權重。第二是加上公式 (4) 的限制式，以求算子目標二的值。

如果將本研究的模式刪除子目標二和公式(4)，就退化成過去文獻的基板良率配對模型，此退化模型因為具特殊結構，因此可用匈牙利法快速求解。雖然計算時間很快，但是只能提供一組解，在多組解的情況，無法將所有解求出。如前所述，此種退化模型只能確保基板配對良率最大化，但無法確保最少化上下排序機的次數。

本研究所構建的模型不具此特殊結構，無法以匈牙利法快速求解，本研究以商用軟體What'sBest [8]求解。此軟體的求解方法，是先將整數限制去除，求解出理論的目標上限值 (Z_{ub})，再以分枝界限法(branch and bound) [6] 來求最佳整數解。此種求解方法，當整數變數的數量增加時，潛在解 (potential solution) 的數量會呈指數 (exponentially) 增加，求最佳整數解的過程需花費很長的時間。

因此在求解此類問題時，通常設相對最佳解的容忍值 (relative optimality tolerance, r)， $0 \leq r \leq 1$ 。假設在分枝界限法的求解過程中，其最佳整數解的目標值為 Z_s 。當 $\frac{Z_{ub} - Z_s}{Z_{ub}} \leq r$ ，吾人即停止分枝界限法的程序，接受 Z_s 為最佳解。

3.4 設定值 ω 和 r 的推估

為了確保基板配對良率為最佳解，吾人需謹慎選擇子目標二的權重值 ω 和相

對誤差值 r 。 ω 的值的設定可分析如下，因為 Y 是平均良率，因此 $0 \leq Y \leq 1$ ， M 是來源卡匣的移動總次數，其移動上限最多為 N^2 次。為確保子目標一必定達成，吾人至少需設 $\omega = (1/10^5 \cdot N^2)$ ，亦即排序機的上下次數最多可省 N^2 次，其最多只能抵換 0.001%的良率。假設 $N = 10$ 則 $\omega = 10^{-7}$ 。

本研究推估 r 值的方法敘述如下，根據目標函數的定義， Z_s 和 Z_{ub} 可分別表達成 $Z_s = Y_s - \omega M_s$ ， $Z_{ub} = Y_{ub} - \omega M_{ub}$ 。若基板配對良率為最佳解，則 $Y_s = Y_{ub}$ 。吾人進而可推得下列公式。

$$\frac{\omega(M_s - M_{ub})}{Z_{ub}} \leq r$$

設若基板的配對良率為 100%，則 Z_{ub} 的值可設為 1，若基板上下排序機的次數離最佳解只差 10 次，則 $M_s - M_{ub} = 10$ 。因為 $\omega = 10^{-7}$ ，因此 r 值可設為 $r = p \cdot 10^{-6}$ ， p 值代表吾人所希望的求解品質， p 值越小求解品質越好，但是計算時間也越長。



3.5 啟發式演算法

上述的整數規劃模式 CIP，可求得最佳良率解的基板配對關係，以及在不同相對容忍值 (r) 下的卡匣移動次數， r 值訂的越小，解的品質越好，但是求解時間會增加。為改善 CIP 求解時間過長的問題，本研究發展一演算法來縮短求解時間。此演算法分成兩階段。第一階段是利用 CIP 求得一初始解(initial solution)。第二階段是發展一啟發式演算(稱為 Tune_Matrix)法來改善該初始解。

第一階段求初始解的步驟敘述如下。首先設定一求解時間上限(T_{ub})，利用 CIP 求得 T_{ub} 時間內的最佳解 $X = [X_{(k,i),(m,j)}]$ $1 \leq k \leq N, 1 \leq m \leq N, 1 \leq i \leq n, 1 \leq j \leq n$ ， X 代表TFT基板和CF基板的配對決策。參閱上述符號的定義，若 $\sum_{j=1}^n \sum_{i=1}^n X_{kimj} \geq 1$ ，則 $q_{km} = 1$ ；若 $\sum_{j=1}^n \sum_{i=1}^n X_{kimj} = 0$ ，則 $q_{km} = 0$ 。因此可據此求得一矩陣 $Q = [q_{km}]$ ， $1 \leq k \leq N, 1 \leq m \leq N$ 。吾人以一程序 $Q = Form_Cas(X)$ 來代表此轉換關係，並且定

義 $n(Q) = \sum_{k=1}^N \sum_{m=1}^N q_{km}$ ， $n(Q)$ 代表 Q 矩陣中元素值為 1 的總個數， $n(Q)$ 的值越大，代表此基板配對解要上下排序機的次數較多。

第二階段是發展一互換演算法 Tune_Matrix，來修改初始解 X 使 $n(Q)$ 的值降低，以得到較佳的解。吾人首先定義此互換演算法的相關符號，其次說明此演算法的基本構想，然後敘述此演算法的詳細步驟。

吾人定義一函數 f 使 $f(k, i) = r$ ， $f(m, j) = s$ 以簡化指標的表達；另定義一函數 g 使 $g(r) = k$ ， $g(s) = m$ 以找出每一基板所隸屬的卡匣。據此，良率參數矩陣可表達成 $A = [A_{r,s}]$ ，第一階段 CIP 所求得基板配對矩陣為 $X = [X_{r,s}]$ ，卡匣配對矩陣 $Q = \text{Form_Cas}(X) = [q_{km}]$ 。

茲將此互換演算法的基本構想說明如下。參閱圖 3.1， A 矩陣的任兩元素 $A_{r1,s1}$ 和 $A_{r2,s2}$ ，若滿足下列兩關係式 $A_{r1,s1} + A_{r2,s2} = A_{r1,s2} + A_{r2,s1}$ ，而且 $X_{r1,s1} = X_{r2,s2} = 1$ 則稱 $\{X_{r1,s1}, X_{r2,s2}\}$ 為一組「可換元素」(a pair of changeable elements)。若將 $X_{r1,s1}, X_{r2,s2}$ 兩元素的值按下列兩種情況(Case1 和 Case2)換置， X 的配對良率不會改變，但是其 $n(Q)$ 值可能會降低、增大、或不變，此演算法乃據此修正初始解 X 以降低 $n(Q)$ 。

$$\text{Case 1: } X_{r1,s1} \leftrightarrow X_{r1,s2} \quad \text{AND} \quad X_{r2,s2} \leftrightarrow X_{r2,s1}$$

$$\text{Case 2: } X_{r1,s1} \leftrightarrow X_{r2,s1} \quad \text{AND} \quad X_{r2,s2} \leftrightarrow X_{r1,s2}$$

在上述兩互換情況， $q_{g(r1),g(s1)}$ 、 $q_{g(r2),g(s2)}$ 、 $q_{g(r1),g(s2)}$ 、 $q_{g(r2),g(s1)}$ 值都會如下變化，互換前 $q_{g(r1),g(s1)} = 1$ ， $q_{g(r2),g(s2)} = 1$ ；互換後變成 $q_{g(r1),g(s2)} = 1$ ， $q_{g(r2),g(s1)} = 1$ 。吾人設此四個值的加總為 $n(q) = q_{g(r1),g(s1)} + q_{g(r2),g(s2)} + q_{g(r1),g(s2)} + q_{g(r2),g(s1)}$ 。上述兩種互換情況，若存在一種互換會使 $n(q)$ 值減少，則稱 $\{X_{r1,s1}, X_{r2,s2}\}$ 為一組「增值可換元素」(a pair of value-added changeable elements)。 $X_{r1,s1}, X_{r2,s2}$ 彼此互稱為增值對口元素(value-added counterpart)。

若 $\{X_{r1,s1}, X_{r2,s2}\}$ 是一組「增值可換元素」，可以有兩種互換作業(Case 1 和

Case2), 吾人選擇使 $n(q)$ 值降低最多的互換作業, 若兩互換作業所降低的 $n(q)$ 值一樣, 則選擇Case 1。上述互換的作業步驟吾人以一程序 $Procedure_Exchange(X_{r_1, s_1}, X_{r_2, s_2})$ 來描述。此程序互換後所降低的 $n(q)$ 值以 $Value_Exchange(X_{r_1, s_1}, X_{r_2, s_2}) = n(q_o) - n(q_f) = n(Q_o) - n(Q_f)$ 表示, 其中 $n(q_f)$ 代表互換後的 $n(q)$ 值, $n(q_o)$ 代表互換前的 $n(q)$ 值, $n(Q_f)$ 代表互換後的 $n(Q)$ 值, $n(Q_o)$ 代表互換前的 $n(Q)$ 值。若 $Value_Exchange(X_{r_1, s_1}, X_{r_2, s_2}) > 0$, 則 $X_{r_1, s_1}, X_{r_2, s_2}$ 為一組「增值可換元素」。Exchange($X_{r_1, s_1}, X_{r_2, s_2}$)的數值可能為2、1或0, 因為 $2 \leq n(q_o) \leq 4$ 而且 $2 \leq n(q_f) \leq 4$ 。

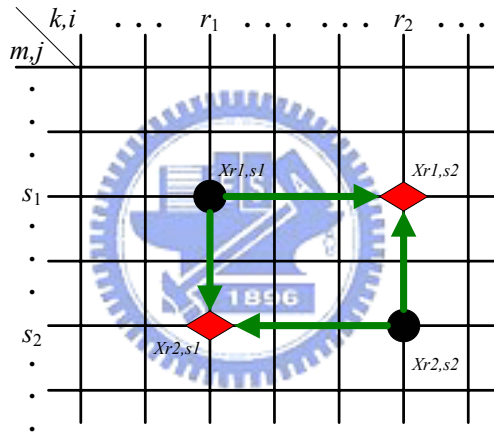


圖 3.1 CIP 求解後基板配對關係矩陣

茲將第二階段的演算法Tune_Matrix的執行步驟敘述如下, 假設第一階段已經利用CIP找出初始解 $X = [X_{rs}]$ 。

Procedure Tune_Matrix

Step 0: 初始設定 $r_0^* = 1, s_0^* = 1$

Step 1: 找出 X 矩陣中元素值為1的集合 $S = \{(r, s) \mid X_{rs} = 1, X_{rs} \in X\}$

Step 2: 對 S 中的每一元素 (r, s) , 找出其所有的「增值對口元素」

$$\psi(r, s) = \{(p, q) \mid \text{Value_Exchange}(X_{rs}, X_{pq}) > 0\}$$

計算 X_{rs} 與其「增值對口元素」互換時，所能降低 $n(Q)$ 最多的量，

$$d(r, s) = \max_{(p, q) \in \psi(r, s)} \{\text{Value_Exchange}(X_{rs}, X_{pq})\}$$

若有多個 X_{pq} 可選時，選 q 值最小，若仍存在多個，選 p 值最小。

Step 4: 決定 X 的互換元素

$$(r^*, s^*) = \arg \max_{(r, s) \in S} \{d(r, s)\}$$

$$\Delta q = d(r, s)$$

若 (r^*, s^*) 有多個，選 $|s^* - s_0^*|$ 最小，若仍有多個，選 $|r^* - r_0^*|$ 最小

$$(p^*, q^*) = \arg \max_{(p, q) \in \psi(r^*, s^*)} \{\text{Value_Exchange}(X_{r^*s^*}, X_{pq})\}$$

Step 5: 進行互換作業，更新 X 和 Q

IF $\Delta q > 0$ 進行互換作業，

Call Procedure_Exchange($X_{r^*s^*}, X_{p^*q^*}$) /*更新 X^* */

$Q = \text{Form_Cassette}(X)$ /*更新 Q^* */

$r_0^* = r^*$, $s_0^* = s^*$ /*記錄 X 目前互換的位置*/

Go to Step 1 /*繼續找增值互換*/

Else STOP 輸出 X /*找不到任何增值互換*/

上述求解基板配對的演算法可彙總說明如下，第一階段設定 T_{ub} 值用 CIP 求出初始解 X ，第二階段用Tune_Matrix來修正 X 。第一階段亦可考慮以SIP求解，但是所得解的品質可能不同，本研究將在下一章比較第一階段用SIP和CIP的求解結果。

第四章 實例驗證

本章針對前章所提的演算法進行實例驗證。首先敘述測試所用的軟硬體，其次介紹驗證的情境，最後分析測試結果。

4.1 軟體與硬體

此驗證所使用的電腦為 Pentium-IV 2.8GHz, 256MB DDRAM, 以 What'sBest 軟體 [8] 來求解第一階段的 CIP 模式，以 Visual Basic 語言來編寫來第二階段的演算法 Tune_Matrix。

What'sBest 軟體使用分支界限法 (B&B) 作為混整數規劃求解方法，在分支節點(Node)搜尋策略上提供了：深度優先 (Depth first)、最差優先 (Worst first) 及最佳優先 (Best first) 三種。本研究實驗發現解此問題使用深度優先的分支節點搜尋法，可以在較短的時間內得到最佳解，因此本研究使用深度優先作為分支界限的搜尋策略。

4.2 測試情境設計

本研究測試的案例 TFT 基板的平均良率為 90%，CF 基板的平均良率為 85%，兩者之良率分配皆為二項分配 (Bernoulli Distribution)。總共考慮 9 個測試情境：3 種卡匣數 ($N = 10, 15, 20$)，及 3 種面板數 ($p = 6, 12, 30$)，一卡匣可承載的基板數固定為 $n = 20$ 個，因此共有 $3 \times 3 \times 1 = 9$ 種測試情境。

其次，在 ω 和 r 的設定上， ω 設定為 1×10^{-7} ， r 的設定針對每一個 N 值設定 5 個不同的 r 值 (表 4.1)，亦即每一種情境，依照 r 值的不同有 5 個答案， r 值越小代表對求解品質的要求越高。

表 4.1 相對最佳容許值設定

卡匣數 (N)	Relative Optimality Tolerance (r)				
$N = 10$	0.0010%	0.0008%	0.0006%	0.0005%	0.0004%
$N = 15$	0.0020%	0.0018%	0.0016%	0.0014%	0.0012%
$N = 20$	0.0030%	0.0028%	0.0026%	0.0022%	0.0020%

4.3 CIP/SIP 求解品質與效率

上述 9 種測試情境，每個情境用 SIP 求解一組答案，用 CIP 求解 5 組不同 r 值的答案，因此每個情境共有 6 組答案，求解結果如表 4.2 所示。由該表可知，就基板配對平均良率 Y 而言，每一個情境的 6 個答案其配對良率皆一樣，亦即皆為最佳的配對良率。

就排序機所需的移動次數(M)而言，CIP 的求解績效比 SIP 為佳，但是求解時間較長，以第一個情境為例($N = 10, n = 20, p = 6$)，SIP 的移動次數 $M = 70$ ，CIP 最少的移動次數為 $M = 33$ ，移動次數減少了 37 次。然而計算時間增長許多，SIP 只需 22 秒，CIP 卻需要約 2 小時。

當 N 和 p 值增大時，CIP 所需的時間大幅增加，以第 9 個情境為例($N = 20, n = 20, p = 30$)，SIP 的解為 $M = 249$ ，約需 3 分鐘，CIP 最佳的解為 $M = 195$ ，移動次數可降低 54 次，但是計算時間約需 13.5 小時，計算時間實在太長，在實務上不太可行。

由表 4.2，吾人可知，當 N 值增大時或 p 值增加時，計算時間皆會增長。當 r 值越小時，求解的品質越好，但計算時間成指數成長。

4.4 Tune_Matrix 的求解品質與效率

在上述SIP和CIP所求的解當中，假設吾人設定 (T_{ub}) 為 1 小時，先利用 SIP/CIP 求出一初始解 X ，然後用 Tune_Matrix 來修正 X 。結果如表 4.3 所示，就排序機的所需的移動次數來看，第二階段 M2 皆可較第一階段 M1 進一步的減少移動次數，尤其當 N 增大和 P 減少時移動次數的減少更為顯著，以第 7 情境為例 ($N = 20, n = 20, p = 6$)，在 SIP 第二階段 M 值可以減少 77 次，CIP 第二階段 M 值可減少 97 次。在第二階段求時間 (T_2)，在 9 種情境下最大求解時間 ($N = 20, n = 20, p = 6$) 只須要在 21 秒內即可完成求解。

在 SIP 與 CIP 移動次數 (M) 和總執行時間 (T) 的比較上，在第二階段的移動次數上二者的差異相距不太，但在總執行時間上，SIP 均可在 4 分鐘以內完成第一及第二階段的求解時間，CIP 求解時間則須比 SIP 多出 1 個小時的求解時間。在此得到結論，我們可以 SIP 為第一階段的求解方法，再使用第二階段的 Tune_matrix 的方法，除了可以縮短求解時間外，也可以達到與 CIP 相同移動次數。

表 4.3 Tune_Matrix 結果分析

情境 (N, n, p)	第一階段 求解方法	第一階段 M 值 (M1)	第二階段 M 值 (M2)	可減少的 M 值 (M1-M2)	第一階段 求解時間 (T1)	第二階段 求解時間 (T2)	總求解時間 (T1+T2)
(10,20,6)	S1-1(SIP)	70	49	21	00:00:22	00:00:03	00:00:25
	S1-2(CIP)	48	44	4	01:00:00	00:00:04	01:00:04
(10,20,12)	S2-1(SIP)	86	66	20	00:00:23	00:00:03	00:00:26
	S2-2(CIP)	53	51	2	01:00:00	00:00:03	01:00:03
(10,20,30)	S3-1(SIP)	92	83	9	00:00:23	00:00:03	00:00:26
	S3-2(CIP)	70	70	0	01:00:00	00:00:02	01:00:02
(15,20,6)	S4-1(SIP)	125	76	49	00:00:59	00:00:10	00:01:09
	S4-2(CIP)	135	85	50	01:00:00	00:00:11	01:00:11
(15,20,12)	S5-1(SIP)	153	111	42	00:01:00	00:00:10	00:01:10
	S5-2(CIP)	143	102	41	01:00:00	00:00:11	01:00:11
(15,20,30)	S6-1(SIP)	168	149	19	00:01:02	00:00:09	00:01:11
	S6-2(CIP)	133	133	0	01:00:00	00:00:05	01:00:05
(20,20,6)	S7-1(SIP)	189	111	78	00:02:46	00:00:20	00:03:06
	S7-2(CIP)	230	125	105	01:00:00	00:00:21	01:00:21
(20,20,12)	S8-1(SIP)	224	144	80	00:02:45	00:00:19	00:03:04
	S8-2(CIP)	226	140	86	01:00:00	00:00:20	01:00:20
(20,20,30)	S9-1(SIP)	249	219	30	00:02:58	00:00:19	00:03:17
	S9-2(CIP)	210	203	7	01:00:00	00:00:14	01:00:14

表 4.2 CIP/SIP 規劃結果彙整

求解方法	卡匣數	基板數	面板數	r	基板配對良率	卡匣移動次數	求解時間
S1-1(SIP)	10	20	6	-	87.33%	70	00:00:22
S1-2(CIP)	10	20	6	8.0E-06	87.33%	68	00:16:31
S1-3(CIP)	10	20	6	7.0E-06	87.33%	61	00:45:12
S1-4(CIP)	10	20	6	6.0E-06	87.33%	51	00:59:19
S1-5(CIP)	10	20	6	5.0E-06	87.33%	44	01:01:18
S1-6(CIP)	10	20	6	4.0E-06	87.33%	33	01:51:40
S2-1(SIP)	10	20	12	-	85.71%	86	00:00:23
S2-2(CIP)	10	20	12	8.0E-06	85.71%	66	00:08:13
S2-3(CIP)	10	20	12	7.0E-06	85.71%	59	00:10:57
S2-4(CIP)	10	20	12	6.0E-06	85.71%	50	01:12:33
S2-5(CIP)	10	20	12	5.0E-06	85.71%	43	01:32:17
S2-6(CIP)	10	20	12	4.0E-06	85.71%	41	15:54:08
S3-1(SIP)	10	20	30	-	83.28%	90	00:00:23
S3-2(CIP)	10	20	30	8.0E-06	83.28%	72	00:07:08
S3-3(CIP)	10	20	30	7.0E-06	83.28%	72	00:25:57
S3-4(CIP)	10	20	30	6.0E-06	83.28%	69	01:10:10
S3-5(CIP)	10	20	30	5.0E-06	83.28%	69	01:50:46
S3-6(CIP)	10	20	30	4.0E-06	83.28%	68	06:58:28
S4-1(SIP)	15	20	6	-	87.17%	125	00:00:59
S4-2(CIP)	15	20	6	2.0E-05	87.17%	165	00:12:54
S4-3(CIP)	15	20	6	1.8E-05	87.17%	157	00:30:26
S4-4(CIP)	15	20	6	1.6E-05	87.17%	137	00:53:59
S4-5(CIP)	15	20	6	1.4E-05	87.17%	119	01:19:37
S4-6(CIP)	15	20	6	1.2E-05	87.17%	100	02:54:15
S5-1(SIP)	15	20	12	-	85.58%	153	00:01:00
S5-2(CIP)	15	20	12	2.0E-05	85.58%	168	00:11:25
S5-3(CIP)	15	20	12	1.8E-05	85.58%	148	00:19:35
S5-4(CIP)	15	20	12	1.6E-05	85.58%	130	00:28:24
S5-5(CIP)	15	20	12	1.4E-05	85.58%	119	04:33:54
S5-6(CIP)	15	20	12	1.2E-05	85.58%	99	05:12:31
S6-1(SIP)	15	20	30	-	83.69%	168	00:01:02
S6-2(CIP)	15	20	30	2.0E-05	83.69%	169	00:15:42
S6-3(CIP)	15	20	30	1.8E-05	83.69%	151	00:17:54
S6-4(CIP)	15	20	30	1.6E-05	83.69%	137	01:28:24
S6-5(CIP)	15	20	30	1.4E-05	83.69%	135	03:30:27
S6-6(CIP)	15	20	30	1.2E-05	83.69%	131	19:26:47
S7-1(SIP)	20	20	6	-	85.79%	189	00:02:46
S7-2(CIP)	20	20	6	3.0E-05	85.79%	242	00:28:27
S7-3(CIP)	20	20	6	2.8E-05	85.79%	233	00:35:33
S7-4(CIP)	20	20	6	2.6E-05	85.79%	223	01:25:25
S7-5(CIP)	20	20	6	2.2E-05	85.79%	180	04:40:47
S7-6(CIP)	20	20	6	2.0E-05	85.79%	169	17:19:57
S8-1(SIP)	20	20	12	-	84.96%	224	00:02:45
S8-2(CIP)	20	20	12	3.0E-05	84.96%	247	00:33:27
S8-3(CIP)	20	20	12	2.8E-05	84.96%	232	00:58:05
S8-4(CIP)	20	20	12	2.6E-05	84.96%	218	01:05:26
S8-5(CIP)	20	20	12	2.2E-05	84.96%	181	01:50:59
S8-6(CIP)	20	20	12	2.0E-05	84.96%	170	05:15:36
S9-1(SIP)	20	20	30	-	83.28%	249	00:02:58
S9-2(CIP)	20	20	30	3.0E-05	83.43%	249	00:37:58
S9-3(CIP)	20	20	30	2.8E-05	83.43%	234	00:43:20
S9-4(CIP)	20	20	30	2.6E-05	83.43%	215	00:52:29
S9-5(CIP)	20	20	30	2.2E-05	83.43%	201	05:10:45
S9-6(CIP)	20	20	30	2.0E-05	83.43%	195	13:21:09

第五章 結論與未來發展方向

5.1 結論

本論文針對 TFT-LCD 的基板配對製程，發展一配對決策的演算法，此演算法是由兩模組所構成，第一模組稱為 CIP，乃是一整數線性規劃模式，產生一初始解。第二模組稱為 Tune_Matrix，是一啟發式演算法，目的是修正第一階段所產生的初始解。

此演算法可確保最佳的基板配對良率，並有效降低排序機的搬運次數，同時此演算法的計算速度也很快。不論就求解品質或是就求解時間而言，都比過去的相關研究為佳。



5.2 未來研究方向

本論文方法在排序機上下貨的卡匣移動分析，假設在單一機器手臂及單一的終端埠及來源埠，僅考慮在找出較少的卡匣移動次數，並未考量卡匣上下貨時可以透過派工法來進一步減少卡匣移動次數，及當多個機器手臂及多個終端埠及來源埠時，如何決定卡匣派工的邏輯。

本論文未來可延伸的研究方向上，未來可分析排序機上合理的機器手臂、終端埠及來源埠的數量，並發展卡匣上下貨時移動順序的派工法，目的決定如何指派 TFT 卡匣到排序機，使一個 CF 卡匣完成排序時，能保留下一個 CF 卡匣所對應 TFT 卡匣，減少 TFT 卡匣移動時間，進一步的減少製程週期時間。

參考文獻

中文部份

- [1] 李秀玉 (1999) 應用賽局理論分析我國薄膜電晶體液晶顯示器產業之競爭策略，交通大學科技管理研究所碩士論文
- [2] 李俊昇 (2003) 液晶面板組裝廠批量製程派工法則之設計，交通大學工業工程研究所碩士論文
- [3] 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
- [4] 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，交通大學工業工程研究所碩士論文
- [5] 謝仲為 (2002) 先進規劃與排程系統應用於 TFT-LCD 產業之研究，東海大學工業工程研究所碩士論文



英文部分

- [6] Hiller, FS, and Lieberman, GJ (1990) Introduction to Operations Research 5th ed. McGraw-Hill New York
- [7] Jeong, B, Kim, SW and Lee, YJ (2001) An assembly scheduler for TFT LCD manufacturing. Computers and Industrial, 41(1):37-58
- [8] Lindo system, Inc.(2002) What'sBest Manual-Option:Integer Solver. <http://www.lindo.com/>
- [9] Shin, HJ and Leon,VJ (2004) Scheduling with product family set-up time: an application in TFT LCD manufacturing. International Journal of Production Research, 42(20) : 4235-4248
- [10] Toba H.(2004) A Tight Flow Control for Job-Shop Fabrication Lines With Finite Buffer. *IEEE Transactions on Automation Science and Engineering* : Accepted for future publication, pp(99):1-6