

東 海 大 學

工業工程與經營資訊學系

碩士論文

應用基因演算法於多階平行機台排程  
-以晶矽太陽能電池為例

研 究 生：莊閔智

指 導 教 授：鄭辰仰 博士

王立志 博士

中 華 民 國 一〇〇 年 七 月

**A genetic algorithm for multi-stage parallel machines  
scheduling problems – A case study for solar Cell industry**

By  
Min-Chih Chuang

Advisor: Dr. Chen-Yang Cheng  
Dr. Li-Chih Wang

A Thesis  
Submitted to the Institute of Industrial Engineering and Enterprise  
Information at Tunghai University  
in Partial Fulfillment of the Requirements  
for the Degree of Master of Science  
in  
Industrial Engineering and Enterprise Information

July 2011  
Taichung , Taiwan , Republic of China

# 應用基因演算法於多階平行機台排程 - 以晶矽太陽能電池為例

學生：莊閔智

指導教授：鄭辰仰 博士

王立志 博士

東海大學工業工程與經營資訊學系

## 摘要

近幾年混合流程型生產(Hybrid Flow Shop; HFS)的研究已相當成熟，但這些研究有一個共同的假設，即一張訂單只分配一台機台的模式，無法解決每張訂單由多台機台同時加工的產業應用。因此，學者提出平行機台資源工作(Multiprocessor task)的排程架構，在混合流程型生產的生產環境下，需先規劃出各訂單的機台資源配置方式，透過各種排程演算法獲取此機台資源配置下的最佳生產順序，以達到較好的生產排程結果，但機台資源的配置與訂單的生產順序是互相影響的，在已決定機台資源配置的情況下調整生產順序，無法找到最佳的生產排程組合。所以本研究將探討訂單分割之混合流程型生產系統的問題，以最小化之最大完工時間為目標，在多階製程中，每道製程擁有多台完全平行機台，該如何將訂單分割為多張工單，並規劃工單之生產順序與該工單應配置到哪些機台上，以縮短最大完工時間。而晶矽太陽能電池的生產環境，則屬於混合流程型的生產模式，因此針對太陽能電池的製程之四大特性，平行機台加工、專用機台、獨立整備時間與相依整備時間，提出一基因演算法，欲解決訂單如何分割於混合流程型生產系統的問題。最後本研究將與最佳化方式求解一混整數線性規劃的方式進行比較，本研究方法不論在小規模或大規模的問題中，其求解品質與績效，均有良好的表現。另外，也透過實際案例之比較，其結果也優於案例公司之生產排程方法。因此，本研究針對太陽能產業特性所提出之求解方法，可提供業界於進行訂單分割與機台排程時的決策參考。

**關鍵字詞：**太陽能電池、混合流程式、生產排程、平行加工、基因演算法

# **A genetic algorithm for multi-stage parallel machines scheduling problems – A case study for solar Cell industry**

Student: Min-Chih Chuang

Advisor: Dr. Chen-Yang Cheng

Dr. Li-Chih Wang

Department of Industrial Engineering and Enterprise Information  
Tunghai University

## **ABSTRACT**

In recent years, a lot of outstanding research on hybrid flow shop, HFS has been done. There is one common hypothesis among the research, that is one-job-on-one-machine pattern. Under the hypothesis, one order cannot be carried out by numerous machines simultaneously. Therefore, Multiprocessor task and scheduling were advocated by scholars. When hybrid flow shops is adopted, the allocation of machine resources for each order should be scheduled in advance. By conducting several types of algorithms of scheduling to generate the allocation of machine resources, the best production sequence can be created. In that case, better performance can be achieved. However, allocation of machine resources and production sequence are highly interactive. If allocation of machine resources has been determined, adjustment on production sequence is unable to contribute to the best production scheduling. This study is going to explore the issue about hybrid flow shop for order splitting. The final purpose is minimum makespan. In multi-stage manufacturing process, several identical parallel machines are involved in each manufacturing process. In order to shorten the makespan, the method of splitting order into numerous manufacture orders, scheduling production sequence for manufacture orders, and allocate manufacture orders to assigned machines are concerned. In terms of solar cell production, the type of this production is classified as hybrid flow shop. To find solution to order splitting on hybrid flow shop, four characteristics of solar cell production will be discussed by genetic algorithms. At the end of this study, the result of this research and mixed integer linear programming from optimization method will be compared. The methodology of this study can be applied to clarify both quality problem and effects for small problem size as well as large problem size. In addition, the result is also proved to be more effective than case study's production scheduling. The finding of this study can provide a better solution for order splitting and machine scheduling. It also can be referred by solar cell industry for strategy making.

**Keywords: Solar cells, Production scheduling, Parallel-machine scheduling, Genetic algorithm**

## 致謝

首先誠摯的感謝指導教授鄭辰仰博士及王立志博士，兩位老師悉心的教導使我得以一窺生產排程領域的深奧，不時的討論並指點我正確的方向，使我在這些年中獲益匪淺。老師對學問的嚴謹更是我輩學習的典範。

本論文的完成另外亦得感謝的大力協助。因為有你的體諒及幫忙，使得本論文能夠更完整而嚴謹。

兩年裡的日子，實驗室裡共同的生活點滴，學術上的討論、言不及義的閒扯、讓人又愛又怕的宵夜、趕作業的革命情感、因為睡太晚而遮遮掩掩閃進實驗室……，感謝眾位學長姐、同學、學弟妹的共同砥礪，你/妳們的陪伴讓兩年的研究生活變得絢麗多彩。

感謝淑芬學姊不厭其煩的指出我研究中的缺失，且總能在我迷惘時為我解惑，也感謝鈺勛、章昱、晉璋、閔雄、蕙菁同學的幫忙，恭喜我們順利走過這兩年。實驗室的阿楷、小蔡、中俊、阿卜、悅清當然也不能忘記，你/妳們的幫忙及搞笑我銘感在心。

女朋友玫羽在背後的默默支持更是我前進的動力，沒有玫羽的體諒、包容，相信這兩年的生活將是很不一樣的光景。

最後，謹以此文獻給我摯愛的雙親。

莊閔智 謹致於  
東海大學工業工程與經營資訊學系  
民國一〇〇年七月

# 目錄

摘要.....	i
ABSTRACT.....	ii
致謝.....	iii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	3
1.3 研究方法與步驟.....	3
1.4 論文架構.....	4
第二章 文獻探討.....	5
2.1 混合流程型生產(HYBRID FLOW SHOP).....	5
2.1.1 啟發式法則 (Heuristics) .....	7
2.1.2 搜尋演算法 (Metaheuristics) .....	8
2.1.3 混合法 (Hybrid approaches) .....	8
2.1.4 小結.....	9
2.2 基因演算法.....	11
2.2.1 基因演算法介紹.....	11
2.2.2 基因演算法之方法與流程.....	11
第三章 多階平行機台之基因演算法.....	14
3.1 晶矽太陽能電池.....	14
3.2 製程特性與問題定義.....	17
3.2.1 晶矽太陽能電池的生產製程.....	17
3.2.2 晶矽太陽能電池之製程特性.....	22
3.3 問題描述.....	26
3.4 數學模式.....	27
3.4.1 假設條件.....	27
3.4.2 符號定義.....	28
3.4.3 混整數線性規劃模式.....	29
3.5 多階平行機台之基因演算法說明.....	33
3.5.1 染色體設計與解碼.....	34
3.5.2 初始化染色體族群.....	35

3.5.3 計算染色體之適應函數值.....	36
3.5.3.1 適應值計算.....	36
3.5.4 選擇及複製.....	45
3.5.5 交配.....	46
3.5.6 突變.....	47
3.5.7 終止條件.....	48
第四章 績效評估.....	49
4.1 實驗方式與環境建構.....	49
4.1.1 實驗環境.....	49
4.1.2 基因演算法參數.....	51
4.2 本研究之模式與混整數線性規劃模型比較.....	51
4.2.1 結果比較.....	52
4.3 實驗設計.....	54
4.3.1 實驗規劃說明.....	54
4.3.2 Makespan 數據分析.....	55
4.3.3 CPU Time 數據分析.....	56
4.4 本研究之模式與現況方法比較.....	57
4.4.1 現行啟發式規劃說明.....	58
4.4.2 結果比較.....	60
4.5 小結.....	62
第五章 結論與未來方向.....	63
5.1 結論.....	63
5.2 未來方向.....	63
參考文獻.....	64
附錄一：MAKESPAN 之實驗數據表.....	67
附錄二：CPU TIME 之實驗數據表.....	68
附錄三：實際案例之甘特圖.....	69

## 圖目錄

圖 1.1 單一訂單分配至多機台示意圖（機台配置已知） .....	2
圖 1.2 單一訂單分配至多機台示意圖（機台配置未知） .....	2
圖 1.3 本論文之研究架構.....	4
圖 2.1 HFS 標準架構圖 .....	6
圖 2.2 HFS 生產系統因素之分類圖（本研究整理） .....	7
圖 2.3 基因演算法之邏輯.....	13
圖 3.1 本國太陽能產業鏈結構.....	15
圖 3.2 太陽能電池種類（本研究整理） .....	16
圖 3.3 太陽能電池製程流程與製程特性.....	16
圖 3.4 太陽能電池 BOM .....	17
圖 3.5 太陽能電池種類比例估計.....	17
圖 3.6 晶矽太陽能電池製造流程.....	18
圖 3.7 蝕刻製程（長生能源） .....	18
圖 3.8 磷擴散爐（長生能源） .....	19
圖 3.9 磷玻璃蝕刻機（長生能源） .....	19
圖 3.10 抗反射層沉積（長生能源） .....	20
圖 3.11 網印（長生能源） .....	20
圖 3.12 快速燒結（長生能源） .....	21
圖 3.13 雷射晶邊絕緣（長生能源） .....	21
圖 3.14 效能測試分類（長生能源） .....	22
圖 3.15 平行機台示意圖.....	23
圖 3.16 蝕刻製程專用機台.....	24
圖 3.17 網印製程圖.....	25
圖 3.18 太陽能電池測試分類圖.....	26
圖 3.19 分割訂單與機台選擇示意圖.....	27
圖 3.20 基因演算法流程圖.....	34
圖 3.21 本研究所採用染色體之編碼方式.....	35
圖 3.22 專用機台限制.....	36
圖 3.23 適應值模擬之案例.....	36
圖 3.24 第一順位之訂單在 Stage 1 之計算適應值甘特圖.....	38

圖 3.25 所以訂單在 Stage 1 之適應值計算甘特圖.....	39
圖 3.26 訂單最早完工優先法則.....	40
圖 3.27 所有訂單在 Stage 2 之適應值計算甘特圖.....	41
圖 3.28 所有訂單在 Stage 5 前之適應值計算甘特圖.....	41
圖 3.29 所有訂單在 Stage 5 之適應值計算甘特圖.....	43
圖 3.30 所有訂單在 Stage 6 之適應值計算甘特圖.....	45
圖 3.31 位置交配圖.....	46
圖 3.32 雙點交配圖.....	46
圖 3.33 鄰近雙點突變.....	47
圖 3.34 位元突變.....	47
圖 4.1 兩種方法的求解時間趨勢.....	53
圖 4.2 Problem1(3, 3, 4)之甘特圖 .....	54
圖 4.3 Makespan ANOVA 分析之柏拉圖 .....	56
圖 4.4 CPU Time ANOVA 分析之柏拉圖 .....	57
圖 4.5 生產規劃流程示意圖.....	59
圖 4.6 Priority-based 啟發式規劃示意圖.....	59
圖 4.7 現行機台配置之甘特圖.....	60
圖 4.8 Real case problem 之收斂圖 .....	61

## 表目錄

表 2.1 相關文獻整理.....	10
表 3.1 適應值案例之各細部參數.....	37
表 4.1 產品類型資訊.....	49
表 4.2 實驗問題.....	50
表 4.3 基因演算法參數設定.....	51
表 4.4 本研究模式與 CPLEX 之結果與差距.....	52
表 4.5 本研究模式與 CPLEX 之求解時間.....	53
表 4.6 控制因子及水準.....	55
表 4.7 實驗設計表.....	55
表 4.8 Makespan 之 ANOVA 分析表.....	56
表 4.9 CPU Time 之 ANOVA 分析表.....	57
表 4.10 廠商案例實際需求資料.....	58
表 4.11 本研究模式與現況方法之結果.....	61
表 4.12 本研究模式與現況方法之結果差距.....	61
表 4.13 本研究模式於 Real case problem 下之求解時間.....	62

# 第一章 緒論

## 1.1 研究背景與動機

隨著製造業的競爭面臨日趨激烈的情況下，傳統的生產環境，如零工式(Job-Shop)生產與流程式(Flow-Shop)生產，已稍嫌不足，且因許多製造商因應顧客訂單的龐大需求，以至於在生產線中增加機台的數量，呈現出每道製程中擁有多機台的生產環境，因此在這新的製造環境中，Salvador (1973)學者提出了混合流程型生產(Hybrid flow shop; HFS)，它是由平行機台(Parallel machine scheduling; PMS)與流程式生產排程(Flow shop scheduling; FSS)所組成的(Ribas, Leisten, & Framin, 2010)，PMS 問題為如何分配訂單到機台上，而 FSS 問題在於如何決定訂單生產順序，由於兩種生產特性的結合，對於在排程問題上，複雜度也相對增加，Ribas (2010)等學者認為此種類的生產環境中，不但要考慮訂單的加工順序，同時還須考慮分配訂單至機台的派工問題。

Chen 和 Lee (1999)學者指出，在過去幾年中，HFS 的研究已相當成熟，但這些研究有一個共同的假設，即一張訂單只分配一台機台(One-job-on-one-machine pattern)的模式，此研究無法解決每張訂單由多台機台同時加工的產業應用。因此，他提出了所謂平行機台資源工作(Multiprocessor task)的排程架構(圖 1.1)，同樣在 HFS 的生產環境下，需先規劃出各訂單的機台資源配置數量，再透過各種排程演算法，獲取在這樣的機台資源配置下的最佳生產順序，以達到較好的生產排程結果(Engin, Ceran, & Yilmaz, 2011)，但機台資源的配置與訂單的生產順序，兩者是互相影響的，在已決定機台資源配置的情況下，再調整生產順序，並無法找到最佳的生產排程組合。

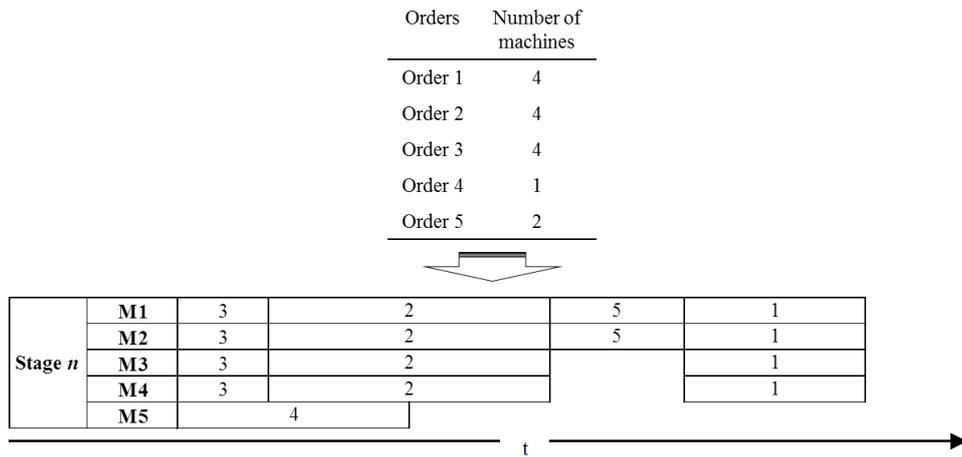


圖 1.1 單一訂單分配至多機台示意圖（機台配置已知）

現今的生產環境一直在變化，新興產業也不斷的崛起，衍生出更多或更複雜的生產環境，其中，近年來話題不斷的太陽能電池產業，便是如此。從學術研究的角度，係屬於 HFS 環境中平行機台資源工作的生產架構，但並非完全相同，因在進行生產排程時，生管人員無法清楚的知道該訂單的機台資源配置方式，如圖 1.2 所示，每張訂單皆可安排該製程中最大的機台數量，且安排之機台數量可動態調整，以能在最短的時間內完成該訂單；而訂單生產順序也只採用較傳統的派工法則進行安排，所以在機台資源配置與訂單生產順序都是未知的情況下，讓整個生產排程複雜性增加不少，過去研究中也較少著墨。因此，本研究將從案例公司，探討晶矽太陽能電池生產排程的特性，如平行機台、專用機台、獨立整備、相依整備等，強調此案例為 HFS 生產環境的延伸，提出解決的方法，並適當進行評估績效，使其方法能解決這類的問題。

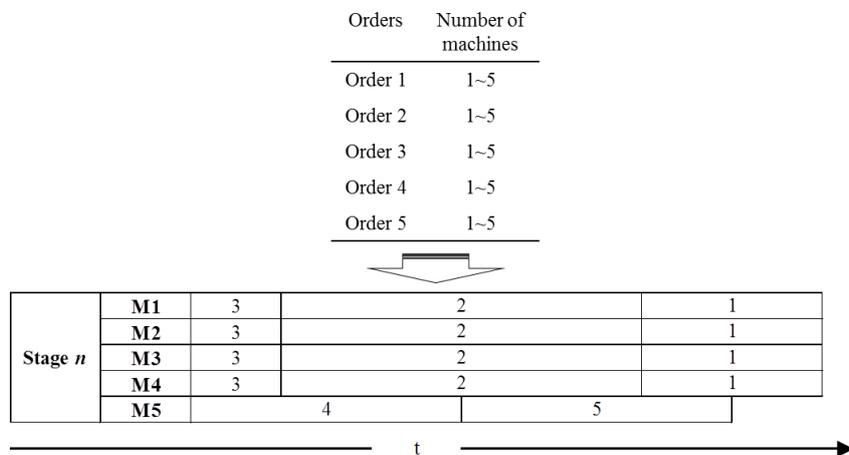


圖 1.2 單一訂單分配至多機台示意圖（機台配置未知）

## 1.2 研究目的

隨著產業的發展，製造環境不斷改變的情況下，過去定義 HFS 環境的生產型態，已無法完全符合新型態的生產模式，便無法透過先前的研究，來解決新的問題，需要近一步的調整與探討才能解決。HFS 問題從一開始標準的一訂單搭配一機台的情況，至一訂單規劃多機台的環境，但前提必須為各訂單之機台資源為已知的情況，但現在所面臨的問題是，訂單生產順序與資源機台配置皆為未知，增加了生產排程的難度，如晶矽太陽能電池製造廠，即面臨這樣的問題。

我國積極投入的太陽能電池產業，目前除了提升太陽能電池的轉換效率外，對於電池製造廠的排程規劃，也是一重要課題。產品需求市場逐漸擴大，必須提高生產能力，縮短交貨時間，以滿足顧客訂單需求，所以太陽電池製造廠商重視生產效率與訂單達交率。因此，訂單生產順序的安排、製造現場機台的配置與機台資源的選擇，為晶矽太陽能電池製造商在執行排程規劃時所面臨到的問題。因此，本研究將考量晶矽太陽能電池製造業生產規劃的相關特性，利用基因演算法(Genetic algorithm; GA)規劃各訂單的生產順序與機台資源的配置，在已知的訂單與製程相關資訊，如訂單需求量、產品型態、現場機台數與製程數等，求解出排程的配置方式，使規劃後之最大完工時間(Makespan)最小。研究目的如下：

1. 歸納晶矽太陽能電池製造業為多階製程與平行機台資源之生產模式。
2. 探討晶矽太陽能電池於製造生產時的製程特性。
3. 考量晶矽太陽能電池製造業之製程特性，提出生產排程之規劃方法，利用基因演算法求得目標為最小 Makespan 的生產排程。

## 1.3 研究方法與步驟

本研究針對過去的 HFS 環境的架構，提出一延伸此架構下的新議題，提出一符合這議題特性的實際案例，晶矽太陽能電池製造業，來加以印證，在考量晶矽太陽能電池的製程特性，應用基因演算法求解。首先，本研究將探討過去在 HFS 研究中，關於訂單順序與平行機台資源皆為變數下的相關文獻，以作為提出此生產規劃演算法的依據，進而提出 Makespan 最小下

之基因演算法，透過詳細步驟說明演算法細部之流程與邏輯。最後，本研究所提生產規劃之基因演算法將與使用混整數線性規劃(Mixed integer linear programming; MILP)求解的方式進行比較與效益分析。

## 1.4 論文架構

本研究共分為五章如圖 1.3 所示，第一章緒論說明研究背景與動機及研究目的，並概略說明研究方法。第二章文獻探討則針對 HFS 環境下，生產規劃及基因演算法的相關文獻進行蒐集與探討。第三章簡單說明晶矽太陽能電池製造業的生產流程與特性，並描述問題，接著針對所提出 Makespan 最小化之基因演算法說明，並詳細介紹演算過程。第四章透過與混整數線性規劃求出之最佳解比較，評估演算法績效。第五章總結本研究所獲致的成果，並提出後續研究的建議。

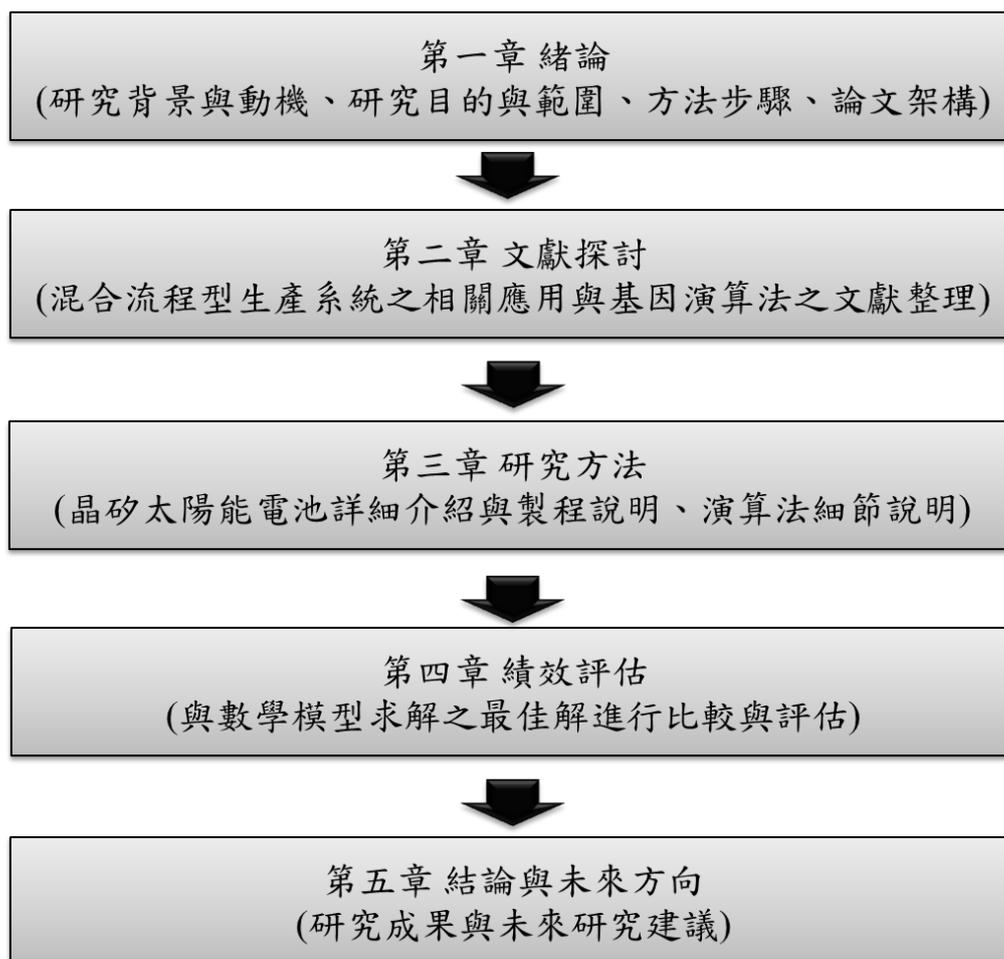


圖 1.3 本論文之研究架構

## 第二章 文獻探討

### 2.1 混合流程型生產(Hybrid Flow Shop)

混合流程型生產在現實中是一種常見的製造環境，它的定義為有  $n$  個訂單在  $m$  製程階段中進行加工，且有一個目標函式來優化整個生產排程。而在 HFS 的排程問題上，可由兩個一般性的排程問題所組成，為分別為平行機台與流程式生產排程，PMS 問題為如何分配訂單到機台上，而 FSS 問題在於如何決定訂單生產順序，其又依照機台特性，Allahverdi (1994)等學者依據個別機器的生產效率或型態可分為三類：

1. 完全平行機台(Identical parallel machines)：同一工作在各機台的作業時間皆相同。
2. 等效平行機台(Uniform parallel machines)：同一工作在各機台的作業時間一機台的效率而有所不同，且成比例。
3. 非等效平行機台(Unrelated parallel machines)：同一工作在各機台的作業時間依機器的效率而有不同，且不成比例。

而標準的 HFS 生產系統如圖 2.1 所見，它擁有以下常見的特徵(Ruiz & Vázquez-Rodríguez, 2010)：

1. 製程階段  $m$  至少兩階。
2. 所有的製程階段  $k$  中，至少有一道製程的平行機台數必須大於或等於二。
3. 所有的訂單加工的順序皆相同，如：階段 1、階段 2、...、階段  $m$ 。
4. 所有的訂單與加工機器可以在時間為零時開始規劃。
5. 每台機器在同一時間內最多只加工一件訂單。
6. 每件訂單在同一時間內只能在一部機器上加工。
7. 設置時間可以忽略及各製程站間的緩衝無限制。

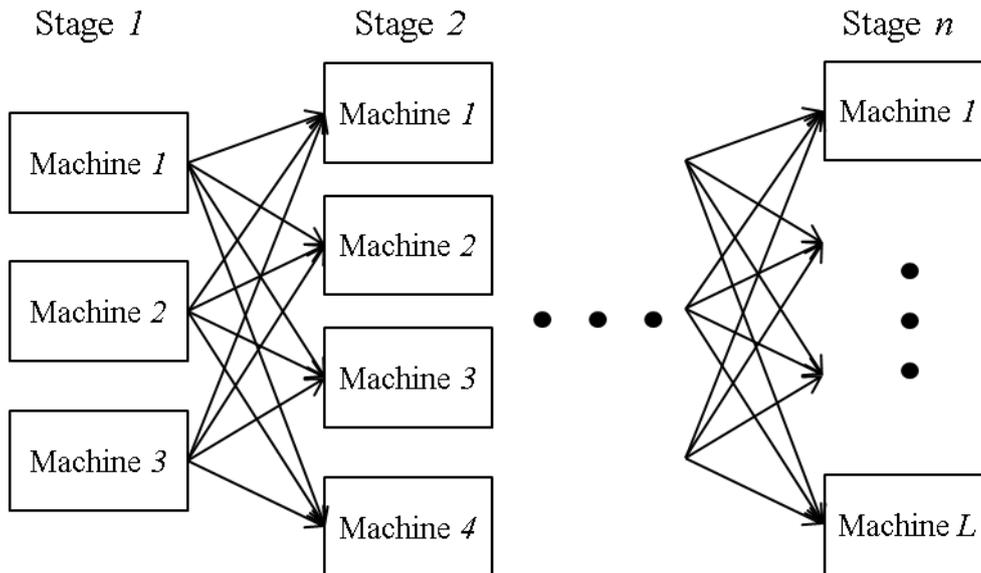


圖 2.1 HFS 標準架構圖

而近年來，許多專家學者致力於研究 HFS 生產排程的問題，雖然此問題可使用最佳化方式求解，但 Garey 和 Johnson (1979)兩位學者的研究中，已經證明在 HFS 下求取最小化的 makespan 排程，即是一個 *NP-Complete* 的問題，因此當想要求解的問題變大時，最佳化方法將難以求得最佳解，或是需要花費龐大的時間，因此許多專家學者紛紛提出啟發式搜尋演算法來解決這類的問題。因此，為了縮短解題的時間，啟發式演算法與派工法則相繼被提出，有採取較簡單啟發式演算法則，應用於兩階段的混合流程型生產環境中(Gupta et al., 1997)；亦有採取啟發式搜尋演算法的方式，如禁忌搜尋(TabuSearch; TS)(Wardono & Fathi, 2004)、模擬退火法(Simulated Annealing; SA)(Jina, Yanga, & Itob, 2006)、類神經網路(Neural Networks; NN)(Wang, Jacob, & Rolland, 2003)、基因演算法(Bertel & Billaut, 2004)與螞蟻演算法(Ant Colony System; ACS)(Alaykýran et al., 2007)，來解決混合流程型製造系統排程問題。此外，關於 HFS 生產系統中，本研究根據過去的文獻，整理出可依照生產機台的特性與訂單的限制來區分，如圖 2.2 所示，機台特性如前一章節所敘述的共為三種，而訂單限制可區分為六種，包含不完全作業、批量分割、訂單優先權、搬運時間、設置時間與回流，這些特性是產業中的關鍵影響因素，影響整個排程的結果，這些特性將於後續的文獻中探討。

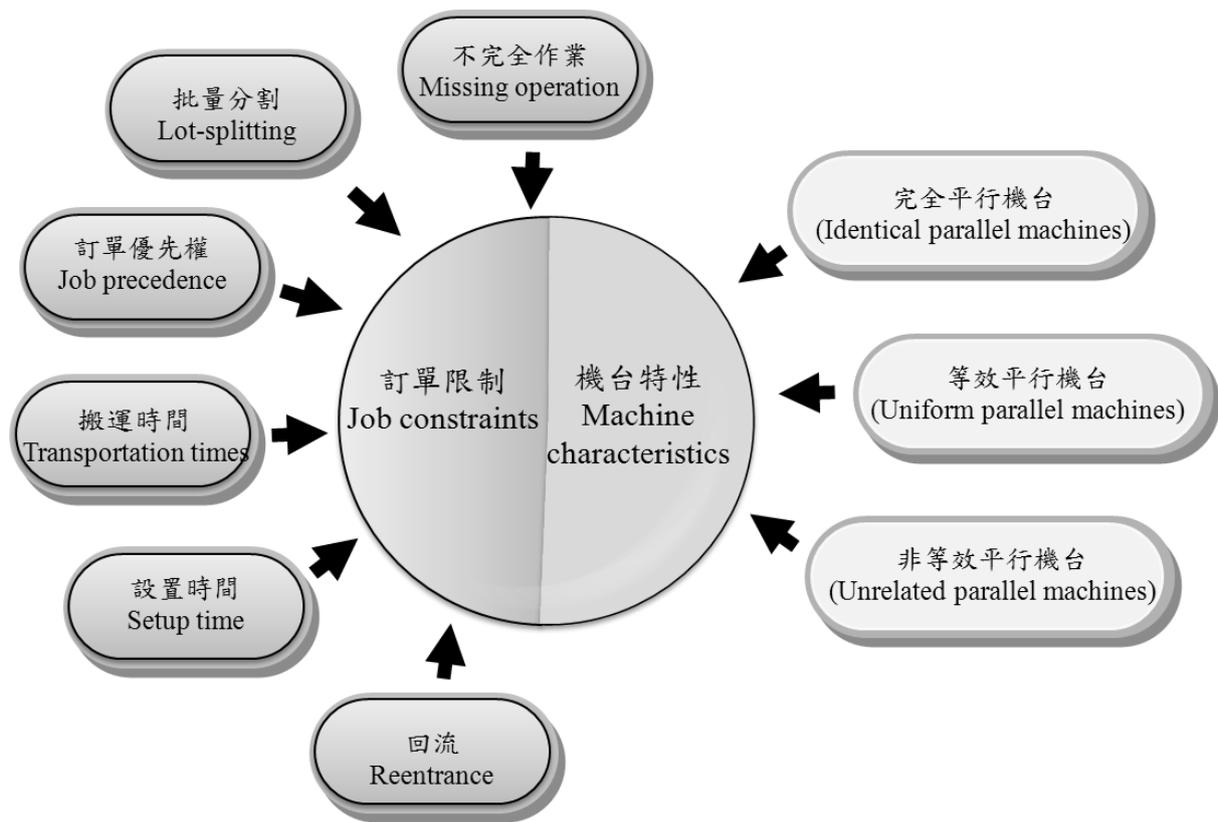


圖 2.2 HFS 生產系統因素之分類圖（本研究整理）

### 2.1.1 啟發式法則 (Heuristics)

Kim (1997)等學者在兩階段的 HFS 排程問題中，在完全平行機台的環境中，考慮移轉批量(Transfer batch)的參數，以派工法則的方法求得 Makespan 最小化。Botta-Genoulaz (2000)學者提出了六個新的啟發式演算法來解決 HFS 的問題，以最大延遲時間最小化為目標，並考慮了運輸時間與訂單的優先順序。Tseng (2008)等學者研究在特定的兩階段的 HFS 問題中，其中有些工作可以略過第一階段，並提出一種啟發式的方法，解決最大完成時間最小化之問題。Low (2008)等學者探討了二階段的非等效平行機台，第一階段受限於機台特性，應用啟發式結合了四種派工規則與強生法則 (Johnson's rule)，以最大完工時間最小化為目標。

### 2.1.2 搜尋演算法 (Metaheuristics)

Jin (2002)等學者使用基因演算法在軟性電路板的排程上，以三階段的HFS問題與最小化Makespan為目標函式，解決了排程問題。Bertel和Billaut (2004)兩位學者提出了一個三階段的HFS生產系統，並使用基因演算法，但使用貪婪演算法(Greedy algorithm)結合派工法則來產生初始族群，並考慮作業再回流的特性。Oğuz和Ercan (2005)兩位學者在HFS生產環境中，對於Makespan最小化的目標中，以基因演算法來求解，並提出新的交配(Crossover)機制，並比較相同的問題，結果優於禁忌搜尋。Yang (2007)等學者應用基因演算法解決在積層陶瓷電容器製造廠的生產排程問題，並考慮順序獨立的設置時間。Belkadi (2006)等學者使用平行基因演算法(Parallel genetic algorithm; PGA)解決HFS生產排程問題，以Makespan最小化為目標，並提出HFS系統類似於一個遷移平行遺傳演算法(Migration parallel genetic algorithm; PGA\_MIG)，可應用遺傳的多樣性和繁殖機制的選擇更符合自然，這種方法提高了計算的速度與縮短時間。Quadt和Kuhn (2007)兩位學者們使用巢狀基因演算法(Nested genetic algorithms; NGA)解決批量配置的排程問題，將訂單分割成數個相同的批量大小，使能夠同時間分散至數台完全平行機台上，進行加工。

### 2.1.3 混合法 (Hybrid approaches)

Akrami (2006)等學者提出了對於多產品的經濟生產批量的排程問題，且站與站之間的緩衝是有限的，他們首先提出了新的混整數非線性規劃(Mixed integer nonlinear program; MINP)的方式，但由於難以求取最佳解，因此採用基因演算法與禁忌搜尋的方式，結果顯示基因演算法與禁忌搜尋都有不錯的績效。Jenabi (2007)學者提出一個在經濟批量的排程問題於HFS生產系統中，其生產環境為非等效平行機台，採取基因演算法和模擬退火法兩種啟發式的方法分別進行驗證，結果顯示基因演算法有較佳的解，而模擬退火法則計算時間較短。Shiau (2008)等學者在HFS系統中提出一種有效的混合方法，以建構式基因演算法(Constructive genetic algorithm; CGA)為基礎，第一階段先評估總成本下的排程，第二階段再透過禁忌搜尋來搜尋改善，以總加權完工時間最小化為目標。

#### 2.1.4 小結

從過去的學者所做的研究中(表 2.1)，可發現基因演算法在 HFS 應用相當廣泛，各種不同的領域，或者是不同生產環境的因素，都得到不錯的結果。其中 Chu 和 Beasley (1997)兩位學者在多階多機台環境下，使用資源成本最小之排程與資源配置問題為例，比較禁忌搜尋、模擬退火法與基因演算法的求解績效，研究中發現基因演算法擁有較佳的求解品質。此外亦有其他學者，將基因演算法結合其他啟發式演算法，透過其他演算法的優勢，將兩種合併以得到更佳的品質。因此，本研究決定使用基因演算法作為本研究的方法，最後將與混整數線性規劃求解之最佳解，進行比較與分析。

此外，太陽能電池之因素包含完全平行機台、批量分割、訂單優先權與設置時間，其中最為重要的批量分割方式，目前已知 Quadt 和 Kuhn (2007)學者所提出巢狀基因演算法算，能將訂單分割成數個相同的批量大小之子批，並且同時依照產品種類不同將其分散到數台完全平行機台上，進行加工，但此以晶矽太陽能電池產業角度看此方法，此方法忽略了兩個重要因素，分別為忽略整備時間與不以訂單角度做排程，由於晶矽太陽能電池產業中，最關鍵性的物料矽晶圓，有多家供應商，而每家供應商所提供之矽晶圓，有些許的差異，在製程的環節中，會有相對應的機台參數設定(Recipe)，若需求是相同規格的產品，但因使用不同供應商所提供之物料，則無法將兩張訂單合併規劃，因此無法忽略訂單角度而採用以產品種類角度作為排程依據，並且本研究為在假設供應商提供的原料皆能隨時取得不受其他因素影響下，進行求解。由於上述兩點在晶矽太陽能電池的排程中，是影響排程的關鍵因素。因此，在文獻較少的情況下，無疑的提升了整個研究計劃的挑戰。

表 2.1 相關文獻整理

作者(年)	機台特性			訂單限制						方法
	完全 平行 機台	等效 平行 機台	非等效 平行 機台	不完全 作業	批量 分割	訂單 優先權	搬運 時間	設置 時間	回流	
Kim et al. (1997)					◎			◎		啟發式法則
Botta-Genoulaz (2000)						◎	◎	◎		啟發式法則
Tseng et al. (2008)				◎						啟發式法則
Low et al. (2008)			◎				◎	◎		啟發式法則
Alisantoso et al. (2003)	◎									免疫演算法
Jin et al. (2002)	◎									基因演算法
Oğuz&Ercan(2005)	◎				◎					基因演算法
Belkadi et al. (2006)	◎									基因演算法
Yang et al. (2007)	◎							◎		基因演算法
Quadt&Kuhn (2007)	◎				◎			◎		基因演算法
Chu&Beasley (1997)										基因演算法 禁忌搜尋 模擬退火法
Bertel&Billaut (2004)		◎							◎	啟發式法則 貪婪演算法 基因演算法
Akrami et al. (2006)	◎				◎			◎		MINP 基因演算法 禁忌搜尋
Jenabi (2007)			◎		◎			◎		基因演算法 模擬退火法
Shiau et al. (2008)	◎									基因演算法 禁忌搜尋
本研究模式 (2010)	◎				◎			◎		基因演算法

## 2.2 基因演算法

### 2.2.1 基因演算法介紹

基因演算法為 John Holland 在 1975 年所提出的一般性最佳化演算法則。其理論基礎可回朔自 1895 年達爾文(Charles Darwin)物種演化書中的「物競天擇，適者生存」之演化及淘汰觀念。藉著模仿自然界遺傳演化的生物特性，透過複製(Reproduction)、交配(Crossover)以及突變(Mutation)等方式對可能的解進行運算，並根據適應函數來評估，逐步演化改善求解品質，以求得近似最佳解。

基因演算法同時以多點方式搜尋最佳解，而非點對點的搜尋，對於多峰谷之函數而言，基因演算法較傳統演算法更可以較快找出整體近似最佳解(Global optimum)，同時也能避免陷入區域最佳解(Local optimum)。此外，基因演算法有下列優點：

1. 產生可解釋的結果
2. 結果易於應用
3. 可以處理的資料型範圍極大
4. 可以用在最適化 (Optimization) 問題上

以基因演算法來搜尋近似最佳解的問題時，已有眾多學者討論過，尤其是於基因演算法運作步驟也有許多學者進行探討，如：Vignaux and Michalewicz (1991)探討參數編碼(Encode)之方式；Ishibuchi et al. (1994)設定控制參數，如：交配率、突變率以及停止運算的條件等。

### 2.2.2 基因演算法之方法與流程

基因演算法主要運用三種基本運算來模仿自然界的生物演化過程，包括天擇(Selection)、交配(Crossover)、突變(Mutation)及停止條件(Stopping criterion)，而在進行這些運算時，必須先決定編碼(Encoding)與適應函數，圖 2.3 為主要流程步驟，其各步驟分別說明如下。

1. 編碼(Encoding)：

在使用基因演算法之前，第一個步驟是將問題表示為演算法所能處理的型態，編碼的設計就好比生物的染色體結構，在據此設計適應函數。一

般的編碼方式係採二進位(Binary)，另外尚有整數(Integer)、實數(Real)及文字(Alphabet)等編碼方式。在基因演算法中需將問題的解答表示成可以運算的形式，即染色體編碼。大部分最佳化問題都有固定數目的變數，最普遍的編碼方是就是將這些變數對應到某個字元或整數，將其編成固定數目的位元，這些位元組合起來後就成為一個染色體，基因演算法即運用此編碼後的染色體進行後續運算。

## 2. 適應函數(Fitness function)：

染色體解碼後代入適應函數即可求得適應值，適應函數主要用以評估每個染色體的優劣，並指出族群演化的發展方向，通常適應函數為問題最佳化的目標函數，但有時目標函數需經過正規化處理。

## 3. 天擇(Selection)：

天擇運算子隨機從母代族群中挑選染色體，以繁衍子代。染色體被選中的機率與其適合度函數值成比例。適合度函數值越高，其被選擇的機率就會越高；反之，則被選擇的機率越低。選擇機制係模擬自然界適者生存的機制。

## 4. 交配(Crossover)：

基因演算法中，藉由母代染色體間的基因互換，母代染色體會隨機的配對，互換基。藉由這種方式，染色體間可互換資訊，以使得子代染色體能有更佳的適合度函數值。交配的方式分為單點交配、兩點交配、多點交配及均一交配。

## 5. 突變(Mutation)：

在自然界中個體有時會因為外在因素的刺激，而使本身的基因產生改變。突變可避免個體在交配的過程當中，失去一些重要的資訊，除此之外，藉由突變，也可以增加個體間的變異性，它讓生物有機會發展出具新特性的個體，使其更具多樣性。

## 6. 停止條件：

一般常見的停止條件可透過設定演化的世代數、電腦運算演化時間或在連續演化多個世代後，最佳化的目標函數直接相同或增加率未超過一定的百分比。

上述介紹了基因演算法的基本定義與概念，由於基因演算法為一近似

最佳化求解工具，特別適合於有效解答空間大、非線性、甚至無法確定可解的問題，是傳統決定性最佳化(Deterministic optimization)或貪婪法則(Greedy heuristics)無法做到的。

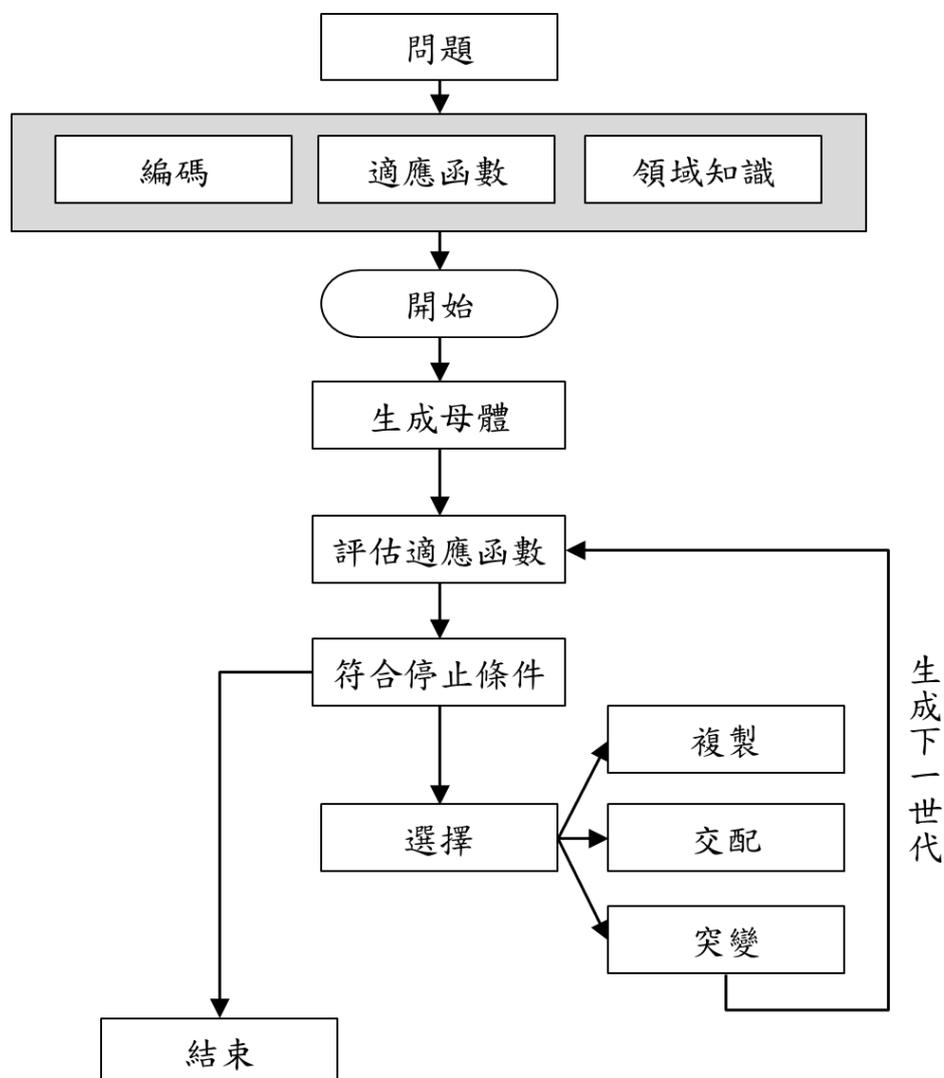


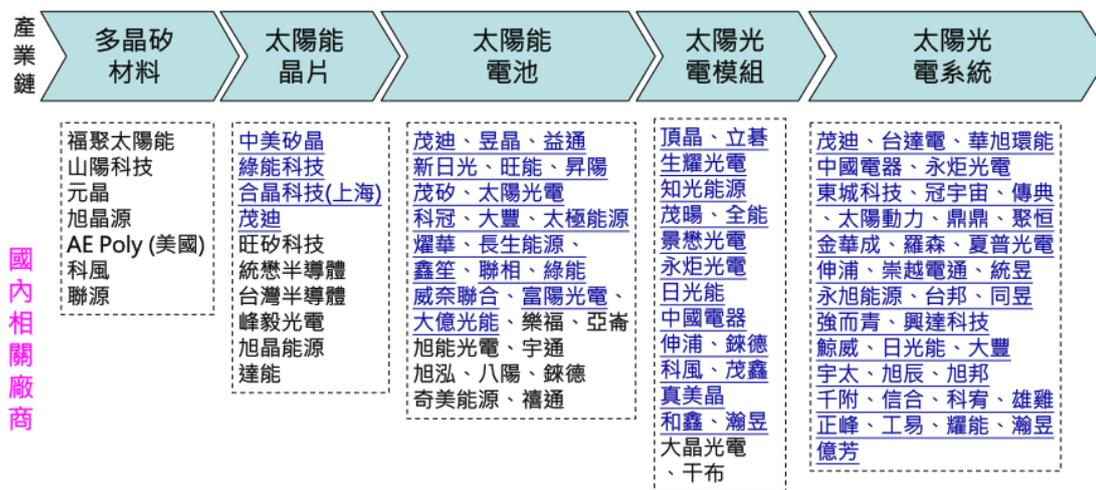
圖 2.3 基因演算法之邏輯

## 第三章 多階平行機台之基因演算法

本章節先詳述晶矽太陽能電池製造業之現況與製程特性，定義待解決之問題，並提出此問題下的數學模型，但前述之文獻已證明為 *NP-Complete*，因而進而說明本研究所提出 Makespan 最小化之基因演算法，在訂單需求及機台產能數量已知，考量晶矽太陽能電池製程特性，以提供較佳的訂單順序配置方式與機台資源規劃結果。本章節主要內容與架構為：3.1 節晶矽太陽能電池製造業現況，3.2 節為生產排程與製程特性說明，3.3 節為問題描述，3.4 節為數學模型，3.5 節 Makespan 最小化之基因演算法。

### 3.1 晶矽太陽能電池

本研究所探討之主題以生產晶矽太陽能電池為主之製造廠，近幾年來因為石油不斷飆漲，加上環保意識高漲，且以目前發電主流的所用的煤、石油、天然氣等化學石燃料來說，有其蘊藏量的限制，使用後無法回復，迫使世界各國必需找尋下一個替代能源，現階段所開發的能源都是往再生能源發展，因為這些再生能源發電所需的原料，是取之不盡用之不竭，有極度明顯的對比，再加上這些再生能源不會產生其他危害環境的副產品，如二氧化碳等。因此，全球太陽能光電產業進而高速發展，且未來成長的無限想像空間。圖 3.1 為本國太陽能產業鏈基本由四個環節構成，分別是矽原料生產 (Silicon)、晶圓製造 (Wafer)、太陽能電池 (Cell)、太陽電池模組 (Modules) 及太陽電池系統 (System)。中游太陽能電池的技術與原本半導體產業製造基礎接近，故吸引台灣半導體大廠業者積極投入，政府也積極發展太陽能產業，將是未來傲視全球的產業之一(莊習武，2006)。



註：加底線為量產廠商，其餘為尚未量產廠商

2005廠商數	0家	2家	8家	5家	25家
2006廠商數	0家	2家	13家	8家	27家
2007廠商數	2家	7家	24家	14家	29家
2008廠商數	8家	10家	28家	18家	36家
2009廠商數	7家	10家	28家	19家	37家

圖 3.1 本國太陽能產業鏈結構

(資料來源：工研院產業經濟與趨勢研究中心，2009/07)

在太陽能電池中，分為三種類(圖 3.2)，而晶矽太陽能電池又分為單晶矽與多晶矽兩種，其製程生產步驟皆相同，但在第一道蝕刻製程中，單晶矽與多晶矽有專用機台，後續的製程中，皆採用相同的製造機台，形成第一道蝕刻製程有專屬機台，後續製程皆為共用機台的生產環境，如圖 3.3 所示，而 BOM 相對較為簡單(圖 3.4)，因此對於晶矽太陽能電池製造業，產能上的規劃較物料複雜許多。就目前產業現況，矽晶類的晶矽太陽能電池為最大宗(圖 3.5)，係因為轉換效率為目前各種類之太陽能電池之冠外，且有機台共用的特性，許多廠商能夠同時生產單晶矽與多晶矽之太陽能電池，而未來十年之佔有率仍在 50% 以上，其重要性不可忽視。

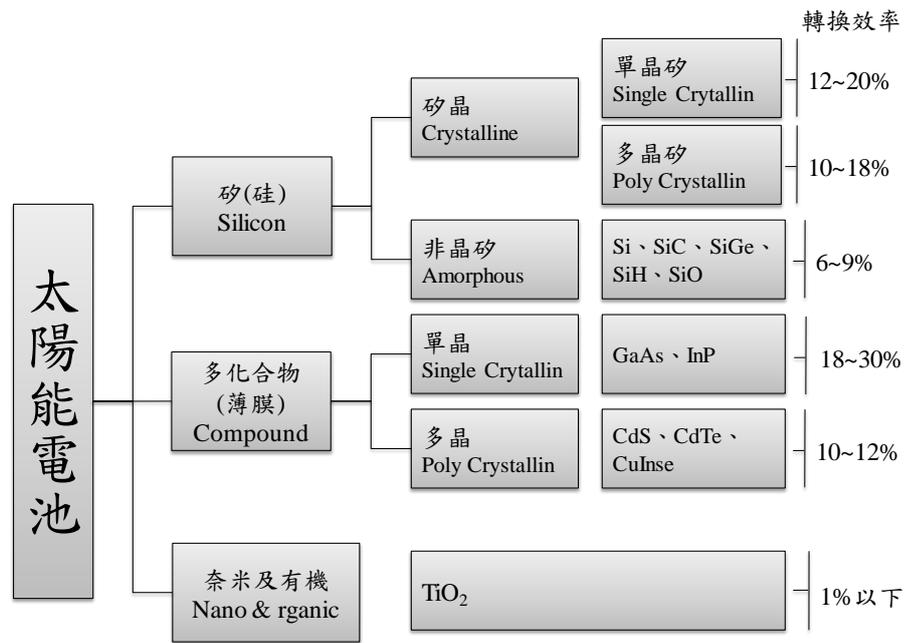


圖 3.2 太陽能電池種類 (本研究整理)

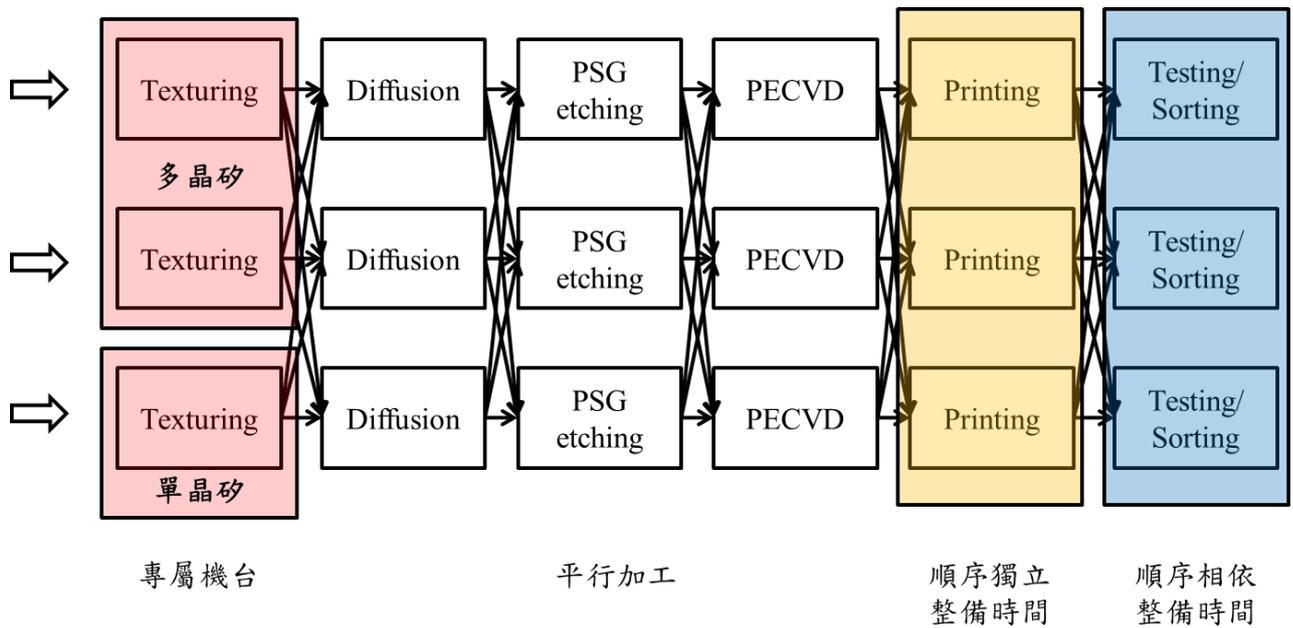


圖 3.3 太陽能電池製程流程與製程特性

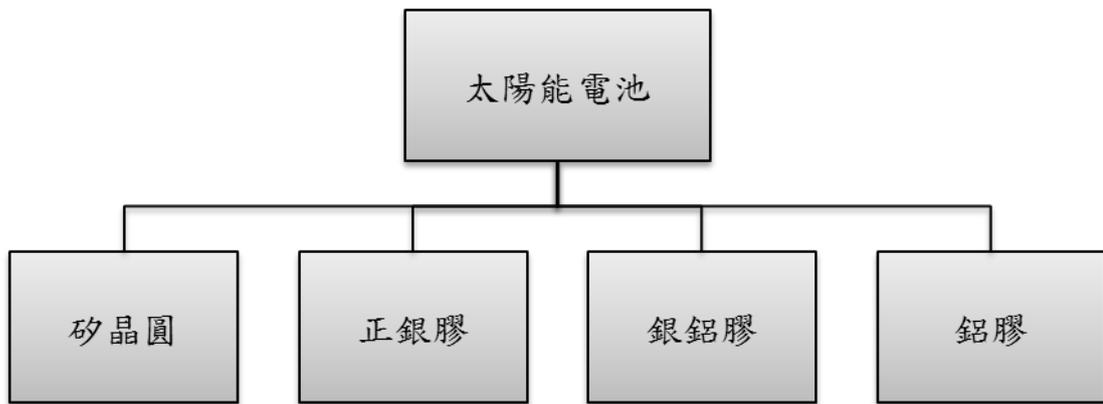


圖 3.4 太陽能電池 BOM

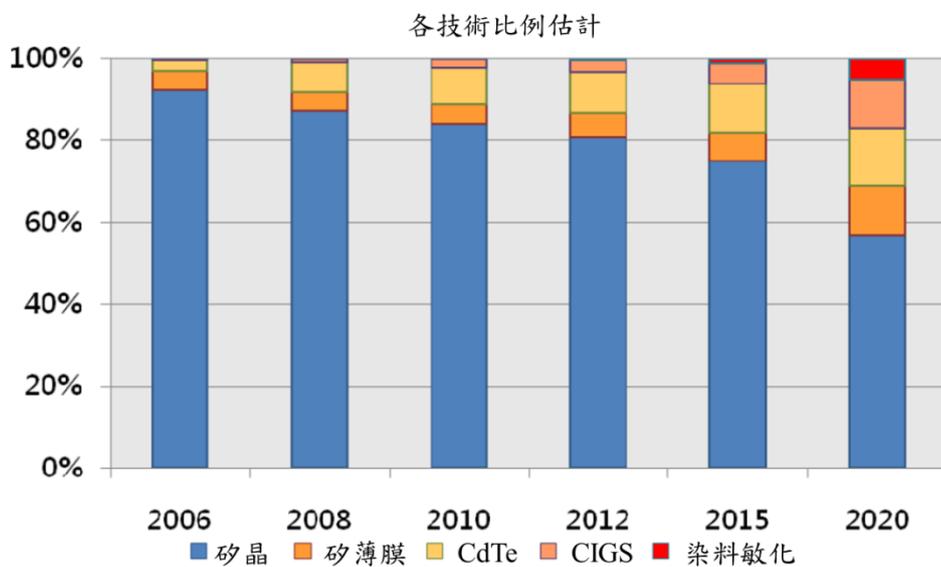


圖 3.5 太陽能電池種類比例估計

(資料來源：工研院產業經濟與趨勢研究中心，2009/08)

## 3.2 製程特性與問題定義

### 3.2.1 晶矽太陽能電池的生產製程

太陽能電池種類繁多，本研究係以矽晶類的晶矽太陽能電池為探討，也為本國之太陽能光電產業，於太陽能電池中，佔最大部分的一種，圖 3.6 為晶矽太陽能電池製程的步驟，屬於流程型生產的模式，其中每道製程的可用機台數量，皆大於或等於一，則符合 HFS 的生產特性，屬於 HFS 的生產環境(Ruiz & Vázquez-Rodríguez, 2010)，其細部各製程說明如下：

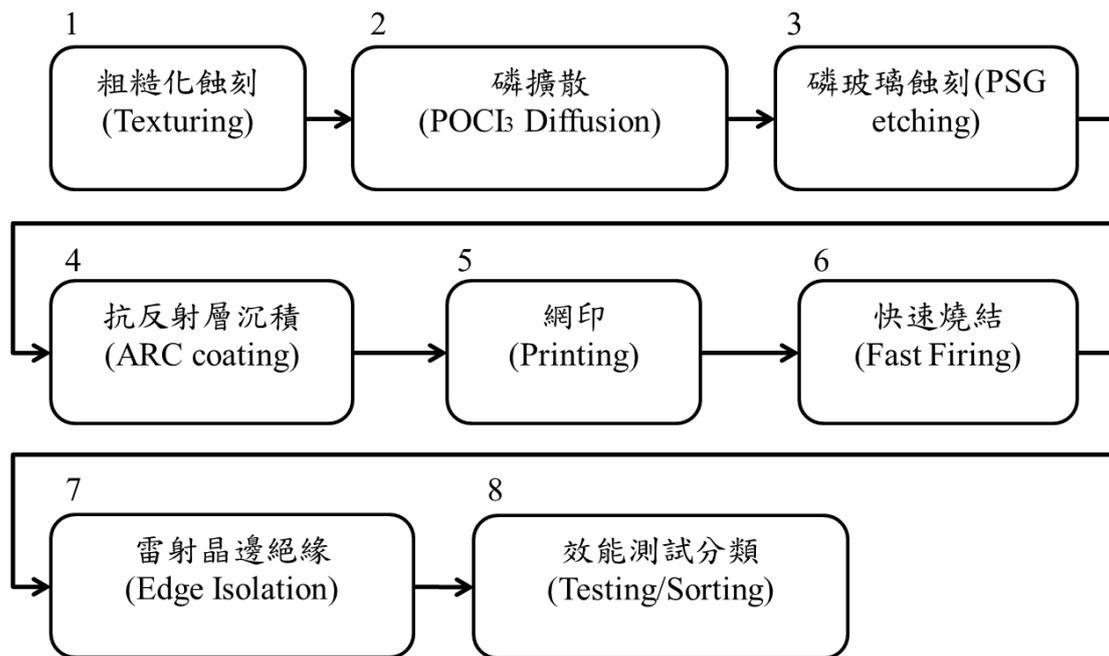


圖 3.6 晶矽太陽能電池製造流程

1. 粗糙化蝕刻(Texturing)：

使晶片表面粗糙化，增加晶片吸光能力(圖 3.7)。

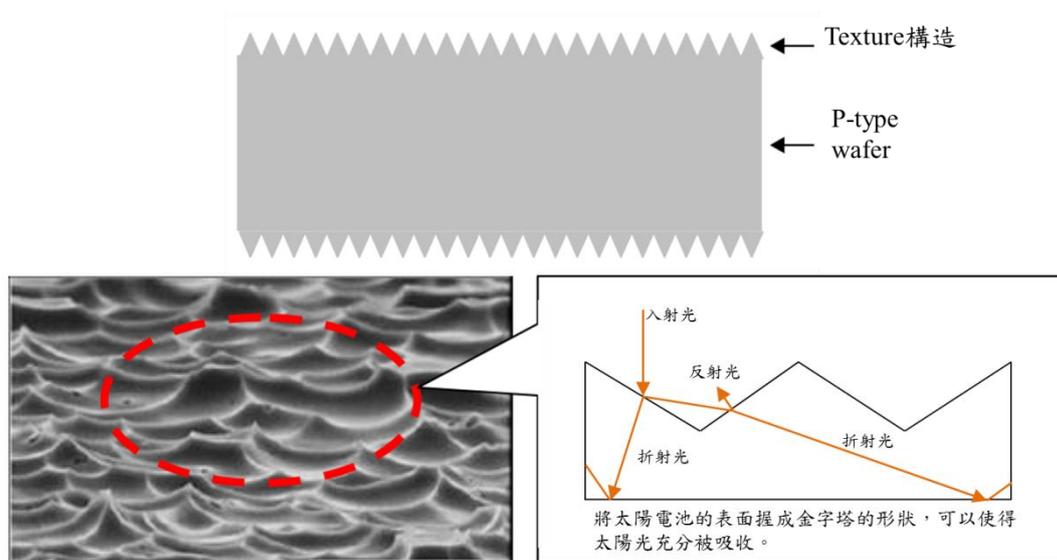


圖 3.7 蝕刻製程 (長生能源)

2. 磷擴散爐(POCl<sub>3</sub> diffusion)：

在 p-type 晶片上形成一層 n-type (圖 3.8)。以氣體方式( $N_2 + O_2$ )將帶有磷原子之不純物( $POCl_3$ )送入，在高溫 ( $850\sim 950^\circ C$ ) 下進行化學反應，

磷原子由濃度高擴散至濃度低之 P 型 Si 上，形成大面積的 P-N 接合二極體。

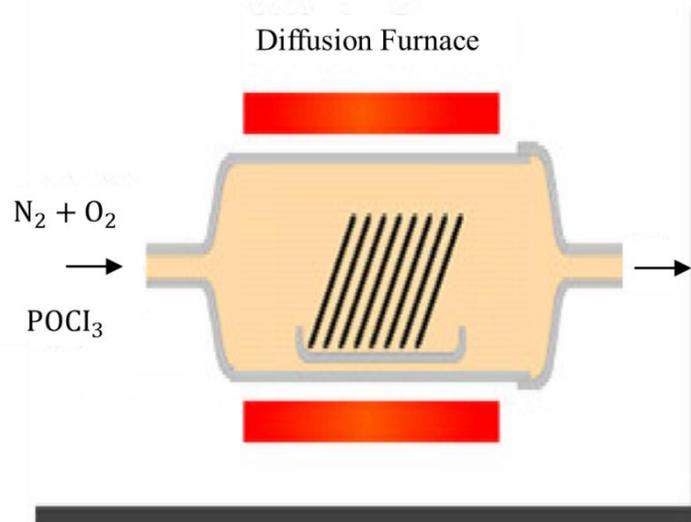


圖 3.8 磷擴散爐（長生能源）

### 3. 磷玻璃蝕刻機(PSG etching)：

將磷擴散後產生的副產品去除(圖 3.9)，磷擴散製程時會於生產過程中，在晶片表面中會有  $SiO_2$  及  $PO_2$  之殘留，該物會影響到電池轉換效率，因此，利用 HF (磷玻璃蝕刻機)將其移除，以達成提升轉換效率的目的。

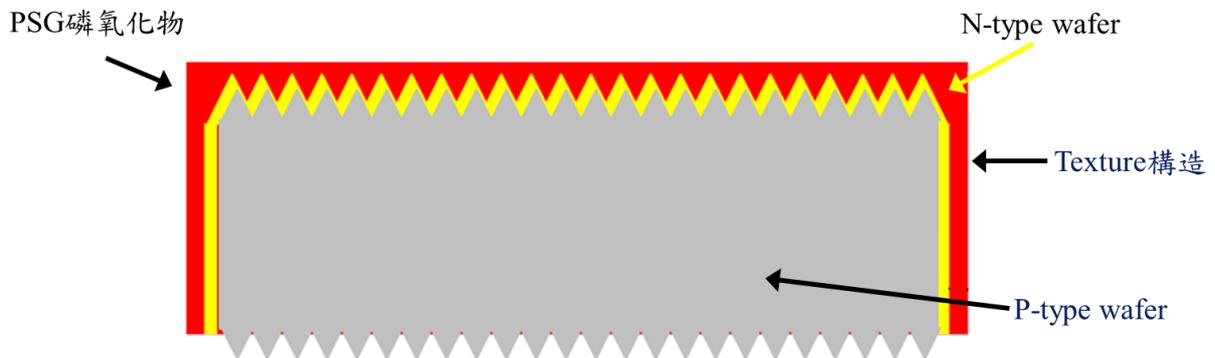


圖 3.9 磷玻璃蝕刻機（長生能源）

### 4. 抗反射層沉積(ARC coating)：

在晶片表面形成一層  $SiN$  AR coating，減少光線的反射(圖 3.10)。利用電漿增益化學氣相沉積法(Plasma-enhanced chemical vapor deposition; PECVD)於晶片擴散上面沉積  $Si_3N_4$ ，用以減少陽光反射，達成增加光

吸收率，提高轉換效率的目的。並保護電池不受輻射及大氣的影響，延緩元件老化，增加元件壽命(Shimizu, Kondo, & Matsuda, 2005)。

反應式：

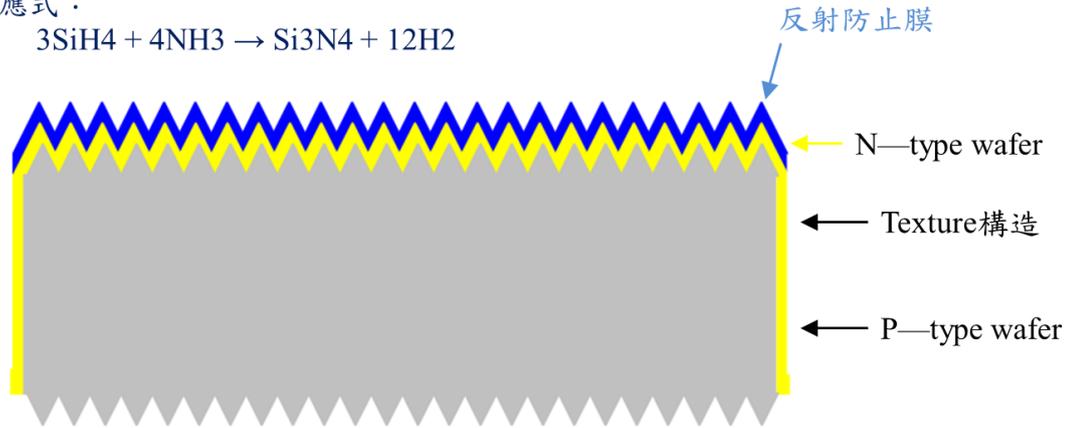


圖 3.10 抗反射層沉積 (長生能源)

5. 網印(Printing)：

在晶片正反面印上所需的電極樣式來收集與傳導電流(圖 3.11)。

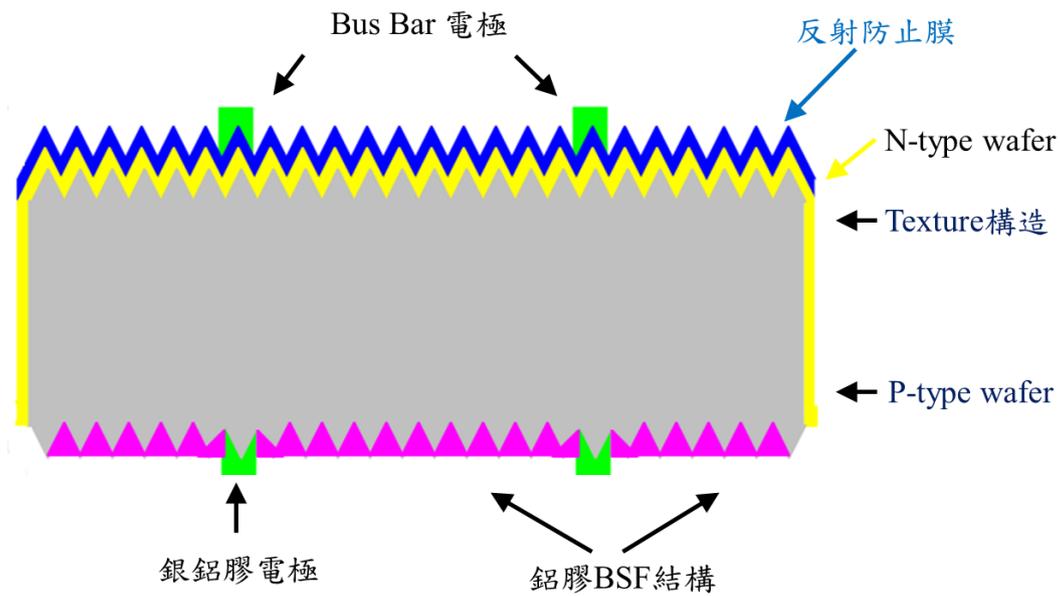


圖 3.11 網印 (長生能源)

6. 快速燒結(Fast Firing)：

將金屬膠的有機物與樹脂燃燒，並使金屬與晶片合金(圖 3.12)。

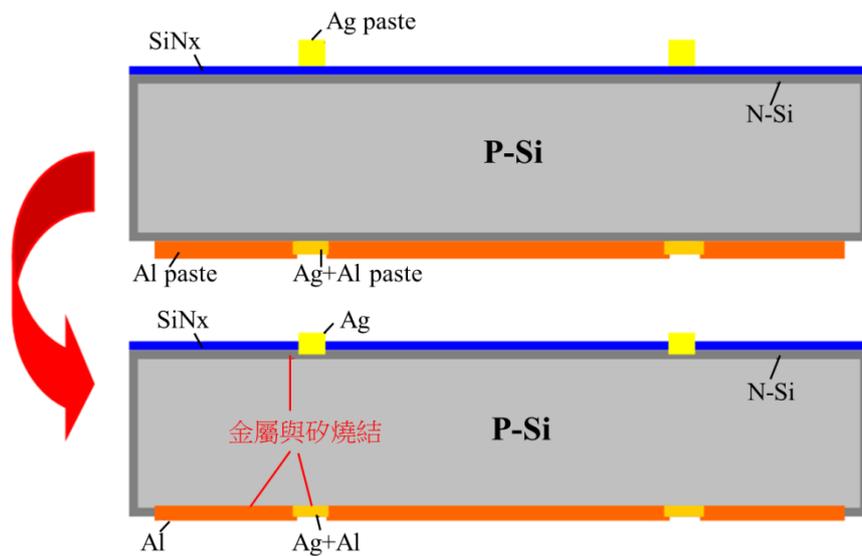


圖 3.12 快速燒結（長生能源）

7. 雷射晶邊絕緣(Edge isolation)：

用雷射將晶片邊緣會短路的地方切斷，Cell 於磷擴散製程會於 Cell 外圍周邊形成磷沉積，造成 P-N 層上下面導通降低電池效能，因此此製程以雷射晶邊絕緣作業將該層絕緣(圖 3.13)。

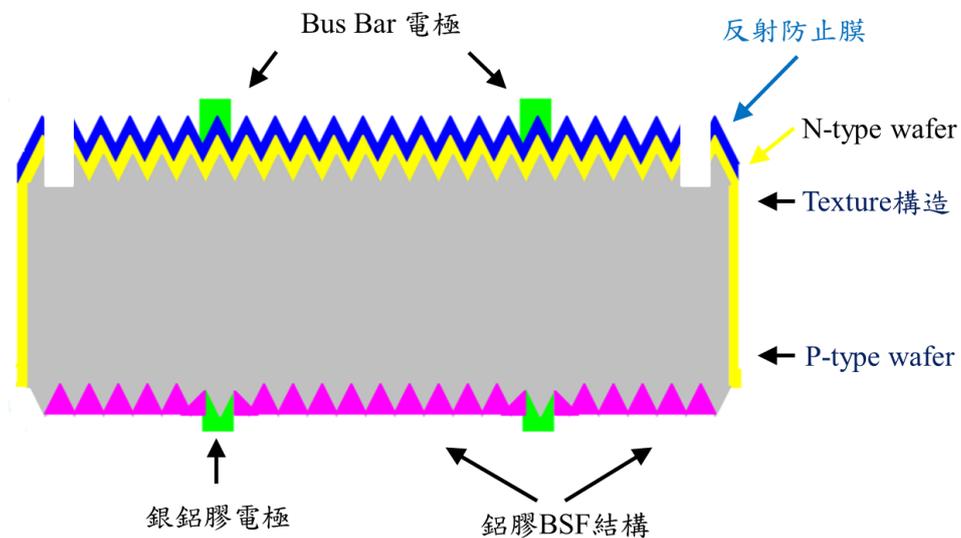


圖 3.13 雷射晶邊絕緣（長生能源）

8. 效能測試分類(Testing/Sorting)：

將做好的太陽能電池作轉換效率測試並分類。如圖 3.14 所見模擬太陽光照射 Cell，量測 Cell 的電壓、電流、功率與轉換效率，並依據外觀、

轉換效率或其他電性做分級，將 Cell 做分類。

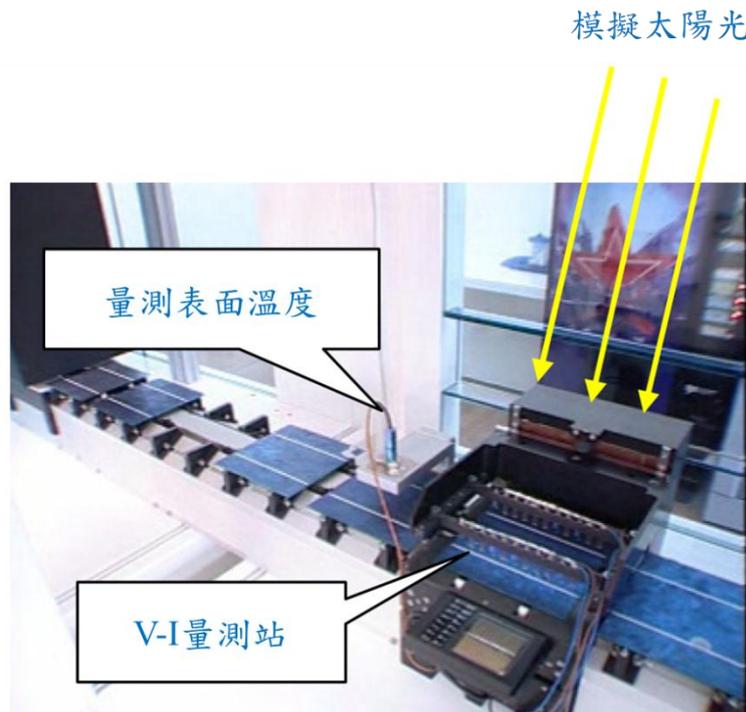


圖 3.14 效能測試分類（長生能源）

### 3.2.2 晶矽太陽能電池之製程特性

晶矽太陽能電池屬於 HFS 的生產環境，共有六道製造程序，各製造程序各有其特性，皆是影響生產排程的重要因素，因此在進行生產排程時，須納入這些製程特性，其詳細之特性說明如下：

#### 1. 平行機台加工：

在晶矽太陽能電池的製程上，機台類型屬於完全平行機台，其每道製程擁有  $n$  台以上的平行機台，目的在於，當訂單之需求量較大，必須將訂單配予一台以上的機台，增加其產能，以縮短訂單完工時間；或某製程因機台需有整備時間，如圖 3.15 所示，(a)為將訂單均分割成三張工單，並分配至三台機台上進行作業，雖然配置三部機台以增加產能，縮短工時，但相對需付出 18 次的整備時間，其完工時間為 33；(b)將一張訂單分配至一台機台，雖然只需要 6 次的整備時間，但因為供給的產能較少，相對需要較多的作業時間，而完工時間為 39；(c)考量機台有設置時間，Stage  $n$  為不需考量產品型態屬於獨立整備，規劃時獨立整備的重點在於必須考量機台的配置數量，因為多給予一台機台的產能，相對多一次的整備時間，所以

必須評估當產能增加，整備次數也增加的情況下，所需作業時間是否減少或增加為關鍵所在；而 Stage  $n+1$  則需考慮產品型態屬於相依整備，而重點為訂單的生產順序，當機台連續加工相同產品型態的訂單，則不需要整備，反之，若為不同之產品型態，則需要整備，該如何控制相同產品型態的訂單，於同一機台上作業，以減少整備次數，縮短作業時間，所以在有考量兩種整備的因素下，完工時間為 31，皆較上述兩種方式為短。因此，平行機台加工雖然增加了生產上的彈性，但要如何規劃，卻成為生管人員在進行排程時的困難處。

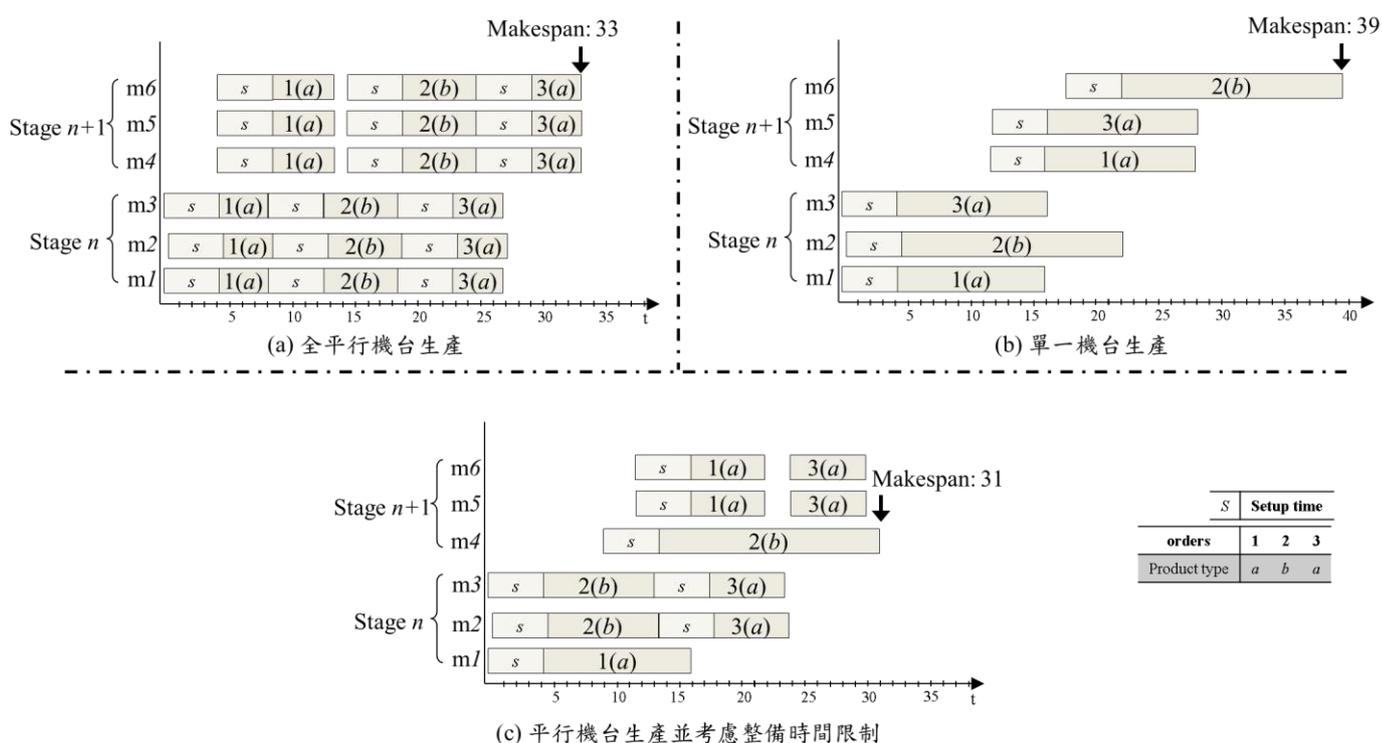


圖 3.15 平行機台示意圖

## 2. 專用機台：

在晶矽太陽能電池中，分為單晶矽太陽能電池與多晶矽太陽能電池，其製造程序均相同，使用相同的機台與相同之生產流程，最大的差異點在於，第一道製程蝕刻(Texturing)有專用機台的情況，蝕刻製程主要目的為使用特定化學溶劑，使晶片表面粗糙化，增加晶片吸光能力，而多晶矽太陽能電池採用酸性蝕刻的方式，而單晶矽太陽能電池採用鹼性蝕刻的方式，因此不同特性之晶矽太陽能電池，其機台在此製程不可混用，如圖 3.16 所

示。也因如此，當多晶矽與單晶矽之專用機台數量不一致時，因產能的不同，將會影響後續製程的排程方式。

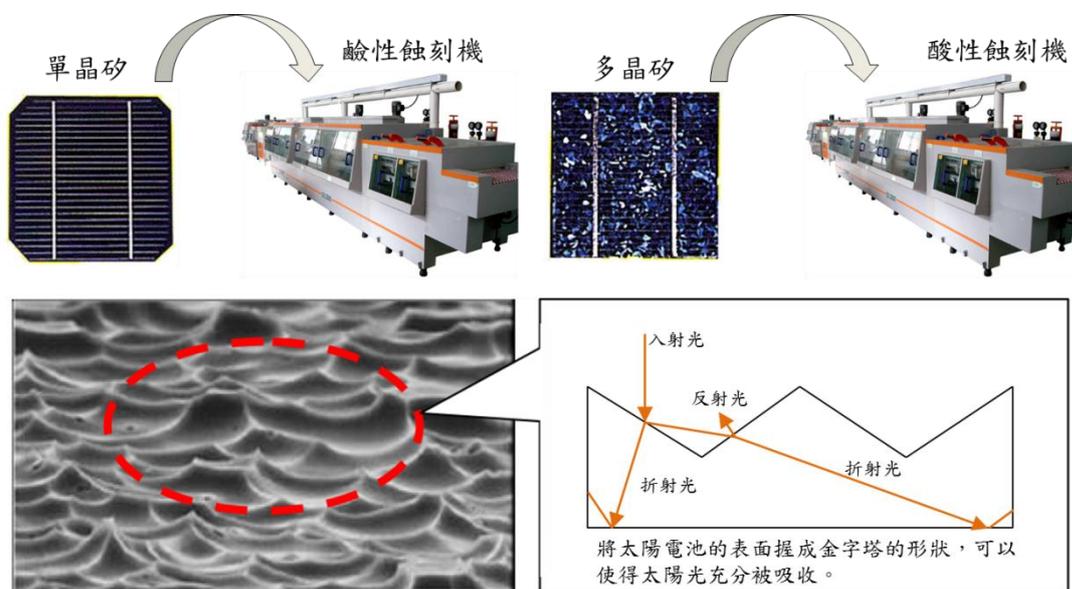


圖 3.16 蝕刻製程專用機台

### 3. 順序獨立整備時間：

網印(Printing)製程中，其在晶片正反面印上所需的電極樣式來收集與傳導電流，圖 3.17 為晶矽太陽能電池版，兩道較寬之電極為匯流電極(Busbar)，左右兩邊所延伸許多細小的電極，則為柵極(Finger)，而網印的圖案有很多樣式，但通常為了實際考量，通常都以簡單化設計並成對稱式；因此，客戶所要求之網印圖案均不相同，如所圖 3.17 所見，圖中(a)的太陽能電池，將網印柵極放大來看，其柵極分布較疏(距離大)，而圖(b)之柵極分布則較密(距離小)，疏與密完全取決於客戶的需求，來設置不同圖案之網版，由於每位客戶的需求相似性極小，因此訂單進入網印製程的先後順序，不影響其設置時間的長短或有無，此為順序獨立設置時間(Sequence independent setup times)。此外，如圖 3.15 所見，獨立整備時間影響著整個排程的績效，如何有效的控制整備時間的次數，進而縮短整體的完工時間，以得到更好的規劃結果。

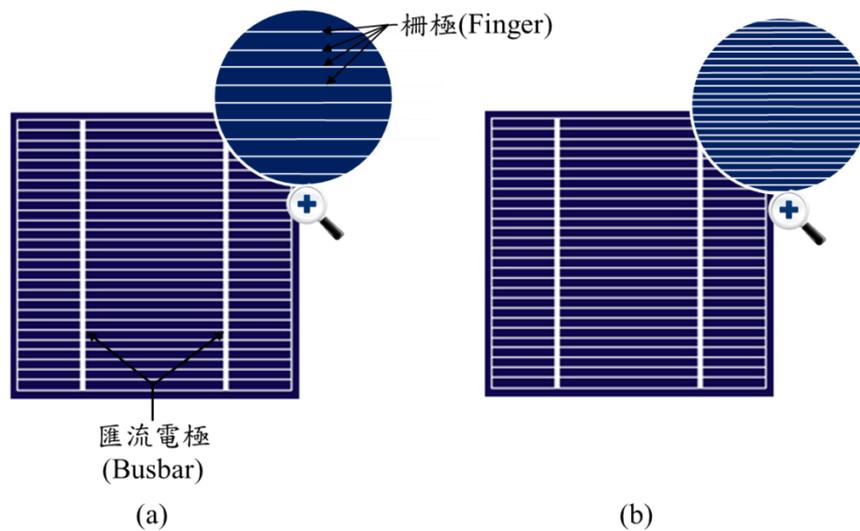


圖 3.17 網印製程圖

#### 4. 順序相依整備時間：

測試(Testing/Sorting)製程中，目的係模擬太陽光照射 Cell，量測 Cell 之電壓、電流、功率與轉換效率，才能得以將 Cell 進行分類，因晶矽太陽能電池表面之電極數(Busbar)，共分為 2 Busbar 與 3 Busbar 兩種，如圖 3.18 所示，因為兩種電極之間的數量不同，在測試製程的機台中，量測的探針需要調整符合兩種電極之數量，以圖中(a)為例，屬於 2 Busbar 的形式，因此需要裝設兩支探針來進行量測之作業，而(b)則屬於 3 Busbar 的形式，機台需要架設三隻探針來進行量測作業，因此當量測機台前一批作業屬於 2 Busbar 的產品形式，當下一批的作業若為相同(2 Busbar)之產品形式，則不需要更改量測機台之探針數；反之，若為 3 Busbar 的形式，則需要進行機台探針數調整之設置，即探針的調整時間會受前後產品類型順序，而影響設置時間的有無，此為順序相依設置時間(Sequence dependent setup times)。因此，如圖 3.15 所見，訂單的產品類型，影響該訂單在此製程的生產順序，該如何訂定較佳的生產順序，得以能減少整備次數並縮短整體完工時間，是考量此限制的重點。

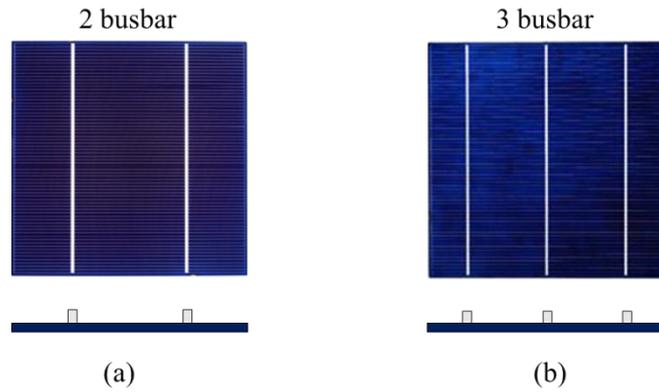


圖 3.18 太陽能電池測試分類圖

因此，本研究以晶矽太陽能電池之生產排程做為案例，針對上述之四大關鍵因素，包含平行機台加工、專用機台、順序獨立整備時間與順序相依整備時間，探討其生產排程之方式，期盼能在 Makespan 最小化的情況進行生產規劃。

### 3.3 問題描述

在太陽能電池製造中，矽晶太陽能電池屬於混合流程型生產環境，除了上述之製程特性，如專屬機台、平行加工、獨立整備時間與相依整備時間，其訂單的產品數量皆相當高，並非短時間內即可生產完成，必須考量批量分割的方式，使得訂單平均分割成數個工單後，使相同訂單下之各工單能同時間在同一個製程中的數台機器上加工，進而縮短生產時間，以圖 3.19 為例，在 Stage  $n$  時，訂單 1 均分為兩個子批量，批次 1 規劃於 Machine 1 上作業，批次二則規劃至 Machine 2 上作業，而下一階段 stage  $n+1$  中，訂單 1 則不進行分割，直接配置於 Machine 4 上作業，其目的必須設法規劃出各種組合，使得整體規劃之完成時間最小。因此除了考量訂單的生產順序外，如何將訂單分割為多少批量，且同一時間上因分配至哪些機台，是本研究欲解決的問題。因此，本研究的問題有下列三點：

1. 決定訂單生產順序
2. 訂單的批量分割
3. 分割後批量的機台配置

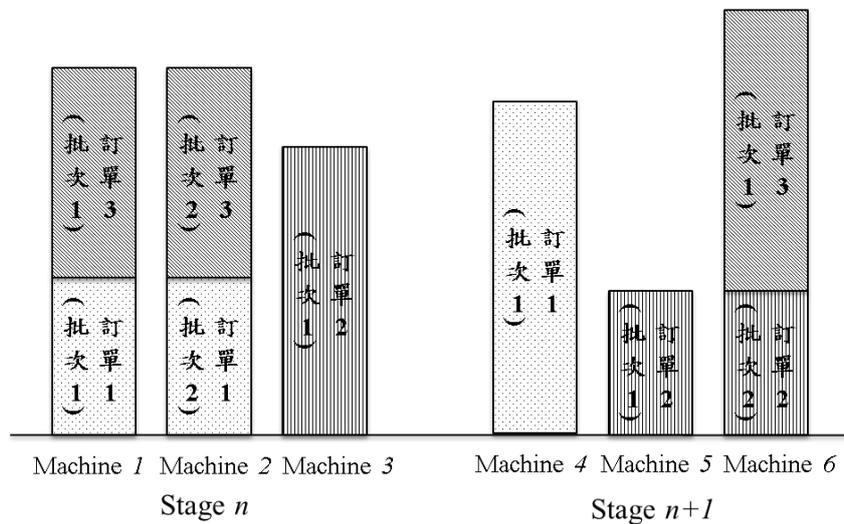


圖 3.19 分割訂單與機台選擇示意圖

將透過晶矽太陽能電池的生產製程特性，演算出較佳的排程結果，目標式採用最大完工時間最小化，此績效指標以作為排程搜尋的目標依據，其許在最低的總完工時間下，獲取最佳的機台排程結果。

### 3.4 數學模式

本數學模式為張晉璋 (2011)所提出，定義出太陽能電池產業排程模式，分別說明此模式之假設條件及符號定義，並以標準型 HFS 數學模式為基準，建構一混整數線性規劃模式(MILP Model)。

#### 3.4.1 假設條件

1. 所有的工單及機台皆可在時間點零開始規劃
2. 每一加工階段中的各機台皆相同
3. 任一台機台一次僅能處理一個作業
4. 作業執行過程中不允許中斷
5. 加工階段間之暫存區為無限產能
6. 當工單的需求量非為各階段中所有機台數目的整數倍時，可忽略其計算時間之小數位
7. 問題的相關資料為確定性且已知的

### 3.4.2 符號定義

#### 下標及參數

$i$  : 工單的數量，其中工單的編號為下標  $i=1, 2, \dots, I$

$u$  : 工單的數量，其中工單的編號為下標  $u=1, 2, \dots, I$

$j$  : 加工階段的數量，其中加工階段的編號為下標  $j=1, 2, \dots, J$

$M_j$  : 在加工階段  $j$  機台的數量，其中機台的編號為下標  $k=1, 2, \dots, M_j$

$L_j$  : 在加工階段  $j$  拆批方式的數量，其中拆批方式的編號為下標  $l=1, 2, \dots, L_j$

$D_i$  : 工單  $i$  的需求數量

$P_{i,j}$  : 工單  $i$  於階段  $j$  的單位加工時間

$r_{j,l}$  : 階段  $j$  其拆批方式  $l$  之生產比率

$s_i^{SI}$  : 工單  $i$  所需之獨立整備時間

$B_i$  : 工單  $i$  所需生產之品項的電極數目 (=2 or 3)

$s_{B_i, B_u}^{SD}$  : 工單  $i$  為工單  $u$  之前一個加工作業時，工單  $u$  所需之相依整備時間

$E_{i,k} = \begin{cases} 1, & \text{工單 } i \text{ 可於機器 } k \text{ 上加工} \\ 0, & \text{otherwise} \end{cases}$

$M$  : 一個非常大的正數

#### 決策變數

連續變數：

$X_{i,j,k}$  : 工單  $i$  於階段  $j$  之機器  $k$  的生產比率

整數變數：

$C_{max}$  : 最大完工時間(Makespan)

$C_{i,j,k}$  : 工單  $i$  於階段  $j$  之機器  $k$  的完工時間

二元變數：

$$Y_{i,j,k} = \begin{cases} 1, & X_{i,j,k} > 0 \text{ (工單 } i \text{ 於階段 } j \text{ 之機器 } k \text{ 上加工)} \\ 0, & X_{i,j,k} = 0 \end{cases}$$

$$W_{i,j,k,l} = \begin{cases} 1, & \text{工單 } i \text{ 於階段 } j \text{ 之機器 } k \text{ 使用拆批方式 } l \\ 0, & \text{otherwise} \end{cases}$$

$$S_{i,u,j,k} = \begin{cases} 1, & \text{在階段 } j \text{ 之機器 } k \text{ 上, 工單 } i \text{ 於工單 } u \text{ 之前加工} \\ 0, & \text{otherwise} \end{cases}$$

### 3.4.3 混整數線性規劃模式

本小節提出一個混整數線性規劃的數學模式，適用於矽晶太陽能電池產業的排程問題，以下將及說明此數學模式的目標函數及限制式。

目標函數

$$\text{Minimize } Z = C_{max} \quad (1)$$

太陽能電池產業生產排程模式之目標為最小化最大完工時間( $C_{max}$ )，而此模式所需考量之限制式如下所示：

#### 1. 最大完工時間之限制式

$$C_{max} \geq C_{i,J,k} \quad \forall i, k \quad (2)$$

最大完工時間( $C_{max}$ )須大於等於所有工單於最後一站中所有機台的完工時間( $C_{i,J,k}$ )。

#### 2. 加工順序之限制式

$$C_{i,j-1,k} + D_i \times X_{i,j,k} \times p_{i,j} \leq C_{i,j,k} \quad \forall i, j, k \quad (3)$$

工單  $i$  於上一階段  $j-1$  之機台  $k$  的完工時間 ( $C_{i,j-1,k}$ )，加上工單  $i$  之子

批在階段  $j$  之機台  $k$  所需之作業時間( $D_i \times X_{i,j,k} \times p_{i,j}$ )，須小於等於工單  $i$  於階段  $j$  之機台  $k$  的完工時間( $C_{i,j,k}$ )，在此式中令  $C_{i,j=0,k} = 0$ ，表工單的排程起始點為時間點零開始。而工單之子批大小取決於機台使用的數量，也就是透過生產比率( $X_{i,j,k}$ )決定，此限制式亦表工單之開工時間必須大於等於工單於前一階段中任一機台的完工時間，工單必須等待其子批皆完工後，才可以開始於下一階段排定生產。

### 3. 批量分割與專用機台之限制式

$$X_{i,j,k} = \sum_{l=1}^{L_j} r_{j,l} \times E_{i,k} \times W_{i,j,k,l} \quad \forall i, j, k; j = 1 \quad (4)$$

$$X_{i,j,k} = \sum_{l=1}^{L_j} r_{j,l} \times W_{i,j,k,l} \quad \forall i, j, k; j > 1 \quad (5)$$

限制式(4)及(5)是用來決定工單於各階段中，使用多少機台排定生產，即決定工單分割為多少個子批，其中工單的所有批量分割/拆批方式皆為已知，由於每個階段中的機台數量皆為已知，因此透過機台數量即可得知工單於任一階段中所有可能的拆批方式，不同的拆批方式下亦代表工單不同的生產比率。除此之外，在第一階段中須考量專用機台限制，因此須透過參數  $E_{i,k}$  來限制，如限制式(4)所示。當工單  $i$  因為專用機台限制無法於機台  $k$  加工時( $E_{i,k} = 0$ )，表工單  $i$  在機台  $k$  的生產比率( $X_{i,j=1,k}$ )為 0。

$$\sum_{l=1}^{L_j} W_{i,j,k,l} \leq 1 \quad \forall i, j, k \quad (6)$$

限制式(6)是用來確保工單在任一階段之機台的生產比率( $X_{i,j,k}$ )僅能由所有可能的生產比率中擇其一或是為零，亦即當工單進行拆批時( $W_{i,j,k,l} = 1$ )，其拆批方式僅能擇其一，或是當工單不進行拆批時( $W_{i,j,k,l} = 0$ )，此時工單在某一階段中某一機台的生產比率則為零( $X_{i,j,k} = 0$ )，即工單  $i$  於階段  $j$  之機台  $k$  不進行生產。

$$\sum_{k=1}^{M_j} E_{i,k} \times X_{i,j,k} = 1 \quad \forall i, j; j = 1 \quad (7)$$

$$\sum_{k=1}^{M_j} X_{i,j,k} = 1 \quad \forall i, j; j > 1 \quad (8)$$

限制式(7)及(8)表工單  $i$  於階段  $j$  之所有機台所生產的批量須滿足工單的需求量，工單  $i$  於階段  $j$  之所有機台的生產比率( $X_{i,j,k}$ )總和要為 1。除此之外，在第一階段中須考量專用機台限制，因此須透過參數  $E_{i,k}$  來限制，如限制式(7)所示，即扣除不可生產之機台外，其他機台的生產比率總和要為 1。

#### 4. 工單作業順序之限制式 (Stage 1~4)

$$\begin{aligned} C_{u,j,k} &\geq C_{i,j,k} + D_u \times X_{u,j,k} \times p_{u,j} - M(2 - Y_{i,j,k} - Y_{u,j,k}) \\ &\quad - M(1 - S_{i,u,j,k}) \quad \forall i, u, j, k; i < u; j = 1, \dots, 4 \end{aligned} \quad (9)$$

$$\begin{aligned} C_{i,j,k} &\geq C_{u,j,k} + D_i \times X_{i,j,k} \times p_{i,j} - M \times (2 - Y_{i,j,k} - Y_{u,j,k}) \\ &\quad - M \times S_{i,u,j,k} \quad \forall i, u, j, k; i < u; j = 1, \dots, 4 \end{aligned} \quad (10)$$

限制式(9)表當工單  $i$  與工單  $u$  於任一階段( $j = 1, \dots, 4$ )中的同一機台加工( $Y_{i,j,k} \& Y_{u,j,k} = 1$ )，且工單  $i$  在工單  $u$  之前處理時( $S_{i,u,j,k} = 1$ )，工單  $u$  於階段  $j$  之機台  $k$  的完工時間( $C_{u,j,k}$ )須大於等於，工單  $i$  於階段  $j$  之機台  $k$  的完工時間( $C_{i,j,k}$ )，加上工單  $u$  之子批在階段  $j$  之機台  $k$  所需的作業時間( $D_u \times X_{u,j,k} \times p_{u,j}$ )，其中  $i$  不等於  $u$  ( $i < u$ )，以表示兩張工單彼此不相同。此限制式主要是透過一非常大的正數  $M$ ，迫使變數  $Y_{i,j,k}$ 、 $Y_{u,j,k}$  及  $S_{i,u,j,k}$  皆為 1 以消去  $M$ ，以避免任兩張工單其作業於相同機台發生重疊的情形。

#### 5. 工單作業順序與順序獨立整備時間之限制式 (Stage 5)

$$C_{u,j,k} \geq C_{i,j,k} + D_u \times X_{u,j,k} \times p_{u,j} + s_u^{SI} \times Y_{u,j,k} - M(2 - Y_{i,j,k} - Y_{u,j,k}) - M(1 - S_{i,u,j,k}) \quad \forall i, u, j, k; i < u; j = 5 \quad (11)$$

$$C_{i,j,k} \geq C_{u,j,k} + D_i \times X_{i,j,k} \times p_{i,j} + s_i^{SI} \times Y_{i,j,k} - M(2 - Y_{i,j,k} - Y_{u,j,k}) - M \times S_{i,u,j,k} \quad \forall i, u, j, k; i < u; j = 5 \quad (12)$$

$$C_{i,j,k} \geq D_i \times X_{i,j,k} \times p_{i,j} + s_i^{SI} \times Y_{i,j,k} \quad \forall i, j, k; j = 5 \quad (13)$$

限制式(11)與(12)如同限制式(9)與(10)，皆是用來決定工單的作業順序，以避免作業重疊。但除此之外，這兩條限制另須考量順序獨立整備時間。這兩條限制式皆是確保當任兩張工單於同一機台加工時，須予以錯開其作業時間，且兩張工單的作業之間須有一順序獨立整備時間，亦即從前一張工單之完工時間點，至後一張工單之開工時間點，兩者間會有一順序獨立整備時間，而此整備時間是為了生產後一張工單而產生。但這兩條限制式並未考量生產前一張工單所需之獨立整備時間，因此須透過限制式(13)以滿足此條件，此式表工單  $i$  於第五階段之機台  $k$  之完工時間須大於等於，工單  $i$  之子批於第五階段之機台  $k$  所需的作業時間，加上工單  $i$  於第五階段之機台  $k$  的順序獨立整備時間，即當任一張工單於首次於機台  $k$  加工時，必須考量獨立整備時間。另外，由此式可知，首次於機台處理之工單其排程起始點為時間點零開始，因此可使得工單之整備動作可預先開始。

## 6. 工單作業順序與順序相依整備時間之限制式 (Stage 6)

$$C_{u,j,k} \geq C_{i,j,k} + D_u \times X_{u,j,k} \times p_{u,j} + s_{B_i, B_u}^{SD} - M(2 - Y_{i,j,k} - Y_{u,j,k}) - M(1 - S_{i,u,j,k}) \quad \forall i, u, j, k; i \neq u; j = 6 \quad (14)$$

$$C_{i,j,k} \geq C_{u,j,k} + D_i \times X_{i,j,k} \times p_{i,j} + s_{B_u, B_i}^{SD} - M(2 - Y_{i,j,k} - Y_{u,j,k}) - M \times S_{i,u,j,k} \quad \forall i, u, j, k; i < u; j = 6 \quad (15)$$

限制式(14)與(15)相當類似於限制式(11)與(12)，但最大差別在於整備時

間的特性不同，在限制式(14)與(15)中，必須考量順序相依整備時間，即整備時間受工單作業順序所影響。

### 7. 定義變數 $Y$ 之限制式

$$X_{i,j,k} \leq M \times Y_{i,j,k} \quad \forall i, j, k \quad (16)$$

限制式(16)表當工單  $i$  之子批於階段  $j$  之機台  $k$  的生產比率 ( $X_{i,j,k}$ ) 大於 0 時，則表示工單  $i$  於階段  $j$  之機台  $k$  上加工，即變數  $Y_{i,j,k}$  將等於 1。

### 8. 變數範圍之限制式

$$Y_{i,j,k} \in \{0,1\} \quad \forall i, j, k \quad (17)$$

$$W_{i,j,k,l} \in \{0,1\} \quad \forall i, j, k, l \quad (18)$$

$$S_{i,u,j,k} \in \{0,1\} \quad \forall i, u, j, k \quad (19)$$

$$X_{i,j,k} \geq 0 \quad \forall i, j, k \quad (20)$$

$$C_{i,j,k} \geq 0 \text{ (integers)} \quad \forall i, j, k \quad (21)$$

限制式(17)、(18)與(19)分別定義決策變數  $Y_{i,j,k}$ 、 $W_{i,j,k,l}$  及  $S_{i,u,j,k}$  為二元變數，其值為 0 或 1。限制式(20)定義決策變數  $X_{i,j,k}$  為非負之連續變數。限制式(21)定義決策變數  $C_{i,j,k}$  為非負之整數變數。

## 3.5 多階平行機台之基因演算法說明

本研究所探討的為 HFS 系統中完全平行機台下的生產環境，不但要考慮工單生產順序，還同時需考慮分配各工單至機台與機台數量的問題，因此本研究發展透過基因演算法的求解方式，針對太陽能電池製造產能規劃問題，對產能規劃做最有效的配置，以使整體產能規劃的最長完工時間最小化。圖 3.20 為本研究提出之基因演算法之細部流程，先產生初始解作為

母代，然後計算這些初始解的適應函數值，接著判斷是否達到終止條件，本研究終止條件設定為世代數，達到設定之世代數後則停止搜尋，如果沒有，則進行選擇及複製，以保留母代中表現較好的染色體至新的子代族群，接著透過交配與突變的機制，產生新的子代，而本研究所設計之染色體編碼方式為混和兩種編碼方式，二進位編碼(Binary encoding)與順序型編碼(Order-based encoding)，因此在交配與突變時，因編碼的本質不同，無法進行相同的交配與突變機制，所以分別採用不同的機制，來進行演算的流程，產能足夠的子代後，再進行適應函數值的計算，執行到設定之演化世代之上限，則停止演算法。

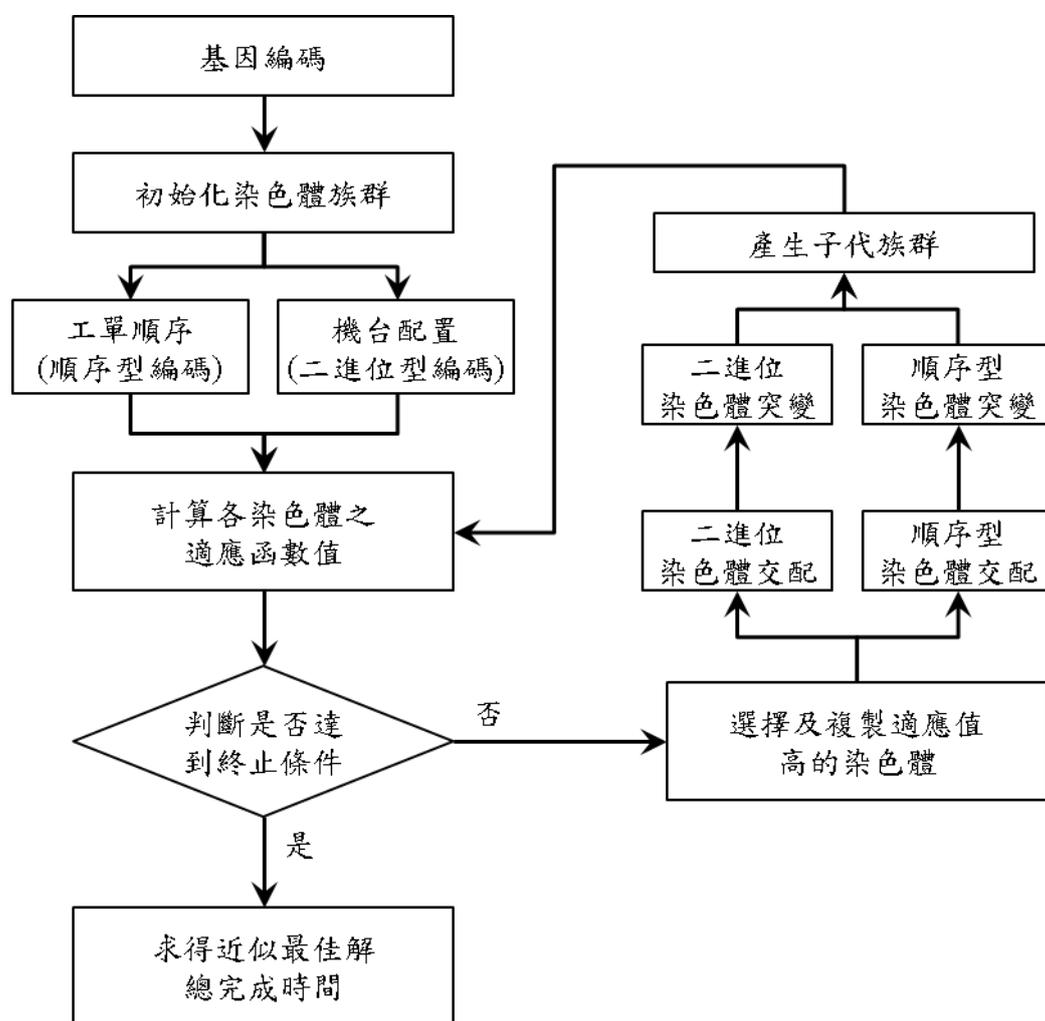


圖 3.20 基因演算法流程圖

### 3.5.1 染色體設計與解碼

因本研究之目的，為找到各工單的生產順序與機台配置的分佈情況，

所以本研究所採用的編碼形式，如圖 3.21 所示，整個染色體由二進位型與順序型所組成，順序型代表工單的生產順序，如 Order 1 的 3，表示 Order 1 至 Order  $n$  中，為第三順位加工，Order 2 則表示第二順位加工之意；而二進位型，則表示機台的分配情況，如 Order 1 的 Stage 5 其二進位配置為 110，則表示在 Stage 5 有三台機器，機器 1 與機器 2 進行加工之動作，機器 3 則不進行加工，或 Order 2 的 Stage 6 為 111，表示此製程的三部機器接進行加工，依此類推；因此根據欲規劃工單之數量，來產生依據此編碼方式的染色體，將這些染色體集成族群。

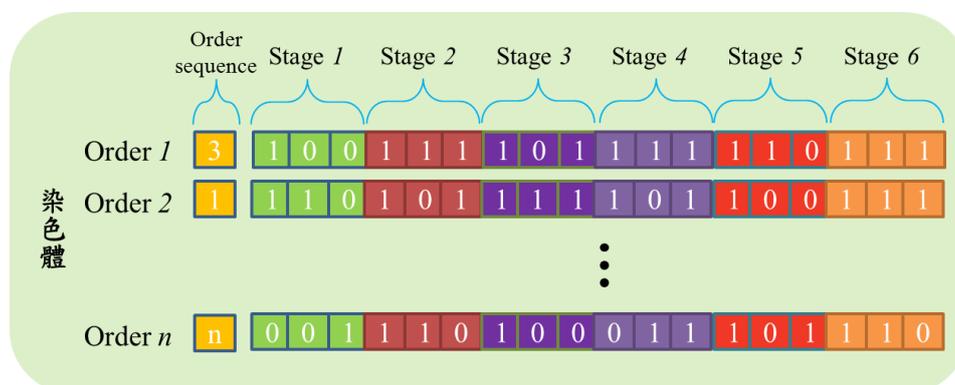


圖 3.21 本研究所採用染色體之編碼方式

### 3.5.2 初始化染色體族群

基因演算法的第一步，需根據編碼的方式隨機產生合適的母代族群，而在本研究的太陽能電池製程特性考量上，有專用機台之限制，如圖 3.22 所示，因此在隨機產生之染色體數時，必須針對工單之產品別，單晶矽與多晶矽之不同，規劃出合理的染色體，以本研究之假設，專用機台製程為 Stage 1，可加工多晶矽太陽能電池為機台 1 與機台 2，機台 3 為加工單晶矽太陽能電池之機台，因此必須針對此因素，限制染色體的配置狀況，以產生合理的染色體與數量。

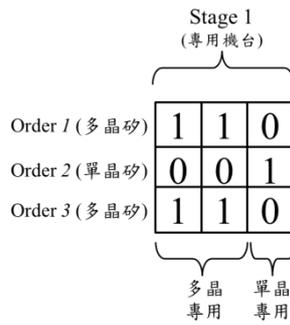


圖 3.22 專用機台限制

### 3.5.3 計算染色體之適應函數值

適應值的設計主要是用於搜尋最佳解的依據，即適應值決定了每一個族群中染色體適應環境的能力，合適的適應值往往可以將染色體的優劣比較出來。而本研究的適應值表示，依據染色體內的基因值，計算其 Makespan，最短的 Makespan 則為最佳的染色體，代表為最佳機台配置下的生產排程結果。因此，本研究在適應值的計算上，係採取產能堆疊的方式，首先透過順序型的基因編碼，或許各工單的生產順序，在參照二進位編碼所產生的機台配置，透過模擬生產來推演整個工單的總加工時間，最後結束工單的時間，則代表出此染色體的目標函數。本研究針對產能模擬的過程，試以一個詳細的流程範例描述，如 3.6.3.1 小節所述。

#### 3.5.3.1 適應值計算

本小節將透過一個範例，作為說明本研究透過撰寫之程式語言的計算後，所求得之適應值，其範例如圖 3.23 所示，共包含三張訂單，Order sequence 欄位表示生產順序，Stage 1 至 Stage 6 的欄位中，表示此訂單在該 Stage 中的機台配置狀況，而表 3.1 則包含製程的加工時間、設置時間與訂單的詳細資訊，作為本範例說明之設定。以下則以此案例進行適應值計算之步驟說明：

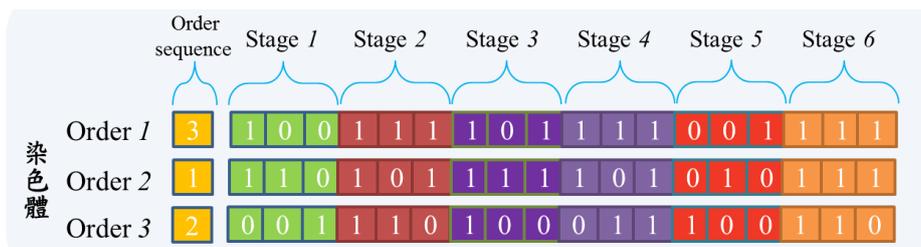


圖 3.23 適應值模擬之案例

表 3.1 適應值案例之各細部參數

製程	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
加工時間	3	3	3	6	3	3
設置時間	0	0	0	0	5	1

(a) 製程加工時間參數

訂單編號	需求量	晶矽類型	Busbar數
1	12	多晶矽	2
2	18	多晶矽	2
3	6	單晶矽	3

(b) 訂單產品需求

符號說明：

- $i$  : 訂單的數量，其中工單的編號為下標  $i=1, 2, \dots, I$
- $u$  : 工單的數量，其中工單的編號為下標  $u=1, 2, \dots, I$
- $j$  : 加工階段的數量，其中加工階段的編號為下標  $j=1, 2, \dots, J$
- $M_j$  : 在加工階段  $j$  機台的數量，其中機台的編號為下標  $k=1, 2, \dots, M_j$
- $D_i$  : 訂單  $i$  的需求數量
- $p_{i,j}$  : 訂單  $i$  於階段  $j$  的單位加工時間
- $T_{i,j}$  : 訂單  $i$  在第  $j$  階段中的總完成時間
- $S_{B_i, B_u}^{SD}$  : 工單  $i$  為工單  $u$  之前一個加工作業時，工單  $u$  所需之相依整備時間
- $B_i$  : 訂單  $i$  所需生產之品項的電極數目 (2 Busbar、3 Busbar)
- $S_i^{SI}$  : 訂單  $i$  所需之獨立整備時間
- $C_{\max}$  : 最大完工時間
- $C_{i,j,k}$  : 訂單  $i$  於階段  $j$  之機器  $k$  的完工時間
- $M_k^{record}$  : 機器  $k$  的加工時間紀錄
- $Q_k^{type}$  : 紀錄機台  $k$  所加工之產品型態 (2 表 2 Busbar ; 3 表 3 Busbar)

### 適應值計算步驟：

Step 1：載入族群染色體的配置狀況，包含各訂單之生產順序與機台配置情況。

Step 2：排列各訂單之生產順序。

Step 3：進行第一階段的產能計算。

Step 3.1：載入第一生產順位之訂單  $i$ ，依照機台配置，計算訂單  $i$  於各機台之完工時間為

$$C_{ijk} = \frac{D_i}{1_j} \times p_{i,j} \quad (21)$$

Step 3.2：記錄訂單  $i$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i,1,k} \quad (22)$$

Step 3.3：計算訂單  $i$  在第 1 階段的總完工時間

$$T_{i,1} = \max(C_{i,j,k})_j \quad (23)$$

由圖 3.23 的範例來看，Order 2 為第一優先生產之訂單，其機台配置為 Stage 1 的機台 1 與機台 2，因此將此訂單分成相同的兩個批次，每一批次需求量為 9 個單位，而每一單位所需之加工時間為 3，因此每一批次所需之處裡時間為  $27(9 \times 3)$  個單位時間，由圖 3.24 的甘特圖所見，可計算出 Order 2 在 Stage 1 時，其完工時間為 27 單位時間。

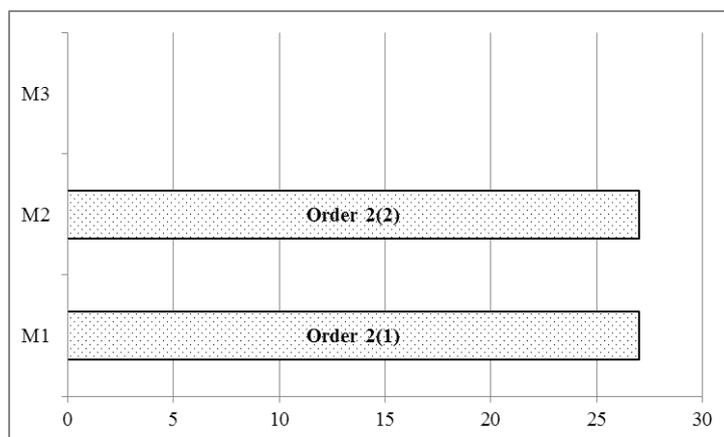


圖 3.24 第一順位之訂單在 Stage 1 之計算適應值甘特圖

Step 3.4：計算下一順位之訂單  $i'$ ，依照機台配置，計算訂單  $i'$  於各機台之完工時間為

$$C_{i',j,k} = M_k^{record} + \frac{D_{i'}}{1_j} \times p_{i',j} \quad (24)$$

Step 3.5：記錄訂單  $i'$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i',1,k} \quad (25)$$

Step 3.6：計算訂單  $i'$  在第 1 階段的總完工時間

$$T_{i',1} = \max(C_{i',j,k})_j \quad (26)$$

Step 3.7：重複 Step 3.4~Step 3.6，直到所有訂單皆計算完畢。圖 3.25 所見，模擬將所有訂單，根據圖 3.23 的案例基因碼，依照機台配置結果，搭配訂單生產順序，並經由上述之步驟，計算各訂單之所需加工時間與產能堆疊，透過甘特圖即可清楚掌握各訂單在各機台上的開完工時間。

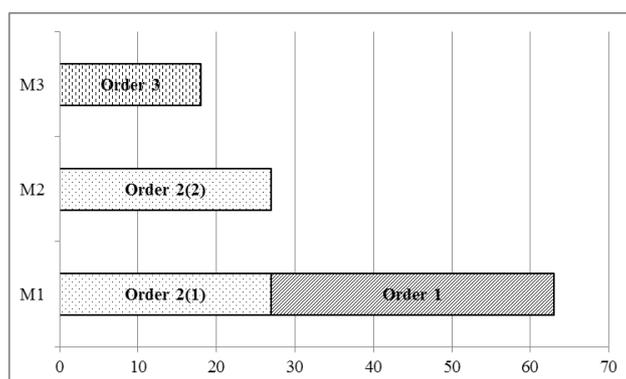


圖 3.25 所以訂單在 Stage 1 之適應值計算甘特圖

Step 4：進行第二階段至第四階段的產能計算。

Step 4.1：重新調整各訂單在各階段中之生產順序，因各訂單之需求水準不盡相同，導致每階段所需加工時間均不相同，因此在階段移轉的部分，採用為訂單最早完工優先(Order early finish firs; OEFF)法則，其 OEFF 法則如圖 3.26 所示，Order 2 因在 Stage  $m$  加工，比 Order 1 提早完工，所以根據 OEFF 法則，將 Order 2 移至 Stage  $m+1$  進行加工。

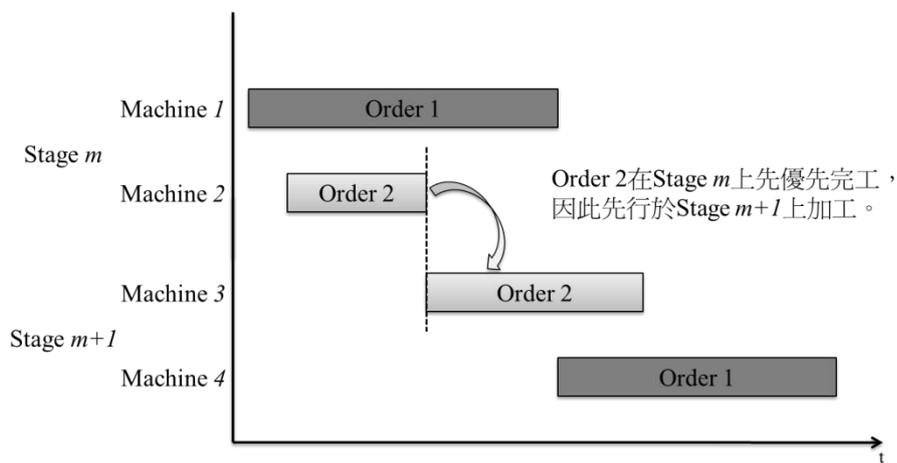


圖 3.26 訂單最早完工優先法則

Step 4.2：載入第一生產順位之訂單  $i$ ，依照機台配置，計算訂單  $i$  於各機台之完工時間為

$$C_{i,j,k} = \frac{D_i}{M_j} \times p_{i,j} \quad (27)$$

Step 4.3：記錄訂單  $i$  在  $k$  機台之完工時間  $M_k^{record} = C_{i,j,k}$ 。

Step 4.4：計算訂單  $i$  在第  $j$  階段的總完工時間

$$T_{i,j} = \max(C_{i,j,k})_j \quad (28)$$

Step 4.5：計算下一順位之訂單  $i'$ ，依照機台配置，計算訂單  $i'$  於各機台之完工時間為

$$C_{i',j,k} = M_k^{record} + \frac{D_{i'}}{M_j} \times p_{i',j,k} \quad (29)$$

Step 4.6：記錄訂單  $i'$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i',j,k} \quad (30)$$

Step 4.7：計算訂單  $i'$  在第  $j$  階段的總完工時間

$$T_{i',j} = \max(C_{i',j,k})_j \quad (31)$$

Step 4.8：重複 Step 4.5~Step 4.7，直到所有訂單皆計算完畢。圖 3.27 所見，模擬將所有訂單，根據圖 3.23 的案例基因碼，依照機台配置結果，搭配訂單生產順序，並經由上述之步驟，計算各訂單之所需加工時間與產能堆疊，透過甘特圖即可清楚掌握各訂單在各機台上的開完工時間。

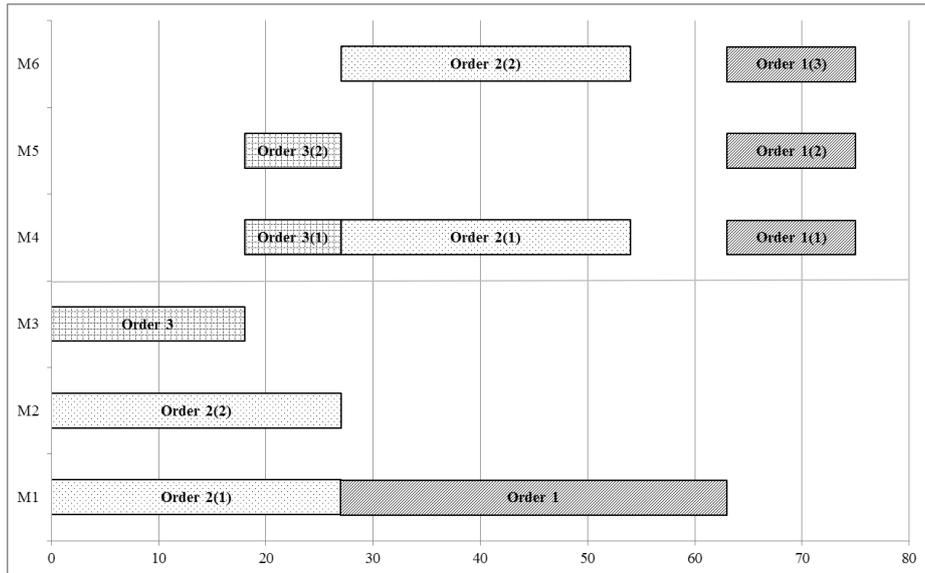


圖 3.27 所有訂單在 Stage 2 之適應值計算甘特圖

Step 4.9：重複 Step 4.1~Step 4.8，直到計算至 Stage 4 完為止，如圖 3.28 所見，執行上述之步驟與法則，即可準確計算出到 Stage 4 為止之各訂單在每個機台的完工時間，以便於計算下一階段製程用。

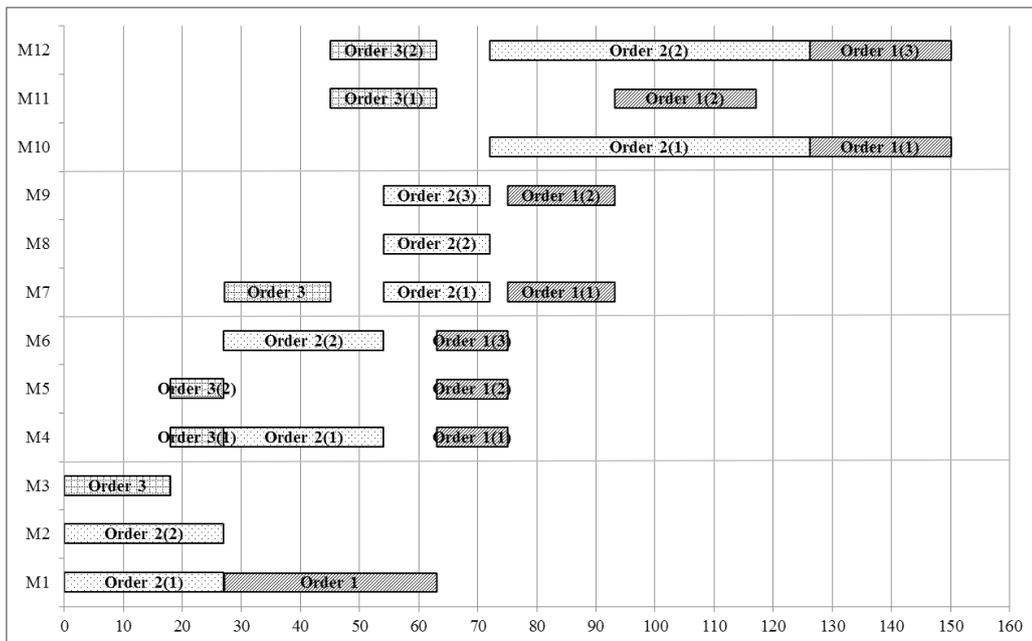


圖 3.28 所有訂單在 Stage 5 前之適應值計算甘特圖

Step 5：進行第五階段製程的產能計算(具獨立整備時間因素)。

Step 5.1：重新調整各訂單在前階段中之生產順序。

Step 5.2：載入第一生產順位之訂單  $i$ ，依照機台配置，計算訂單  $i$  於各機台之完工時間為

$$C_{i,j,k} = \frac{D_i}{5_j} \times p_{i,j} + S_i^{SI} \quad (32)$$

Step 5.3：記錄訂單  $i$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i,5,k} \quad (33)$$

Step 5.4：計算訂單  $i$  在第 5 階段的總完工時間

$$T_{i,5} = \max(C_{i,j,k})_j \quad (34)$$

Step 5.5：計算下一順位之訂單  $i'$ ，依照機台配置，計算訂單  $i'$  於各機台之完工時間為

$$C_{i',j,k} = M_k^{record} + \frac{D_{i'}}{5_j} \times p_{i',j} + S_{i'}^{SI} \quad (35)$$

Step 5.6：記錄訂單  $i'$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i',5,k} \quad (36)$$

Step 5.7：計算訂單  $i'$  在第 5 階段的總完工時間

$$T_{i',5} = \max(C_{i',j,k})_j \quad (37)$$

Step 5.8：重複 Step 5.5~Step 5.7，直到所有訂單皆計算完畢。

以圖 3.29 所見，在 Stage 5 時，因此階段有獨立整備時間，所以在計算該訂單之所需作業時間時，應另加上整備時間，以確保此適應值之計算符合本研究問題之考量因素。

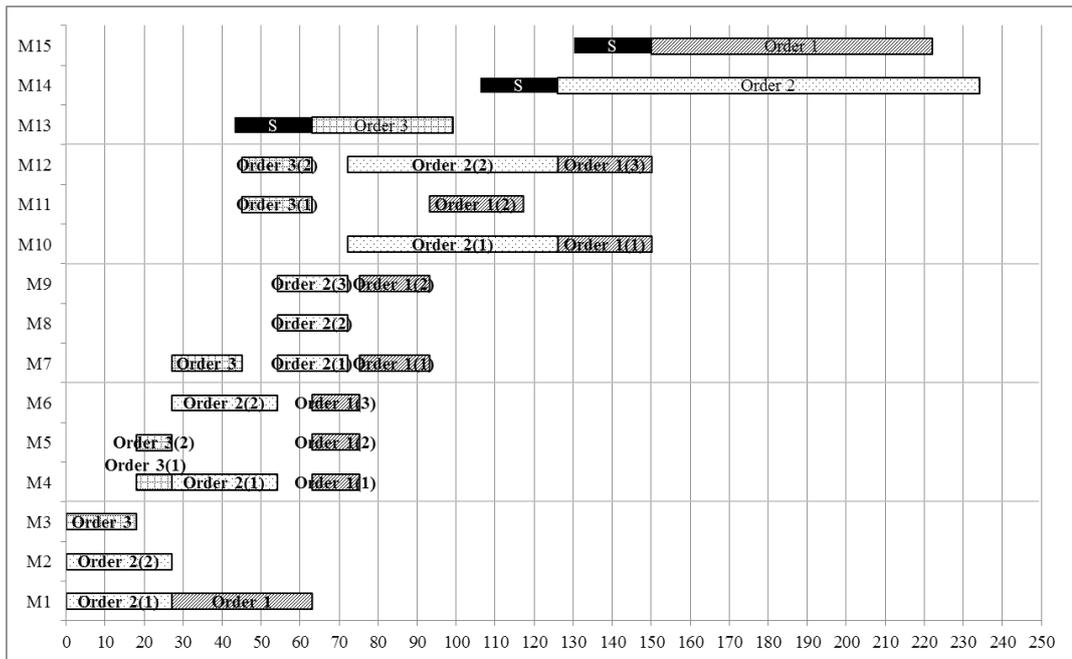


圖 3.29 所有訂單在 Stage 5 之適應值計算甘特圖

Step 6：進行第六階段製程的產能計算(具相依整備時間因素)。

Step 6.1：重新調整各訂單在前階段中之生產順序。

Step 6.2：載入第一生產順位之訂單  $i$ ，依照機台配置，計算訂單  $i$  於各機台之完工時間為

$$C_{i,j,k} = \frac{D_i}{6_j} \times p_{i,j} + S_i^{SD} \quad (38)$$

Step 6.3：記錄訂單  $i$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i,6,k} \quad (39)$$

Step 6.4：記錄訂單  $i$  在  $k$  機台之產品型態

$$Q_k^{type} = B_i \quad (40)$$

Step 6.5：計算訂單  $i$  在第 6 階段的總完工時間

$$T_{i,6} = \max(C_{i,j,k})_j \quad (41)$$

Step 6.6：計算下一順位之訂單  $i'$ ，依照機台配置，計算訂單  $i'$  於各機台之完工時間。

Step 6.6.1：先判斷是否  $Q_k^{type} = B_{i'}$ ，若為是，則

$$C_{i',j,k} = M_k^{record} + \frac{D_{i'}}{6_j} \times p_{i',j} \quad (42)$$

若為否，則

$$C_{i',j,k} = M_k^{record} + \frac{D_{i'}}{6_j} \times p_{i',j} + S_{B_i, B_u}^{SD} \quad (43)$$

Step 6.7：記錄訂單  $i'$  在  $k$  機台之完工時間

$$M_k^{record} = C_{i',6,k} \quad (44)$$

Step 6.8：記錄訂單  $i'$  在  $k$  機台之產品型態

$$Q_k^{type} = B_{i'} \quad (45)$$

Step 6.9：計算訂單  $i'$  在第 6 階段的總完工時間

$$T_{i',6} = \max(C_{i',j,k})_j \quad (46)$$

Step 6.10：重複 Step 6.6~Step 6.10，直到所有訂單皆計算完畢。

以圖 3.30 所見，在 Stage 6 時，因此階段有相依整備時間，所以在計算該訂單之所需作業時間時，應另加上整備時間，而圖中 M16、M17 與 M18 屬於 Stage 6 之作業機台，Order 3 與 Order 1 分別為 3 Busbar 與 2 Busbar 的產品型態(表 3.1)，因此當 Order 3 之生產順序在 Order 1 之前，則 Order 1 就必須增加整備時間，而 Order 2 為 2 Busbar 之產品，與 Order 1 相同，就不須整備，可直接加工處理，以確保此適應值之計算符合本研究問題之考量因素。

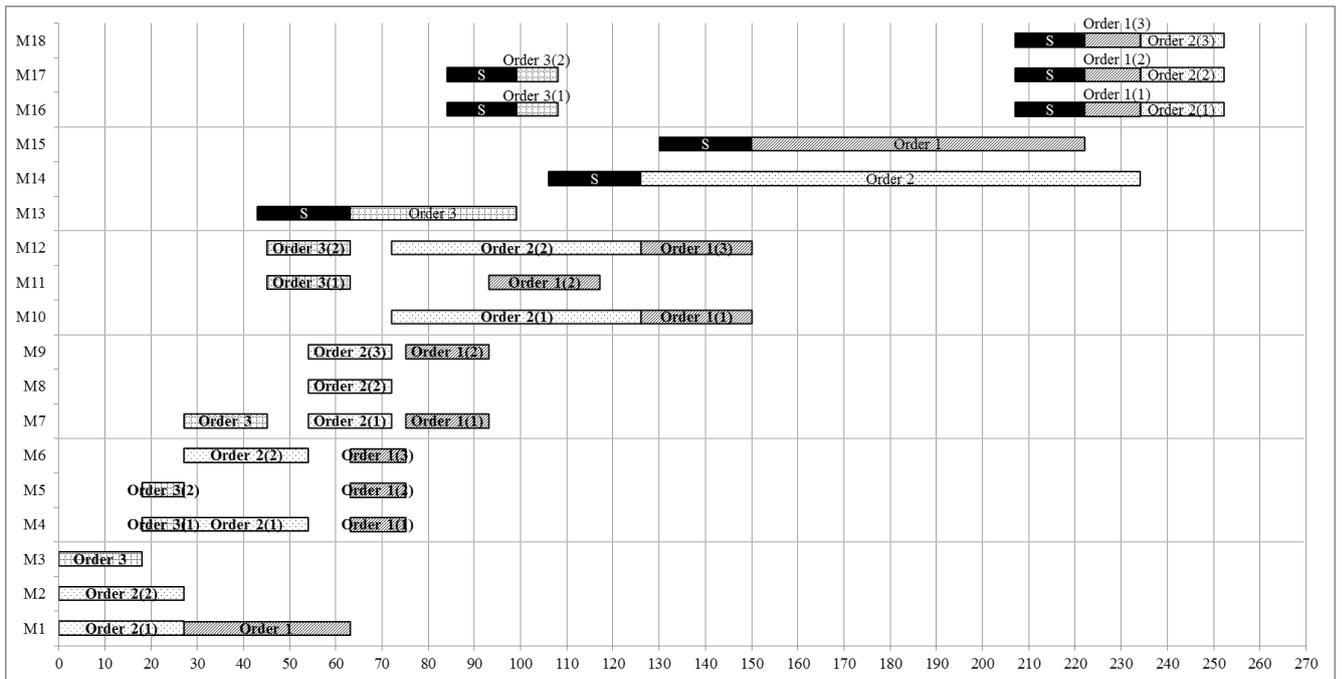


圖 3.30 所有訂單在 Stage 6 之適應值計算甘特圖

Step 7：從所有的訂單中選取最大的完工時間 $C_{max}$ ，即代表此染色體的適應值。

透過上述之詳細步驟與方法，準確計算出此染色體編碼下產能配置結果，擷取出最大完工時間，即代表此染色體的適應值，以做為評估染色體之績效使用。

### 3.5.4 選擇及複製

本研究採用的複製策略為競爭式，透過競爭式複製的方式，將母代透過比較的方式產生至新的子代中，其步驟說明如下：

1. 依照適應值的大小以遞減的方式進行排序。
2. 將排序好的母代族群中的前半部隨機選擇  $n$  條染色體。
3. 將選好的  $n$  條染色體中的適應值進行比較，具有較佳適應值的染色體將被複製至子代中。
4. 重複步驟 2，直到子代族群中設定的染色體的數量完全被複製完成為止。

### 3.5.5 交配

本研究所採用的交配方式有兩種，分別為二進位編碼所使用的雙點交配(Two point crossover)(Lin & Yao, 1997)與順序型編碼用的位置交配(Position based crossover)(Syswerda, 1989)。首先選取將進行交配的母代染色體，隨機產生  $n$  個族群位置，找出  $n$  個族群位置中具有較佳適應值者，將被挑選出來當作執行交配的母代，此為競爭式選擇策略，透過較佳的母代交配產生較佳的子代，而交配方式步驟如下：

#### 1. 位置交配：

首先隨機產生數個基因的位置，如圖 3.31 所示，隨機產生的點為第 2 個、第 5 點、第 6 點與第 9 點，將選取的第一個母代染色體，其將上述之位置內的值，直接遞移到新的子代相同的位置上，而第二條母代染色體，則保留與第一條母代染色體中，隨機選擇點內非相同的值，依序從左至右，插入產生的子代染色體中，得以形成完整的子代。

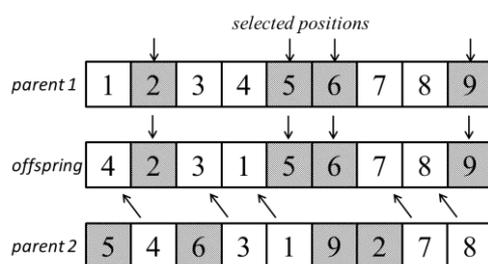


圖 3.31 位置交配圖

#### 2. 雙點交配：

一開始決定兩個交配點，如圖 3.32 所示，產生第一點為第 6 個基因與第 12 個基因為交配點，下一步則將兩交配點間所有的基因和對方的染色體互換，產生兩條新的子代染色體，完成交配動作。

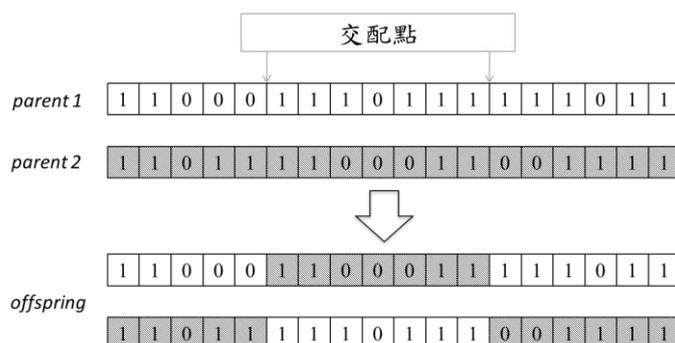


圖 3.32 雙點交配圖

### 3.5.6 突變

突變是避免在演化的過程中，不會因為複製或交配等過程中而遺失了一些有用的資訊，也是用來跳脫區域最佳解的一個重要指標，使得搜尋空間的範圍更為廣大，得以逼近全域最佳解。本研究採用的方式分別為鄰近雙點突變(Adjacent two job change)(Engin et al., 2011)(圖 3.33)與位元突變(Bit flip)(Luo et al., 2001)的突變方式(圖 3.34)，作為順序型與二進位型編碼的突變機制，其詳細突變機制如下：

#### 1. 鄰近雙點突變：

首先抽取將要突變的母代染色體，在此染色體上隨機產生要突變的基因位置，產生後與左邊鄰近的基因位置，相互調換，進而產生新的子代，得以完成此突變機制。

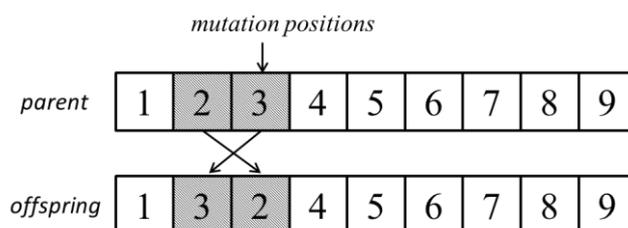


圖 3.33 鄰近雙點突變

#### 2. 位元突變：

主要方式為從挑選的染色體中，隨機選擇一個突變點，將突變點的基因改變，如 0 變成 1 或 1 變成 0，即完成突變。其突變步驟如下：

- (1) 隨機產生突變點。
- (2) 將挑選出來的染色體中，突變點的基因值，將 0 變成 1 或 1 變成 0，即完成突變運算。

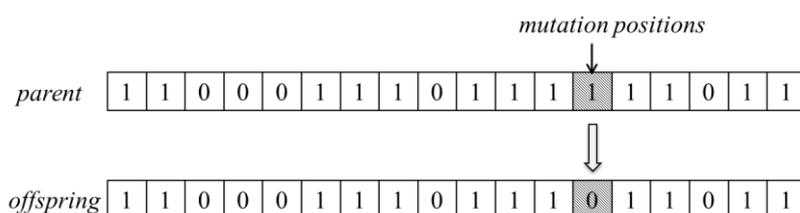


圖 3.34 位元突變

### 3.5.7 終止條件

通常演算法的運算終止條件會根據各問題或演算法之特性而有所不同，其大致上可分為三類：

1. 運算總時間：設定一固定時間，當求解時間超過設定時間時，終止條件成立，停止運算。
2. 運算總回合數：設定一固定回合數(世代)，當求解總回合數超過設定之回合數，終止條件成立，則停止運算。
3. 未改善之回合數：設定一回合數，當求解目標值持續未改善至設定之回合數，終止條件成立，停止運算。

為了避免求解時間過長，須設定某一終止條件，以此終止條件執行時，採目前所搜尋到之最好的目標值，為最終所求得之結果，而本研究採取運算總回合數為演算法的終止條件。

## 第四章 績效評估

本章節透過實驗來評估本方法的績效，定義一組共同的問題，利用本研究之基因演算法進行求解，透過與混整數線性規劃求解的結果，進行比較，分析由本研究所提出之方法的結果品質與求解時間，進行探討。本章節區分為四節，第 4.1 節說明整體實驗的環境建構，第 4.2 節進行與數學模式的比較，呈現其結果與分析，第 4.3 節為探討三因子的實驗設計，第 4.4 節進行與廠商現行模式的方法比較，並呈現結果，第 4.5 節為此章節之小結。

### 4.1 實驗方式與環境建構

#### 4.1.1 實驗環境

##### 1. 需求資訊：

本研究考量的需求資訊除各訂單之產品品項需求量外，每個訂單各包含兩種規格，分別為晶矽的種類與 Busbar 的數目，如表 4.1 所示，晶矽分為單晶與多晶，Busbar 數量分為 2 與 3，其產品品項共四種組合。另外，訂單之需求量、晶矽種類與 Busbar 數目皆透過隨機的方式，來決定各訂單之需求量與產品類型，而本研究將採取訂單數為 3、5、10、20 與 50 等水準，進行實驗分析。

表 4.1 產品類型資訊

產品	晶矽種類	Busbar數量
太陽能電池	多晶矽	2 busbar
		3 busbar
	單晶矽	2 busbar
		3 busbar

##### 2. 製程資訊：

本研究探討為 HFS 生產系統，此系統具有多階段製程之特性，因此為了探討製程數的多寡在規劃時的影響，本研究針對不同製程數的水準，作為本研究實驗時的評估依據，分別在 4 階、6 階與 8 階不同的階段下，進

行實驗，並評估實驗結果之效益。

### 3. 機台資訊：

HFS 環境除具有多階段製程外，其每一階段之機台數也是影響整個排程的關鍵因素之一，機台數量的多寡，相對就是產能供給的水準；因此，本研究為探討不同機台數量之水準，分別在機台數為 3 台、5 台、8 台、10 台與 15 台的不同情況下，進行實驗，並評估實驗結果之效益。

### 4. 進行實驗之作業系統之環境：

本實驗之作業系統為 Windows XP Professional SP3，CPU 為 Intel® Core™2 Quad Processor Q8300 2.5GHz，1.87GB RAM，求解工具為 Microsoft Visual C++ 2008。

綜合上述，本研究將上述之各影響因素，整理如表 4.2，將此實驗共區分為兩種等級，區別在於問題的大與小，共 10 組問題，每一組問題有各自的需求水準(Orders)、供給水準(Machines)與製程數水準(Stages)，其問題規模由小至大，而變數與限制式的數量，也逐筆增加，亦表示問題相對更複雜；因此，使用本研究所提出多階多機台加工之基因演算法，與混整數線性規劃模型採用 IBM ILOG CPLEX Optimizer 軟體進行求解的比較，評估以混整數線性規劃求解之方式，找出最佳解與本研究所提出之方法的差距。

表 4.2 實驗問題

Data Size	Problem Number	Problem Size (orders, machines, stages)	No. of continuous variables	No. of binary variables	No. of integer variables	No. of total variables	No. of constraints
Small	1	(3, 3, 4)	36	180	37	253	770
	2	(3, 5, 4)	60	420	61	541	1604
	3	(3, 8, 4)	60	360	61	481	1522
	4	(5, 5, 6)	150	1200	151	1501	4632
	5	(5, 8, 6)	240	2640	241	3121	9431
Large	6	(10, 8, 6)	480	6480	481	7441	23662
	7	(10, 10, 6)	600	9300	601	10501	32962
	8	(20, 10, 8)	1600	31280	1601	34481	117322
	9	(20, 15, 8)	2400	61200	2401	66001	214962
	10	(50, 15, 8)	6000	24300	6001	255001	699626

### 4.1.2 基因演算法參數

基因演算法的參數主要分為四個部份，每世代染色體數目、任兩染色體發生交配的機率、每世代各染色體發生突變的機率及選擇策略。基因演算法各種參數設定因子及其水準如表 4.3：

表 4.3 基因演算法參數設定

The GA parameters	Value
Population size	700
Crossover rate	0.8
Mutation rate	0.1
Selection Strategy	Tournament selection

## 4.2 本研究之模式與混整數線性規劃模型比較

承 4.1 章節所提之各種實驗環境設計及演算法參數，將執行完成表 4.2 中的所有問題，分別採取使用 CPLEX 求解的方式，與本研究模式使用 Visual C++ 來執行。CPLEX 求解的優點為能求解出最佳解，但是前文已提到本研究之問題，為 *NP-Complete* 的問題，若問題的規模較小，就能在短時間內求解出最佳解來，相對的，當問題規模逐漸變大時，求解難度提高，可能需要長時間來進行求解，甚至有無法求解的情況，為了避免過長的求解時間，必須對此有所限制；而 CPLEX 在演算時，因採用分支界定法 (Branch-and-Bound; B&B) 的方式，所以需要使用電腦本身的記憶體空間，支持整個演算進行，問題越大，所需的記憶體空間也就越大，當記憶體不足時，CPLEX 則會停止運算 (Runs out of memory)，雖無法求得最佳解，但會提供一組在停止運算前的近似解；因此，若發生此情況，本研究則擷取在停止運算後近似解，作為此實驗的評估標準。

另外，本研究模式因採用基因演算法，而在使基因演算法求解問題時，每次的結果不盡相同，所以本研究在每一個問題下，均執行十次，分別紀錄下來，找出十次演算中之最大值、最小值與平均值，進行與最佳解的評估，而該如何評估的兩種方法的差距，本研究採用 Chen、Lin 與 Fang(2010) 所使用的方式，如公式(47)所示：

$$\text{Solution Gap}(\%) = \frac{S-B}{B} \times 100 \quad (47)$$

公式(47)中，S則為本研究模式所求解之結果；B為由 CPLEX 求解之最佳解，假如超出記憶體之運算，則使用近似解做為比較的依據。透過 47 式的計算，將能評估本研究模式與最佳解之間的差異，除了評估求解的品質外，求解之時間，也是本實驗的評估重點。

#### 4.2.1 結果比較

表 4.4 為本次實驗的結果呈現，問題 1 與問題 2 皆能透過 CPLEX 求解出最佳解，而本研究所提出的基因演算法，在問題 1 的部分，求得之最小的解與最佳解之間無任何差距，表示本方法所求得之結果為此問題下的最佳解，若以 10 次的平均結果來看，與最佳解的差距為 0.0392%，非常的接近，即使最大的差距，也在 0.5% 以內；同樣也是能求得最佳解的問題 2，本研究模式亦能求出最佳解，平均值的差距為 0.2381%，其求解績效優異；而問題 3 之後，CPLEX 在求解過程中，皆無法求得最佳，但還是有近似解得以評估，如表 4.4 所見，本研究之方法所求得之結果，均能得到比混整數線性規劃擁有更好的求解品質；問題 6 之後，在 CPLEX 求解下，均為 Runs out of memory 的情況，也無法求得一近似解，但本研究模式皆能求得近似解。

表 4.4 本研究模式與 CPLEX 之結果與差距

Data Size	Problem Number	Problem Size (orders, machines, stages)	CPLEX Optimizer		Genetic Algorithm	
			Solution	Min. Solution (Gap %)	Average Solution (Gap %)	Max. Solution (Gap %)
Small	1	(3, 3, 4)	76,500	76,500 (0)	76,530 (0.0392)	76,800 (0.3922)
	2	(3, 5, 4)	25,200	25,200 (0)	25,260 (0.2381)	25,800 (2.3810)
	3	(3, 8, 4)	103,500*	99,300 (-4.0580)	102,210 (-1.2464)	103,500 (0)
	4	(5, 5, 6)	64,800*	60,570 (-6.5278)	63,645 (-1.7824)	66,480 (2.5926)
	5	(5, 8, 6)	98,100*	45,900 (-53.211)	49,439.2 (-49.603)	55,285 (-43.644)
Large	6	(10, 8, 6)	-	69,750	77,286	79,650
	7	(10, 10, 6)	-	59,640	71,386.9	74,031
	8	(20, 10, 8)	-	115,200	129,192.5	135,540
	9	(20, 15, 8)	-	150,701	155,553.1	165,954
	10	(50, 15, 8)	-	360,000	367,008	374,400

\* CPLEX runs out of memory

表 4.5 為兩種方法下的求解時間記錄，在 CPLEX 方面，若能順利求得最佳解，即可顯示出求解所需時間，若是求解問題過大，則會面臨 Runs out of memory 之情況，但還可取得 Runs out of memory 前的演算時間，本研究

實驗則以此時間作為評估的依據；在本研究模式方面，相同的，也分別紀錄下每組問題中，執行 10 次實驗的 Min. CPU Time、Average CPU Time 與 Max. CPU Time，從表 4.5 可看出，在問題 1 與問題 2 的情境下，CPLEX 與本研究模式都可求出最佳解的情況下，本研究模式所需之求解時間，皆遠低於 CPLEX 的求解方式；而在問題 3 到問題 5 中，透過表 4.4 與表 4.5 可看出，無論是在求解的品質與時間，相較於 CPLEX 下，皆有不錯的績效。

表 4.5 本研究模式與 CPLEX 之求解時間

Data Size	Problem Number	Problem Size (orders, machines, stages)	CPLEX Optimizer		Genetic Algorithm	
			CPU Time (second)	Min. CPU Time (second)	Average CPU Time (second)	Max. CPU Time (second)
Small	1	(3, 3, 4)	58.00	6.49	6.70	6.90
	2	(3, 5, 4)	1257.00	9.47	9.79	10.06
	3	(3, 8, 4)	>2334.69*	75.27	79.28	81.49
	4	(5, 5, 6)	>5140.52*	351.79	367.46	384.60
	5	(5, 8, 6)	>977.22*	263.17	272.99	283.90
Large	6	(10, 8, 6)	>2087.50*	385.25	397.68	408.74
	7	(10, 10, 6)	>1531.16*	476.65	498.59	526.70
	8	(20, 10, 8)	>3107.84*	1393.76	1424.00	1469.74
	9	(20, 15, 8)	>2098.42*	1806.99	1853.44	1913.32
	10	(50, 15, 8)	-	2338.27	2438.58	2600.66

\* CPLEX runs out of memory

由圖 4.1 所見，可看出兩個方法在求解時間上的趨勢，本研究方法的求解時間，由小規模的問題到大規模的問題，其求解時間緩慢的增加，且都遠低於 CPLEX，可看出當問題規模逐漸變大時，本研究模式所需的求解時間，並不會迅速增加，更可明確的得知本研究模式的優異之處。

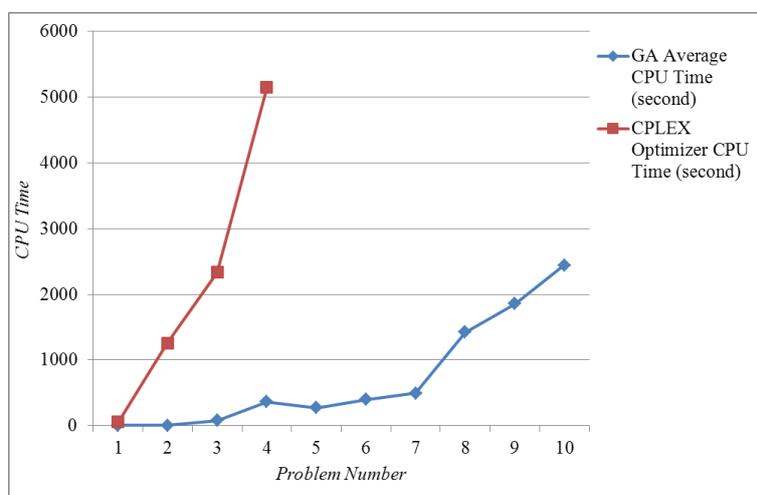


圖 4.1 兩種方法的求解時間趨勢

此外，本研究方法之 Problem1 甘特圖如圖 4.2 所示，從甘特圖可看出，本研究方法在進行排程時，每張訂單在此製程完工後，立即在下一道製程開工，整體規劃上相當緊湊，以獲取最小的 Makespan。

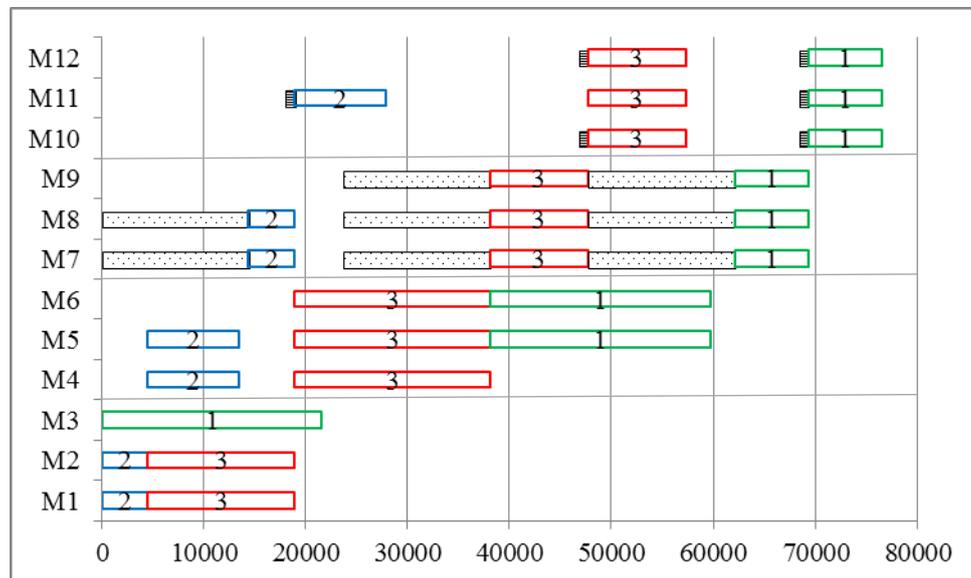


圖 4.2 Problem1(3, 3, 4)之甘特圖

## 4.3 實驗設計

本小節為探討 4.1 節中，10 個問題中的三個主要因子對於 Makespan 與 CPU Time 的影響，分別是需求水準(Orders)、供給水準(Machines)與製程數水準(Stages)。

### 4.3.1 實驗規劃說明

#### 1. 實驗設計

採 3 因子 2 水準之全因子實驗，且每個實驗重複 2 次，實驗順序為完全隨機方式，其各因子資訊與水準如表 4.6 所示，表 4.7 為此實驗設計的實驗組數與完全隨機方式下的實驗順序。

#### 2. 原因

- (1) 全因子實驗：考慮實驗結果可全部涵蓋各種可能，可避免交絡 (Confounding)，及因子的效應無法與其它的效應明確分辨。
- (2) 重複 2 次：為了找出隨機誤差項，以確認因子是否為主要因子。

表 4.6 控制因子及水準

Control Factor			Level	
			Level 1	Level 2
1	Orders	A	3	50
2	Machines	B	3	15
3	Stages	C	4	8

表 4.7 實驗設計表

Machines	Orders	3		50	
	Stages	4	8	4	8
3	1* (5)**	5 (11)	2 (1)	14 (2)	
	9 (12)	13 (16)	10 (14)	6 (9)	
15	3 (4)	7 (8)	4 (3)	16 (6)	
	11 (7)	15 (15)	12 (13)	8 (10)	

註：\*表實驗編號，\*\*表真正實驗順序

### 4.3.2 Makespan 數據分析

進行實驗獲取附錄一之實驗數據後，本研究針對三個因子對於 Makespan 的影響進行變異數分析(ANOVA)，使用 MINITAB R14 軟體計算所得之 ANOVA 表如表 4.8 所示，其中因子 Orders、因子 Machines 與因子 Stages 對於 Makespan 均具有顯著影響，因此可證實本研究所挑選的因子中，對於測定值 Makespan 均為相當重要，而以柏拉圖圖 4.3 所示，其中因子 Orders 對於 Makespan 為具有最大的影響程度，其次是因子 Machines；而各因子間的交互作用，以因子 Orders 與因子 Machines 及因子 Orders 與因子 Stages 有顯著的影響，代表各因子間互相有關連性，證實本研究在進行績效評估時，挑選這些因子的重要性。

表 4.8 Makespan 之 ANOVA 分析表

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Orders (A)	1	8.74E+11	8.74E+11	8.74E+11	10754.13	0*
Machines (B)	1	85717493400	85717493400	85717493400	1055.17	0*
Stages (C)	1	3854050561	3854050561	3854050561	47.44	0*
Orders*Machines (A*B)	1	64402004400	64402004400	64402004400	792.78	0*
Orders* Stages (A*C)	1	2377635121	2377635121	2377635121	29.27	0.001*
Machines*Stages (B*C)	1	308669761	308669761	308669761	3.8	0.087
Orders*Machines*Stages (A*B*C)	1	20151121	20151121	20151121	0.25	0.632
Error	8	649883305	649883305	81235413		
Total	15	1.03E+12				

\* $p < 0.05$

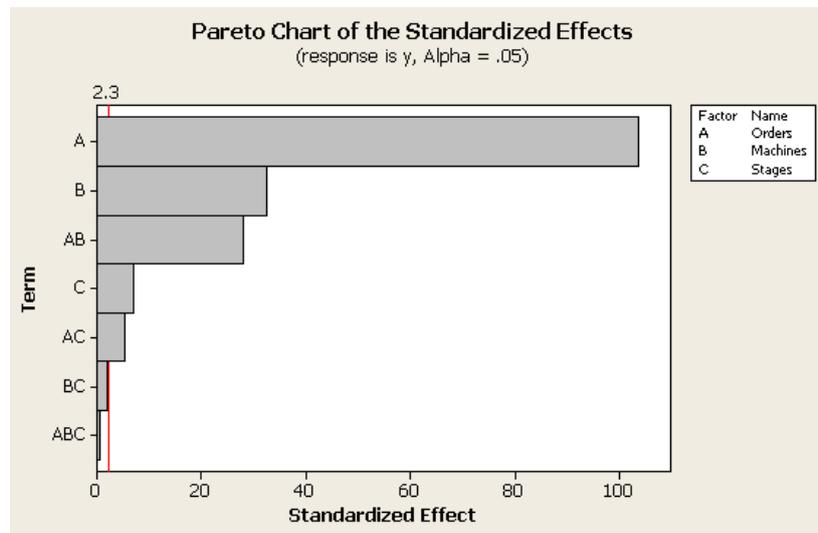


圖 4.3 Makespan ANOVA 分析之柏拉圖

### 4.3.3 CPU Time 數據分析

進行實驗獲取附錄二之實驗數據後，本研究針對三個因子對於 CPU Time 的影響進行變異數分析(ANOVA)，使用 MINITAB R14 軟體計算所得之 ANOVA 表如表 4.9 所示，其中因子 Orders、因子 Machines 與因子 Stages 對於 CPU Time 均具有顯著影響，以柏拉圖圖 4.4 所示，其中因子 Orders 具有最大的影響效益程度，因子 Machines 次之；而各因子間的交互作用，不論是雙因子間或三因子間，均具有顯著的影響，所以對於 CPU Time，因子 Orders、Machines 與 Stages，皆是影響整個演算的關鍵因素。

表 4.9 CPU Time 之 ANOVA 分析表

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Orders (A)	1	7151285	7151285	7151285	1039.87	0.00*
Machines (B)	1	3926528	3926528	3926528	570.96	0.00*
Stages (C)	1	667824	667824	667824	97.11	0.00*
Orders*Machines (A*B)	1	2792078	2792078	2792078	406	0.00*
Orders*Stages (A*C)	1	514959	514959	514959	74.88	0.00*
Machines*Stages (B*C)	1	246173	246173	246173	35.8	0.00*
Orders*Machines*Stages (A*B*C)	1	200355	200355	200355	29.13	0.001*
Error	8	55017	55017	6877		
Total	15	15554219				

\* $p < 0.05$

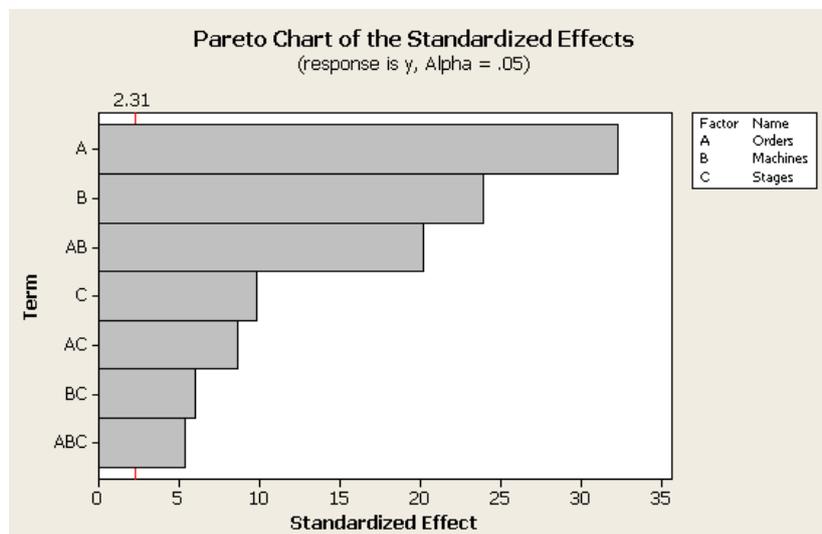


圖 4.4 CPU Time ANOVA 分析之柏拉圖

#### 4.4 本研究之模式與現況方法比較

除了探討各種不同規模下的問題外，本研究以實際案例之需求數據，比較現行以派工法則與優先權為基礎的啟發式規劃方法與本研究所提出之多階平行機台之基因演算法的規劃結果，並以 Makespan 的觀點，探討兩種方法用於訂單生產順序與機台配置在生產排程上的影響。

本實驗數據為案例公司於 99 年 10 月之實際需求數量，如表 4.10 所示，其中需求數量、Busbar 數與到期日，為案例公司的實際需求，但因該公司現況以多晶矽太陽能電池為主，雖然技術能夠生產單晶矽太陽能電池，但還在規劃及測試階段，但因應未來考量，以能夠量產單晶矽太陽能電池為

目標，因此，本研究實驗將考慮單晶矽太陽能電池，隨機抽選兩筆訂單，更改需求為單晶矽太陽能電池，作為本實驗的需求參數。

表 4.10 廠商案例實際需求資料

Orders	The Quantity Demand	Type of Silicon	Number of Busbar	Due Day
1	51000	Polycrystalline	3	10月29日
2	385000	Monocrystalline	3	10月14日
3	257000	Monocrystalline	3	10月29日
4	257000	Polycrystalline	2	10月30日
5	128000	Polycrystalline	2	10月26日
6	58000	Polycrystalline	3	10月27日
7	51000	Polycrystalline	2	10月11日
8	137000	Polycrystalline	2	10月8日
9	482000	Polycrystalline	3	10月25日

#### 4.4.1 現行啟發式規劃說明

本節說明案例公司現行針對生產排程的規劃方式，以訂單能準時達交第一優先，規劃人員在規劃時，都是使用以優先權(Priority-based)與最早到期日(Earliest Due Date, EDD)為基礎的啟發式規劃方法進行規劃，以滿足各顧客之產品訂單交期設定為優先考量，其生產規劃流程如圖 4.5 所示，大致上分為排定訂單優先權、物料配置與生產規劃三個步驟：

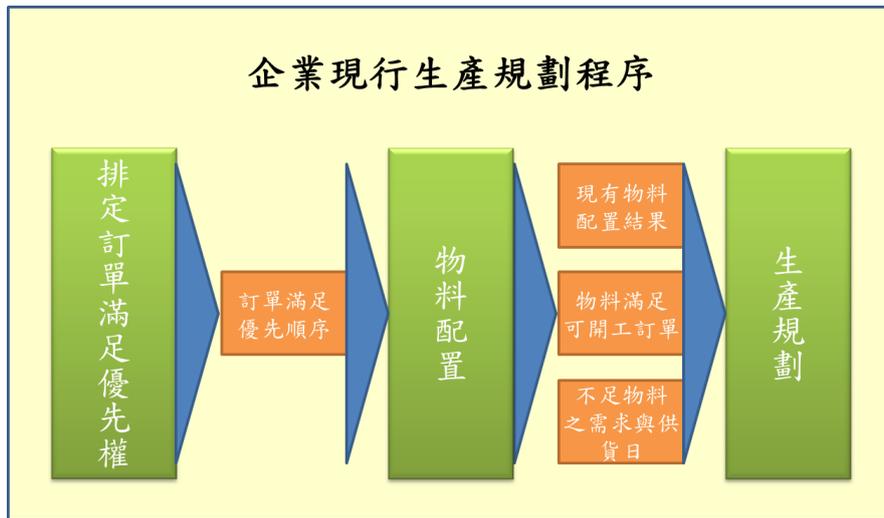


圖 4.5 生產規劃流程示意圖

在晶矽太陽能電池製造業的生產規劃流程中，生管人員會先將業務手中的訂單資料，做匯總之動作，接著透過 Priority-based 啟發式規劃(見圖 4.6)來決定如何在現有產能下，排定滿足訂單的優先順序。

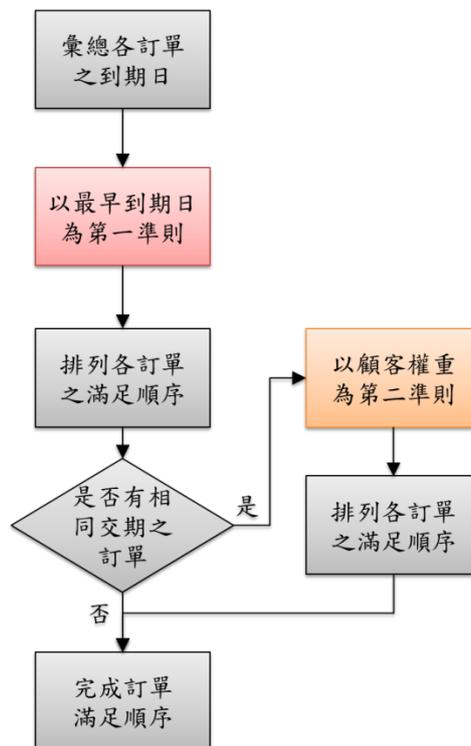


圖 4.6 Priority-based 啟發式規劃示意圖

由圖 4.6 中可知，規劃人員得到滿足訂單的優先順序後，會逐筆依照

優先順序規劃各品項所需用料的配置數量，當物料規劃完成後，即採取以一次滿足一張訂單的情況進行規劃，表示當每一製程有  $n$  台機器時，此訂單會平均切割為  $n$  個批量，同時在各機台加工，等此訂單加工完成後，緊接著製作第二張訂單的需求，如圖 4.7 所示。

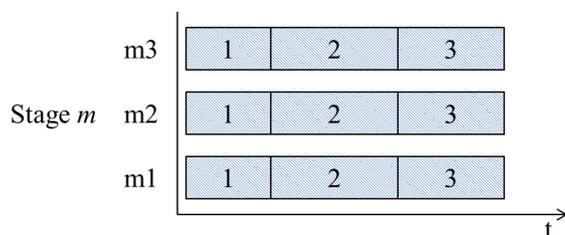


圖 4.7 現行機台配置之甘特圖

因此，案例公司先決定各訂單之投料順序，接著依照順序下發至現場開工，而機台排程的部分，採用全平行分割的排程方式，希望能在最快的時間內，完成該訂單。然而在 3.2.2 節提到，當有專用機台及設置時間的影響下，生產排程的複雜性相對提高，更以圖 3.15 為例，當考量設置時間後，若採用全平行機台分割的方式，配置到所有機台加工，其完工時間略高於非平均分割的方式，相對的機台配置的彈性也較小。有鑑於此，本研究將提出 Makespan 最小化之基因演算法，在假設條件與已知條件下，考量在現有的訂單需求及機台數量下，如何針對現有機台進行配置，以達到 Makespan 最小化之目標。

#### 4.4.2 結果比較

表 4.11 為此次實驗的結果，訂單數量如表 4.10 所示，共 9 張訂單，機台數與製程數量，皆與案例公司現況相同，有 6 道製程，每道製程裡共 3 台機台，其中在第一道製程 Texturing 機台的部分，多晶矽專用機台為 2 台，單晶矽專用機台為 1 台，也與該公司之製造現場相同。本研究之多皆平行機台之基因演算法，與 4.2 節相同，執行 10 次後，分別取出 10 次實驗中 Makespan 的最大值、最小值與平均值，結果如表 4.11 所示，而使用案例公司現行的啟發式法則，其 Makespan 結果為 5719000，而本研究方法之平均解為 5091610，由此可得知，本研究所提出之方法，優於該公司現況之方

法模式。我們更可由表 4.12 發現，透過公式(30)計算出與該公司現況方法的差距，本研究在平均值的結果表現，比公司現況方法較優了-10.9703%，即便在最大值之結果，其Makespan也縮短了-9.1292%，其中最小值之結果，Makespan 更是比該公司的方法減少了-12.3571%，圖 4.8 為本研究方法在此問題下的收斂狀況。

表 4.11 本研究模式與現況方法之結果

Data Size	Problem Size (orders, machines, stages)	Manufacturer method		Genetic Algorithm	
		Solution	Min. Solution	Average Solution	Max. Solution
Real case problem	( 9, 3, 6 )	5719000	5012300	5091610	5196900

表 4.12 本研究模式與現況方法之結果差距

Data	Problem Size (orders, machines, stages)	Min. Solution Gap (%)	Average Solution Gap (%)	Max. Solution Gap (%)
Real case problem	( 9, 3, 6 )	-12.3571	-10.9703	-9.1292

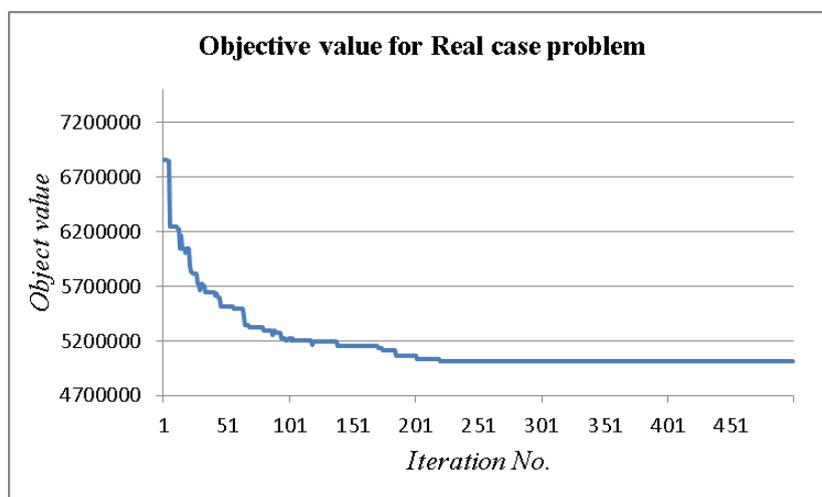


圖 4.8 Real case problem 之收斂圖

表 4.13 為此次實驗下的求解時間，平均求解時間為 91.56 秒，即可求解出比案例公司現行規劃模式較佳的結果。因此，由上述所見，在 Real case 的問題下，本研究所提出之方法，優於案例公司所使用的規劃方式，能得到較佳的生產排程，且求解時間迅速，可明確得知本研究模式的優異之處。

此外，附錄三為此案例下本研究方法的甘特圖，從甘特圖可見，每張訂單幾乎都在 Stage  $n$  完工後，在 Stage  $n+1$  立即開工，表示每張 Order 在每個 Stage 都能緊密相依，不浪費任何時間，已達到最短之 Makespan 的規畫結果。

表 4.13 本研究模式於 Real case problem 下之求解時間

Data Size	Problem Size (orders, machines, stages)	Genetic Algorithm		
		Min. CPU Time (second)	Average CPU Time (second)	Max. CPU Time (second)
Real case problem	( 9, 3, 6 )	88.56	91.56	95.61

## 4.5 小結

本章節透過與數學模型求解工具 CPLEX 的比較，取得各種問題規模下的最佳解或近似解，以求解的品質與時間兩個維度進行比較，結果呈現，不論是在品質或時間，本研究所提的方法之規劃結果，有不錯的績效表現。而在第二個實驗中，使用案例為目前該公司的實際數據，透過與該公司的方法模式比較，也呈現較該公司的方法優異的結果。因此，不論是在最佳化或現行規劃方法的比較上，本研究模式皆呈現不錯的結果。

## 第五章 結論與未來方向

### 5.1 結論

從過去標準的 HFS 中，為單一訂單配置單一機台，本研究則延伸為單一訂單分割配置多機台的問題，提出一多階平行機台之基因演算法，並考量太陽能電池製造業的生產特性，包含多階段製程加工與多平行機台的生產方式，及針對其四大製程特性，平行機台加工、專用機台、獨立整備時間與相依整備時間等影響排程的關鍵因素，解決訂單生產順序與機台規劃之問題。目的為提供規劃人員於安排訂單之生產排程時，以整體的 Makespan 最小化為目標，決定如何將現場的作業機台，適當的給予安排及規劃，得到較佳的生產排程結果。

此外，利用本研究模式與混整數線性規劃求解之方式進行比較，透過實驗結果分別以訂單數量、製程數量與平行機台數量等角度進行觀察，本研究模式不論是在小問題或大問題的情況下，皆有不錯的求解績效，其求解時間更是減少很多。此外，更夠過實際案例的比較，與該公司的現行規劃方式，其結果為本研究模式優於現況方法，因此能提供規劃人員較好的規劃結果，以供參考。

### 5.2 未來方向

本研究所提出的模式，在求解的品質與時間上表現頗佳，但本模式是以一些基本假設為前提下所建立，利用基因演算法解決訂單與機台配置之問題，因此建議未來可改進的研究方向如下：

1. 探討使用不同的交配或突變的方式，使求解的績效能夠有所提升。
2. 基因演算法之參數的設計，本研究採取經驗法則的方式進行設定，未來能夠使用參數最佳化的方法，如田口法，來進行更好的參數探討。

## 參考文獻

- [1] 張晉璋. (2011). 考量專用機台及整備時間之混合流程型批量流問題排程最佳化模式—以太陽能電池產業為例. 碩士, 東海大學, 台中市.
- [2] 莊習武. (2006). 全球能源短缺下看台灣太陽能產業的契機. 碩士論文, 國立清華大學工業工程與工程管理學系碩士班.
- [3] Akrami, B., Karimi, B., & Husseini, S. M. M. (2006). Two metaheuristic methods for the common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: The finite horizon case *Applied Mathematics and Computation*, 183(1), 634-645.
- [4] Alaykýran, K., Engin, O., & Döyen, A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *The International Journal of Advanced Manufacturing Technology*, 35(5-6), 541-550.
- [5] Allahverdi, A., & Mittenthal, J. (1994). Scheduling on M parallel machines subject to random breakdowns to minimize expected mean flow time. *Naval Research Logistics*, 41(5), 677-682.
- [6] Belkadi, K., Gourgand, M., & Benyettou, M. (2006). Parallel genetic algorithms with migration for the hybrid flow shop scheduling problem. *Journal of Applied Mathematics and Decision Sciences*, 1-17.
- [7] Bertel, S., & Billaut, J.-C. (2004). A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159(3), 651-662.
- [8] Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, 64(1-3), 101-111.
- [9] Chen, T.-L., Lin, J. T., & Fang, S.-C. (2010). A shadow-price based heuristic for capacity planning of TFT-LCD manufacturing. *Journal of Industrial and Management Optimization*(1), 209-241.
- [10] Chu, P. C., & Beasley, J. E. (1997). A genetic algorithm for the generalised assignment problem. *Computers and Operations Research*, 24(1), 17-23.
- [11] Engin, O., Ceran, G., & Yilmaz, M. K. (2011). An efficient genetic algorithm next term for previous term hybrid flow shop scheduling with multiprocessor task problems next term. *Applied Soft Computing*, 11(3), 3056-3065.
- [12] Garey, M., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco (Freeman).
- [13] Guangming Lin, & Xin Yao. (1997). Analysing Crossover Operators by Search Step Size. *Proc. IEEE int. conf. Evolutionary Computation*, 107-110.
- [14] Gupta, J. N. D., Hariri, A. M. A., & Potts, C. N. (1997). Scheduling a two-stage hybrid

- flow shop with parallel machines at the first stage. *Annals of Operations Research*, 69(0), 171-191. doi: 10.1023
- [15] Ishibuchi, H., Yamamoto, N., Murata, T., & Tanaka, H. (1994). Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems. *Fuzzy Sets and Systems - Special issue on operations research*, 67(1), 81-100.
- [16] Jenabi, M., Ghomi, S. M. T. F., Torabi, S. A., & Karimi, B. (2007). Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. *Applied Mathematics and Computation*, 186(1), 230-245.
- [17] Jianer Chen, & Chung-Yee Lee. (1999). General Multiprocessor Task Scheduling. *Naval Research Logistics*, 46(1), 57-74.
- [18] Jin ZH, Ohno K, Ito T, & Elmaghraby SE. (2002). Scheduling hybrid flow shops in printed circuit board assembly lines. *Production and Operations Management*, 11(2), 216-230.
- [19] Jina, Z., Yanga, Z., & Itob, T. (2006). Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *International Journal of Production Economics*, 100(2), 322-334.
- [20] Kim, J.-S., Kang, S.-J., & Lee, S.-M. (1997). Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. *Omega*, 25(5), 547-555.
- [21] Low, C., Hs, C.-J., & Su, C.-T. (2008). A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. *Computers and Operations Research*, 35(3), 845-853.
- [22] Luo Pi, Teng Jianfu, Guo Jichang, & Li Qiang. (2001). An improved genetic algorithm and its performance analysis. *Proc. IEEE Int. Conf. Info-tech and Info-net*, 4, 329-333.
- [23] Oğuz, C., & Ercan, M. F. (2005). A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks. *Journal of Scheduling*, 8(4), 323-351.
- [24] Quadt, D., & Kuhn, H. (2007). Batch scheduling of jobs with identical process times on flexible flow lines. *International Journal of Production Economics*, 105(2), 385-401.
- [25] Ribas, I., Leisten, R., & Framin, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37, 1439–1454.
- [26] Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem *European Journal of Operational Research*, 205(1), 1-18.
- [27] Salvador MS. (1973). A solution to a special class of flow shop scheduling problems. In: *Elmaghraby SE, editor. Symposium on the theory of scheduling and its applications.*, Berlin: Springer, 83-91.
- [28] Shiau, D.-F., Cheng, S.-C., & Huang, Y.-M. (2008). Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm. *Expert Systems with Applications*, 37(2), 1133-1143.

- [29] Shimizu, S., Kondo, M., & Matsuda, A. (2005). A highly stabilized hydrogenated amorphous silicon film having very low hydrogen concentration and an improved Si bond network. *Journal of Applied Physics*, 97(3), 033522 - 033522-033524.
- [30] Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms Morgan Kaufmann Publishers, San Mateo, CA*, 2-9.
- [31] Tseng, C.-T., Liao, C.-J., & Liao, T.-X. (2008). A note on two-stage hybrid flowshop scheduling with missing operations. *Computers and Industrial Engineering*, 54(3), 695-704.
- [32] Vignaux, G. A., & Michalewicz, Z. (1991). A genetic algorithm for the linear transportation problem. *IEEE Transactions on System, Man, and Cybernetics*, 21(3), 445-452.
- [33] Wang, H., Jacob, V., & Rolland, E. (2003). Design of efficient hybrid neural networks for flexible flow shop scheduling. *Expert Systems*, 20(4), 208-231.
- [34] Wardono, B., & Fathi, Y. (2004). A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155(2), 380-401.
- [35] Yang, T., Kuo, Y., & Cho, C. (2007). A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. *European Journal of Operational Research*, 176(3), 1859-1873.

## 附錄一：Makespan 之實驗數據表

實驗	A	B	C	M1	M2
1	1	1	1	44400	44400
2	1	1	2	58800	56400
3	1	2	1	31560	31320
4	1	2	2	31560	31560
5	2	1	1	622200	601800
6	2	1	2	668100	688800
7	2	2	1	345614	353893
8	2	2	2	384389	403902

註：A：Orders (A1：3，A2：50)

B：Machines (B1：3，B2：15)

C：Stages (C1：4，C2：8)

M1：第一次測定值

M2：第二次測定值

## 附錄二：CPU Time 之實驗數據表

實驗	A	B	C	T1	T2
1	1	1	1	40.695	41.914
2	1	1	2	64.199	69.461
3	1	2	1	172.747	171.91
4	1	2	2	255.925	236.878
5	2	1	1	455.066	360.785
6	2	1	2	659.383	747.51
7	2	2	1	1711.36	1813.22
8	2	2	2	2857.87	3145.29

註：A：Orders (A1：3，A2：50)

B：Machines (B1：3，B2：15)

C：Stages (C1：4，C2：8)

T1：第一次測定值

T2：第二次測定值

### 附錄三：實際案例之甘特圖

