

國立交通大學
工業工程與管理學系

碩士論文

以組合派工法求解雙流線型工廠排程

A Combined Dispatching Criteria Approach to
Scheduling Dual Flow Shops



研 究 生：陳文旻
指導教授：巫木誠 博士

中華民國九十八年六月

以組合派工法求解雙流線型工廠排程

A Combined Dispatching Criteria Approach to Scheduling Dual Flow Shops

研究生：陳文旻

Student：Wen-Min Chen

指導教授：巫木誠 博士

Advisor：Dr. Muh-Cherng Wu

國立交通大學

工業工程與管理學系

碩士論文

A Thesis

Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Industrial Engineering

June 2009

Hsin-Chu, Taiwan, Republic of China

中華民國九十八年六月

以組合派工法求解雙流線型工廠排程

研究生：陳文旻

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

本論文主要研究一個允許跨廠製造之雙流線型工廠(Dual Flow Shop)排程問題。排程目標為最小化寬裕度的變異係數。所謂的寬裕度，指的是交期與總加工時間之差距。此排程問題包含兩項排程決策：加工途程(Route Assignment, 指派工件到各站)與工件排序(Job Sequencing)。本研究發展一套可同時求解加工途程與加工順序之基因演算法(GA)，而在加工順序方面更使用了多種派工法，讓基因演算法更具多樣性。這些派工法包括單一啟發式演算法、組合派工法以及演化式之派工法則。針對這幾種演算法，本研究皆透過實驗，進行數據間的比較。

實驗結果指出在各種情境下，上述三種方法中，並沒有一種派工法能顯著的贏過其他績效。但是，在大部分的情境下，組合派工法皆能提供相對穩健與較佳之績效結果。

關鍵詞：排程、跨廠、雙流線型生產、組合派工法、基因演算法、交期

A Combined Dispatching Criteria Approach to Scheduling Dual Flow Shops

Student : Wen-Min Chen

Advisor : Dr. Muh-Cherng

Wu

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

This research investigates a dual flow shop scheduling problem, which is in the context of allowing cross-shop processing. The scheduling objective is to minimize the coefficient of variation of slack time, in which the slack time of a job denotes the difference between its due date and total processing time. This scheduling problem involves two decisions: route assignment (assigning jobs to shops) and job sequencing. We developed a genetic algorithm (GA) for making the route assignment decision, which is further varied by including various kinds of dispatching algorithms. These dispatching algorithms include single heuristic rules, combined dispatching rules, and evolutionary approach. Numeric experiments for comparing these scheduling algorithms are carried out. Experiment results indicate that none of the GAs could outperform the others in all scenarios. Yet, the GA equipped with the combined dispatching rules is relatively more robust, performing quite well in most scenarios.

Keywords : scheduling; cross-plant; dual flow shop; combined dispatching criteria; genetic algorithm (GA); due date

誌 謝

本篇論文得以順利完成，首先我必須感謝我的指導教授巫木誠博士細心的指導，在畢業口試前，經過了許多次的波折，有幾次都快讓我覺得無路可走時，老師卻都能為我們找出一條可行之路，真的很感謝老師。此外，在口試期間，也承蒙許錫美教授、彭德保教授、陳文智教授所提出的寶貴意見，讓本論文更趨於完善，僅此致謝。

在碩士班的兩年中也得到了許多人的幫助，像是碩二前的暑假，一起學習程式的 team-昭宏、慈盈、佳玟，以及教導我們程式撰寫的邱志文學長，由於我因為臨時的改題目，比較晚加入學習，可是大家卻都盡力來幫我，讓我能跟上大家的進度，此外也很感謝聰明的 olo-黃亮詮，有問題的時候他都能為我們解答。回想在碩士班中，巫老師齊下的所有人，彼此互相打氣，一路走來有說不完的感謝。最後我還想感謝浩翰，在我心情不好時，能給予我安慰，在我開心的時候能與我一起大笑，更能在我疑惑的時候，給予我最佳的建議，謝謝你。

最後，謹以此論文獻給我最愛的家人，感謝他們能給予我最大的支持，讓我能盡自己最大的努力來完成這份論文，謝謝大家。

陳文旻 于 新竹交大

2009'7'1

目 錄

中文摘要	i
Abstract	ii
誌 謝	iii
圖目錄	viii
第一章 緒論	1
1.1 研究動機	1
1.2 研究議題	1
1.3 研究方法	3
1.4 論文組織	5
第二章 文獻探討	6
2.1 流線型生產排程(Flow Shop scheduling)	6
2.1.1 Flexible Flow Shop 與 Dual Flow Shop 之比較	8
2.2 多廠區相關文獻	10
2.2.1 多廠區生產規劃	10
2.2.2 水平式多廠區相關文獻	11
2.2.3 不跨廠相關文獻	12
2.2.4 跨廠相關文獻	14
2.3 混合實驗設計	17
2.4 基因演算法	20
2.5 小結	22
第三章 研究方法	23
3.1 研究問題	23
3.2 研究假設	24
3.3 問題複雜度	25

3.4 雙流線型工廠模式建構	26
3.4.1 染色體設計	27
3.4.2 染色體解讀	28
3.4.3 加工途程	29
3.4.4 加工順序	34
3.4.5 適合度函數	43
3.4.6 基因演算法求解流程	49
3.5 小結	51
第四章 實驗分析與比較	53
4.1 實驗假設	53
4.2 實驗目的與情境	53
4.2.1 實驗目的	53
4.2.2 實驗標竿	54
4.2.3 實驗情境	54
4.3 實驗分析與比較	61
4.3.1 情境一	61
4.3.2 情境二	63
4.3.3 情境三	65
4.3.4 情境四	66
4.3.5 小結	68
4.4 加工途程分析	69
第五章 結論與未來建議	73
5.1 結論	73
5.2 未來建議	74



表目錄

表 1.1 跨廠範例	3
表 2.1 Flexible Flow Shop 與 Dual Flow Shop 比較表	9
表 2.2 水平式多廠區文獻	12
表 2.3 跨廠文獻分析	15
表 3.1 Stage 1 加工時間與累積加工時間表	32
表 3.2 切割點於 J1、J4 中間	32
表 3.3 切割點於 J3、J2 中間	32
表 3.4 Stage 2 加工時間與累積加工時間表	33
表 3.5 Stage 2 最佳切割點	33
表 3.6 分廠結果	33
表 3.7 8 個工件分廠資訊	38
表 3.8 8 個工件加工時間與交期	38
表 3.9 平均數與標準差的計算	39
表 3.10 標準化後的數值	39
表 3.11 實驗點	41
表 3.12 加工優序計算	42
表 3.13 Stage 1 首次派工情形	45
表 3.14 Stage 1 後續派工情形	45
表 3.15 Stage 2 首次派工情形	46
表 3.16 Stage 2 後續派工情形	48
表 3.17 計算 CV	48
表 4.1 情境一工件加工時間與交期設計方法	55
表 4.2 情境二工件加工時間與交期設計方法	56
表 4.3 情境三工件加工時間與交期設計方法	58

表 4.4 情境四工件加工時間與交期設計方法	60
表 4.5 情境一之派工法績效值(CV).....	62
表 4.6 情境一之派工法改善率	62
表 4.7 情境一之最佳組合權重 (a) 效率協調 (b) 效率不協調.....	62
表 4.8 情境二之派工法績效值(CV).....	63
表 4.9 情境二之派工法改善率	64
表 4.10 情境二之最佳組合權重 (a) 效率協調 (b) 效率不協調.....	64
表 4.11 情境三之派工法績效值(CV).....	65
表 4.12 情境三之派工法改善率	65
表 4.13 情境三之最佳組合權重 (a) 效率協調 (b) 效率不協調.....	66
表 4.14 情境四之派工法績效值(CV).....	67
表 4.15 情境四之派工法改善率	67
表 4.16 情境四之最佳組合權重 (a) 效率協調 (b) 效率不協調.....	67
表 4.17 GA-Comb 在加工效率協調之情境.....	69
表 4.18 GA-Comb 在加工效率不協調之情境	70
表 4.19 GA-GA 在加工效率協調之情境.....	70
表 4.20 GA-GA 在加工效率不協調之情境.....	70

圖目錄

圖 1.1 雙流線型工廠生產流程示意圖	2
圖 2.1 Hybrid Flow Shop Environment.....	7
圖 2.2 Dual Flow Shop 跨廠問題.....	8
圖 2.3 Flexible Flow Shop 跨廠問題	8
圖 2.4 實驗設計求解空間示意圖 (a)二因子(b)三因子.....	17
圖 2.5 混合實驗求解空間示意圖 (a)二因子(b)三因子.....	17
圖 2.6 單體實驗 (a)單體晶格設計(b)單體質心設計.....	18
圖 2.7 擴充單體晶格設計	18
圖 2.8 基因演算法的求解流程	20
圖 3.1 雙流線型(Dual Flow Shop)工廠排程	23
圖 3.2 染色體設計方式	27
圖 3.3 染色體的兩種解讀	28
圖 3.4 Stage 1 分廠示意圖	30
圖 3.5 8 個 Job 的分廠示意圖	31
圖 3.6 染色體三站加工途程示意圖	34
圖 3.7 EDD 派工法說明案例.....	36
圖 3.8 SPT 派工法說明案例.....	36
圖 3.9 LSF 派工法說明案例.....	37
圖 3.10 Augmented Simplex-Lattice Design.....	41
圖 3.11 染色體解讀示意圖	42
圖 3.12 線性順序交配(LOX).....	50
圖 3.13 SWAP 示意圖	50
圖 3.14 基因演算法概略流程	51
圖 3.15 本研究排程適用範圍	52
圖 4.1 GA-GA 分廠示意圖	71

第一章 緒論

1.1 研究動機

半導體業的蓬勃發展，生產技術快速進步的同時，業界面臨的是源源不絕的外部需求，而當市場需求旺盛時，更會有需求大於供給的情況產生，此時，企業為了避免訂單的流失，促使各公司紛紛透過水平上的擴廠、增加生產線等來擴充產能，以承接更大量的訂單，也因此形成了多廠區的生產型態。

過去多廠區型態的生產排程模式，主要建立在單一廠區的生產方式上。多廠區中的單廠個別生產規劃可分為兩階段，第一階段首先進行訂單指派的規劃，大多採全面性的訂單管理，分配好各廠訂單種類後，再進行第二階段各廠獨自的產能規劃。此種生產方式，無法同時考量多廠間的生產績效、產能利用等，因此當市場需求不確定下，造成臨時訂單變動時，已排定的排程無法給予充裕的緩衝時間來因應訂單的調動，而工廠更無法立即透過擴廠、增加生產線等，需要大量時間與金錢的方式來解決因訂單變動所產生的產能不足問題。因此若繼續依照現有的排程方式進行單廠各別加工，可能造成訂單的遲交，間接的影響到企業的聲譽。所以當多廠間的單廠排程無法滿足龐大而又多變的市場需求時，為了有效地利用生產設備、增加產能，又不增加額外的投資下，跨廠支援因此而產生。

所謂的跨廠支援，即是生產方式與機台性質相近的工廠，當一廠產能不堪負荷，而另一廠尚有剩餘產能時，彼此可以互相進行廠與廠之間的支援行為，即是將無法負荷的工件轉由另一廠來進行生產，藉此不僅可滿足兩廠客戶需求，更可以充分的利用閒置產能，換言之，就是為企業帶來更多可用的產能。所以本研究主要針對此類具有跨廠特性的流線型生產排程問題進行規劃，為此型態的問題提出最佳排程建議。

1.2 研究議題

如上節所述，客戶臨時的訂單需求變動，將會造成廠內已排定之排程的混亂，使得其他工件也都接連著受到影響。所以如何能在短期提供臨時所需的產能，使得跨廠排程

的研究逐漸變的重要。本研究針對此類排程問題，以雙廠流線型工廠為對象，提出一套生產排程的方法，希望能為不同情境下的工廠，提出最有效的排程建議。

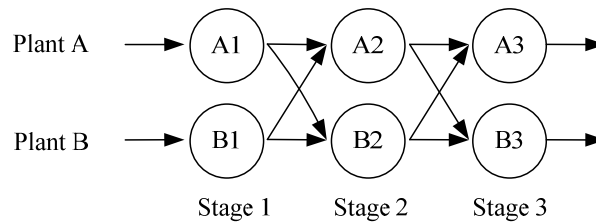


圖 1.1 雙流線型工廠生產流程示意圖

本論文所研究的排程問題為雙流線型工廠排程(圖 1.1)，所謂的雙流線型工廠，指的是兩個生產功能相近，且為流線型(Flow Shop)生產方式的廠，能夠透過彼此工件間的跨廠加工，達到產能的互相協調。

由於實際工廠生產線極為複雜，本論文為了便於進行後續排程規劃的探討，故將其簡化為兩廠(Plant A、Plant B)、三站的形式。針對 Plant A 與 Plant B 兩廠，決定出允許工件跨廠的兩個跨廠點後，接著便可依照這兩點，將整條生產線劃分為前、中、後三段製程，由圖 1.1 來看，A1、B1 代表著前段製程，A2、B2 代表中段製程，而 A3、B3 代表著後段製程。每段製程間的跨廠點，代表工件流經此處時，可以選擇是否進行跨廠加工，也就是工件在此處是可被允許跨廠的位置。

針對兩廠或兩廠以上的生產規劃，將會面臨到兩項決策，分別為加工途程(Route Assignment)與加工順序(Job Sequencing)的決策。所謂的加工途程，指的是顧客下單後，必須決定彼此的工件配置問題，哪些工件是要分給 A 廠，哪些要分給 B 廠；而所謂的加工順序，即是根據各廠所分配到的工件，進行廠內加工順序的配置。

在過去文獻中，多廠區生產規劃的文獻大多僅針對加工途程進行分析與研究，當工件分配至各廠後，便以先進先出的方式進行工件的加工，對於加工順序之決策，並未多做分析。部份的文獻，會針對這兩項決策進行兩階段性的排程，第一階段先透過計算各廠產能負荷，決定出工件的加工途程；第二階段再針對第一階段配置好的工件，運用單

一派工法進行單廠各自的生產排程規劃。但是，兩階段性的排程並無法同時考量同一個績效指標，所以最終的績效結果或許並非能達到最佳。針對這個問題，劉謹銘於 2009 年提出一套能夠同時考量多廠區加工途程與加工順序兩項決策的演算法，透過平衡各廠產能與最早交期最先加工(Earliest Due Date；EDD)法則，運用基因演算法進行演化，以求解雙流線型工廠的生產排程問題。

在過去多廠區生產排程文獻中，可以發現都是以單一派工法則來決定加工順序，但是，Dabbas et al.(2003)與 Lin et al.(2005)在其研究中，都曾驗證過單一派工法則在某些情境或是績效指標下，改善率是有限的。所以劉氏的雙流線型工廠排程中，並非皆能以單一派工法來達到最佳績效。故本論文將針對此問題提出一套組合派工法，融合多種單一派工法的優點，為各種生產情境給予最佳的派工建議。

1.3 研究方法

本論文的最主要是為了解決一個雙流線型工廠排程問題，在具有相似生產線的兩個廠，且彼此有跨廠的合作關係時，可以透過本研究決定出兩廠間的工作配置與加工順序。

論文依照多廠區的兩項決策分兩部份做介紹，首先是決定工件的加工途程，此部份在本研究是依據兩廠產能負荷程度進行分廠，決定工件在各廠、各站的加工位置，換言之，就是決定兩廠的跨廠行為。舉例來說，如表 1.1 所示，J1~J8 代表的是工件 1 到工件 8，它們放置的位置，代表在兩廠的加工位置，所以由此表可以發現，工件 2 在第一站位於 B 廠加工，第二站換到 A 廠，第三站又回到 B 廠加工，也就是說工件 2 在 A、B 兩廠中具有跨廠行為產生。

表 1.1 跨廠範例

	第一站	第二站	第三站
A 廠	J1,J3,J5,J4	J1,J2,J3,J5,J4	J1,J3,J5,J4
B 廠	J2,J6,J7,J8	J6,J7,J8	J2,J6,J7,J8

第二部份要接著決定工件的加工順序，本研究採用組合派工法來進行求解，其主要作法是先使用混合實驗(Mixture Designs)和反應曲面(Response Surface Method)最佳化來求得不同派工法權重的想法，接著再將多個派工法依據個別权重進行線性組合。組合派工法的貢獻就是他會針對不同的情境給予派工方式的建議。假設某個情境下，若運用單一法則 EDD 進行派工會讓績效最佳，則透過組合派工法所求得的权重值便會使 EDD 之权重最高，反之，若某情境是適合使用最短加工時間最先加工(SPT)法則，則會使 SPT 所占的权重值最高。

所以，本研究的主要貢獻便是針對不同的生產情境，能夠同時給予最適當的跨廠與派工建議，讓最終績效指標最佳化。

在績效指標的決定方面，本研究採用變異係數(CV; $CV = \text{標準差} / \text{平均數}$)作為最終的目標式。會採用變異係數作為目標式的原因，是由於本研究的排程屬於工廠生產前的預排排程，當企業接獲訂單之後，便會開始預排每個工件的開始加工時間。為了能讓工件臨時變動時，能有足夠的緩衝時間來因應，不必因為產能不足而重排程，所以在進行工件預排時，會希望達到以下兩個目的：

1. 每個工件都具有足夠緩衝時間。
2. 滿足客戶訂單交期。

為了避免臨時訂單變動造成原排程的變更與額外的損失，規劃結果希望能使變動的工件，具有足夠的緩衝時間，也就是完工時間與交期間的寬裕度能最大化，使得訂單被要求提前出貨時，也能保有足夠的時間可以運用，達到第一項目的。但是實際情況下，我們並無法得知哪個工件將會被提前出貨，因此本研究為了兼顧到所有的可能性，希望能讓所有工件的寬裕度差異最小，所以我們可以透過最小化寬裕度的標準差來達到這個目的。

但在這樣的指標下產生一個限制，就是所有工件必須都要在交期內完工，因為標準差小只能讓每個值都很接近，並無法保證不延誤交期。而實際工廠中，也不能完全保證每個工件都能按時達交，為了能夠擁有足夠多的寬裕時間，所以提出另一項指標—寬裕度平均最大化，藉此讓工件能滿足客戶訂單交期，以達到本研究第二項目的。

因此，為了同時達到最小化標準差與最大化平均數兩項指標，本研究使用變異係數作為最終的績效指標。後續將針對各種情境，只要透過本研究的雙流線型工廠排程模型，便可以決定是否進行跨廠，以及應該使用的派工法則為何。

1.4 論文組織

本論文內容共含五章。第一章敘述研究動機、研究問題與研究方法介紹。第二章為相關文獻介紹，首先探討與雙流線型工廠有關之生產型態，並進行多廠文獻之比較、混合實驗與求解工具基因演算法之介紹。第三章針對本研究的兩項決策，進行問題分析與定義，說明如何決定工件加工途程，以及組合派工法方法介紹。第四章為實例驗證。第五章提出未來可延伸之研究方向。



第二章 文獻探討

本章將介紹與本研究相關之文獻，首先針對流線型(Flow Shop)生產方式做分析，接著介紹與本論文較為接近的多廠文獻與組合派工相關文獻，最後再簡略的介紹一下本研究所使用之基因演算法相關內容。

2.1 流線型生產排程(Flow Shop scheduling)

本論文研究的是雙流線型(Dual Flow Shop)生產排程，Flow Shop的問題在過去幾十年以前就已經被廣泛的研究，依照Allahverdi et al. (1999)的說法，Flow Shop首先被Johnson (1954)所研究，他針對兩部機台，目標式為最小化Makespan來進行求解，後續就越來越多人針對Flow Shop等議題進行不同的研究。

在2008年Allahverdi et al.對於Flow Shop的解釋為，假設總共有 m 個站，每個站擁有一部或多部機台，則工件皆必須依照相同的加工順序於 m 個站中進行生產，即稱為Flow Shop生產方式。並將過去的Flow Shop文獻分為四種類型，分別是Flow Shop、No-Wait Flow Shop、Flexible (Hybrid) Flow Shop、Assembly Flow Shop。

在這四種類型中，本研究的Dual Flow Shop似乎與其中的Flexible (Hybrid) Flow Shop極為相似，所以為了驗證兩者的異同處，在後續將為兩者作一個簡單的分析與比較。

1. Flow Shop

Flow Shop指的是一個工作站中僅含有一部機台，且所有的工件會依照相同的順序在各站進行加工。在過去，這種問題在所有的流程是生產排程中最常被拿來做研究。

2. No-Wait Flow Shop

No-Wait Flow Shop指的是當前一個作業完成之後，下一個作業會立刻開始，中間沒有等候時間，此排程問題最早由 Adiri 和 Pohoryles 在 1982 年提出(Adiri and Pohoryles, 1982)，在這篇文獻中探討 m 台機台在無閒置時間限制下的流程型工廠問題，並且提出在多項式裡，能使平均工作完成時間最小化的目標限制。

3. Flexible (hybrid) Flow Shop

Flexible (Hybrid) Flow Shop 意指至少有一站的機台超過一部。假設生產線共有 u 站，至少一站擁有 m 個平行機台，所有工件會依照相同的順序進行加工，每個工件最多僅能被一台機台所製造，每個機台最多也只能加工一個工件。在流線型的複雜度方面，Linn and Zhang (1999) 更進一步將製造複雜性分成三類：(1) 兩站 HFS、(2) 三站 HFS、(3) k 站 ($k > 3$) HFS。這三者中唯一的差異就是在解決每站兩機台的流程問題時，求解時間非常大 (Johnson, 1954)；而機台增加到三台時，此問題就成了 NP-hard 的問題 (Garey and Johnson, 1979)。

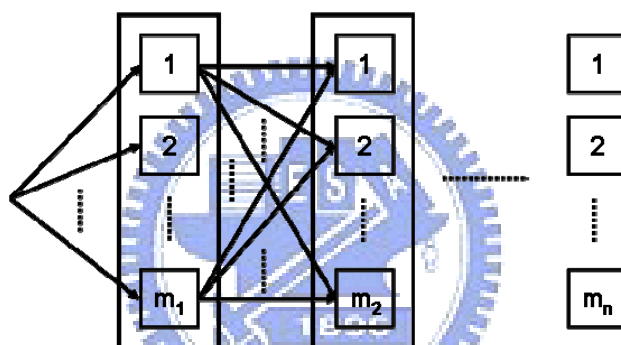


圖 2.1 Hybrid Flow Shop Environment

4. Assembly Flow Shop

Assembly Flow Shop 意指每個工件若必須進行 $m-1$ 次特定的加工，第一站中每個工件會被指派到事先決定好的機台進行加工，接著在第二站進行裝配。舉例來說，假設在一個只有兩站 n 個工件的 Assembly Flow Shop 排程問題中，每個工件必須進行 $k+1$ 次的作業，此時也將有 $k+1$ 個不同的機台來進行這些不同的作業，每個機台一次只能加工一個工件，針對每個工件，前 k 個作業會在第一站的平行機台中進行處理，而最後一項作業會再第二站進行加工；第一站中的前 k 項作業會被指派到不同的機台進行加工，而在第二站的最後一項作業將在第一站所有 k 個作業都完工之後才會開始進行，這兩站的裝配排程問題在業界已經有相當多的應用。

2.1.1 Flexible Flow Shop 與 Dual Flow Shop 之比較

Dual Flow Shop 所探討跨廠問題，過去對於此問題的研究並不多，而本論文所研究的型態如下：

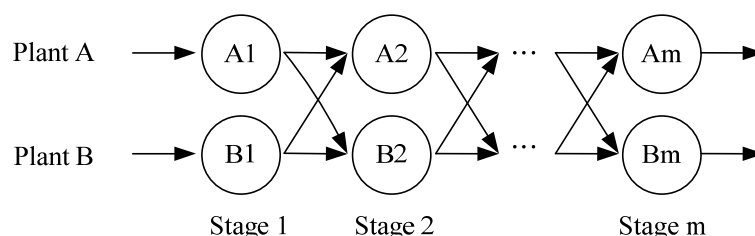


圖 2.2 Dual Flow Shop 跨廠問題

如圖 2.2，兩廠皆為流線型生產型態，兩廠中各站機台功能類似，所以當工件進來後，若 A 廠無法負荷那麼多的工件需求，而 B 廠機台產能允許的話，皆能在各站進行跨廠支援。但是粗略來看，此型態其實與各站皆為兩平行機台的 Flexible Flow Shop 生產型態相當類似，如圖 2.3。

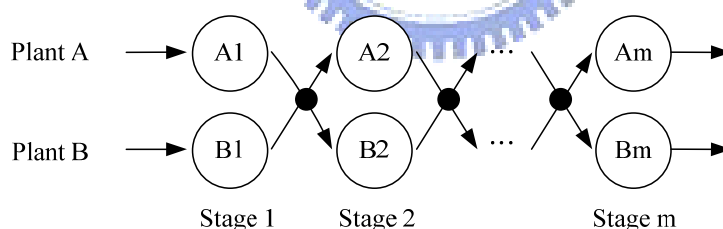


圖 2.3 Flexible Flow Shop 跨廠問題

Flexible Flow Shop 生產型態中，每站的兩個機台其功能也相類似，工件進來後，假若兩機台皆空閒時，便可以任選一機台進行加工。若粗略來看，兩者生產型態是相當類似的，此時便會令人產生疑惑，如果可以將 Dual Flow Shop 的問題看作 Flexible Flow Shop 的問題，那本論文是否沒有研究的意義，為了分清楚兩者間的差異，以下將針對這兩種生產型態，做更詳細的比較：

表 2.1 Flexible Flow Shop 與 Dual Flow Shop 比較表

	Flexible Flow Shop	Dual Flow Shop
Route 決策問題	沒有 Route 決策問題 ，有空即加工	有 Route 決策問題 ，產能與跨廠運輸間的權衡，跨廠須考量運輸時間
運輸時間	無 or 很小(搬運時間)	有(運輸時間)
運輸(搬運)時間計算點	兩站之間，不管是否換機台皆有且相同	跨廠才有運輸時間
生產型態	不能單獨切作兩條 Flow Shop	可視為兩條單獨 Flow Shop

表 2.1 將兩者之間的差異做個比較，Flexible Flow Shop 與 Dual Flow Shop 第一項不同為 Route 的決策問題，指的是工件由本站換到下一站加工時，是否必須事先決定工件的流向，也就是決定工件的加工途程位置。由 Flexible Flow Shop 的模型中可以發現，當工件在第一站的任何機台加工後，到了第二站進行加工時，選擇機台的依據不會受第一站加工位置的影響，例如第一站在圖 2.3 上方的機台加工，而第二站則是看哪個機台有空就在哪個機台加工；但 Dual Flow Shop 就不同了，當第一站在圖 2.2 中的 A 廠進行加工後，要到第二站時，主要是先在同一廠做，當產能不足而另一廠有多餘產能時才會進行跨廠加工，所以本研究的跨廠排程會考量前一廠的加工位置，並不是哪個機台有空就能在那加工。

第二項與第三項與運輸(搬運)時間有關，過去的研究中 Flexible Flow Shop 兩站之間大都沒有所謂的運輸時間(或稱搬運時間)，只有部分文獻有討論到兩站之間的搬運時間，而且是廠內間站與站的搬運，只要工件從本站到下一站都需要設定搬運時間，若假設搬運時間相同時，則在進行跨機台的決策時，便不需要將搬運時間考量進去；至於 Dual Flow Shop，本論文設定跨廠皆須運輸時間，且運輸時間是比照真實世界來設定，假設兩個可以互相進行支援的廠，一座在桃園，另一座在新竹，運輸時間相較於 Flexible Flow Shop 的搬運時間是多上許多。雖然 Dual Flow Shop 單廠站與站間的工作轉移雖然也是會有搬運時間存在，但是相較於廠際間的運輸時間，搬運時間是相當小的，所以

Dual Flow Shop 才會延伸出跨廠的決策問題。由此來看，Flexible Flow Shop 與 Dual Flow Shop 間的運輸(搬運)時間除了差在時間長短以外，Flexible Flow Shop 主要是站與站之間就必須搬運時間，而 Dual Flow Shop 原則上是跨廠工件才會有運輸時間，所以兩者是相當不同的。

第四項不同在於生產型態的不同，Flexible Flow Shop 不能單獨拆作兩條 Flow Shop 來看，因為工件選擇在站中哪個機台加工是依照誰先有空先加工；Dual Flow Shop 若不進行跨廠，其實是可以視為兩條單獨的 Flow Shop。

以上這四點是 Flexible Flow Shop 與 Dual Flow Shop 間基本的不同點，除了上述原因以外，本研究之所以會想針對 Dual Flow Shop 進行延伸，更為實際的原因是為了能夠透過跨廠加工，藉此以最少金錢、最少時間來提供兩廠最大的產能。所以其實兩者的差異是很大的，因此本論文將針對較少人所研究的跨廠支援進行探討，並建構一套生產排程模式，為企業提供最佳的排程建議。



2.2 多廠區相關文獻

目前針對多廠區的文獻很多，為了繼續做深入的研究，所以首先必須先瞭解何謂多廠區，而它又分做哪些種類，本節將針對多廠區作一個詳細的介紹。

2.2.1 多廠區生產規劃

一般而言，所謂的多廠區，在目前看來有兩種不同的多廠型態。有些學者將其定位在供應鏈中，廠間如供應鏈般的前後關係，透過上下游多個廠進行整合與排程關係，生產功能則有上下游不同的差異，其為垂直式多廠概念(Frederix, 1996；Timpe et al., 2000)；而部份學者則將其解釋為一個公司同時擁有多個相類似的生產工廠，但廠間有地理位置或組織差異，其為橫向多廠概念(Chen et al., 1999；Sauer et al., 1998)。本文所建構的多廠區型態屬於後者水平式多廠區型態。

針對水平式的多廠區相關文獻定義有許多，但大體上的觀念是相同的。Caroloin (1995)將多廠區定義為在同一企業下的兩個廠對於某產品之某段製程具有相同或互補的製程能力，產品的製造流程可以使用不同廠區內的資源。而在 Thierry et al.(1995)中，也將多廠區定義為當一個公司的生產設施分佈於不同的地理位置，即稱其具有多廠特性。Vercellis (1999)也提到，將訂單需求分配至不同區域生產，以每個具有相同生產特性之工廠共同完成訂單需求。也就是將散佈在各地功能類似的工廠視為多廠規劃的主體，是屬於橫向的多廠關係。

一般來說，具多廠生產之製造廠區，工廠大多坐落在不同地區，生產規劃的方式亦由單廠擴充至多廠的型態相對衍生出複雜的生產規劃，在 Sauer et al.(1998)研究中也指出，將產品分配至不同廠進行加工時，會產生複雜的交互依賴關係，因此在考量多廠區生產型態時，與單廠區相比，其實是複雜許多的。

2.2.2 水平式多廠區相關文獻

水平式多廠區可以分做兩種，分別為跨廠與不跨廠。所謂的跨廠，是廠與廠之間在一開始決定好要加工的工件途程後，後續工件在各自廠內加工時若遇到產能不足或其他限制，而另一廠能夠支援時，便可進行工件跨廠的動作；而所謂的不跨廠，即是廠與廠之間在一開始決定好要加工的工件途程後，便各自進行單廠工件排序，其中不管是否有產能閒置或產能不足等其他限制，都不互相進行工件跨廠的動作，有些文獻甚至只作工件途程上的分配。依照上面的論述，本研究又可以將跨廠與不跨廠分作兩步驟，工件途程(Route Assignment)、工件排序(Job Sequencing)，表 2.2 將水平式跨廠與不跨廠相關文獻進行分類：

表 2.2 水平式多廠區文獻

	不跨廠	跨廠
Route Assignment	Thierry et al. (1995) 陳建銘 (2003) Sambasivan et al. (2005)	張美滿 (2002) Toba (2005) Wu and Chang (2007) Wu et al.(2008)
Route Assignment + Job Sequencing	Sauer et al. (1998) Vercellis (1999) Sauer et al. (2000) 郭乃蓁 (2000) Guinet (2001) Chen et al. (2005)	劉謹銘 (2009) 本研究

表 2.2 將多廠區文獻依照跨廠與不跨廠分作左右兩區塊文獻，依照只作加工途程 (Route Assignment) 決策與加工途程 (Route Assignment)、加工順序 (Job Assignment) 皆有考量，將文獻再分為上下兩個區塊。也就是說，左上角是僅考量加工途程之不跨廠文獻，反之，右下角區塊的文獻則是加工途程與加工順序兩項決策皆有考量的跨廠文獻。以下將在 2.2.3 節針對不跨廠文獻、2.2.4 節針對跨廠文獻做簡單介紹

2.2.3 不跨廠相關文獻

1. 僅討論工件途程之不跨廠文獻：

Thierry et al. (1995) 將多廠定義為”當一個公司的生產設備位於不同的地理位置即為多廠生產”，指出多廠區問題主要決定產品在不同產區間的生產以及不同時段間的運送問題。將限制滿足問題 (Constraint Satisfaction Problem) 領域的滿足限制方法，應用於多廠規劃之線性規劃模式的求解，並同時考慮到有些生產單位在切換生產產品時需要較長設置時間的特性。限制滿足問題的求解法是藉由限制來一步步縮小求解空間，等到無法再縮小時，就利用各種啟發式方法來快速找到好的解。此研究有考慮到當某些生產

單位在切換生產產品時需要較長設置時間的特性，所以各生產單位在規劃時的時間區間大小不同。

Sambasivan et al. (2002) 提出一啟發式法則運用 Lagrangean 來解決多廠多階的廠內運送(inter-plant transfers)問題。

陳建銘(2003) 針對擁有多製造廠區的企業面臨緊急訂單時，迅速判斷利潤最大的緊急訂單承接組合，方法分兩階段，第一階段規劃方法為典型的背包問題(Knapsack Problem)原則，考量企業獲利目標，求得緊急訂單利潤最大承接評估模式，第二階段以各廠區內一般訂單、第一階段所求出的解，以線性規劃模式求出總成本最小化的跨廠區產能規劃。

2. 分階段討論工件途程與工件排序之不跨廠文獻：

Sauer et al. (1998)使用模糊理論的概念來處理傳統上會用固定值代替或是忽略不記的一些不確定因素如產能和運輸時間等。在 Sauer et al. (1998)的研究中，將多廠區生產系統的排程問題分為全域(global)排程、區域(local)排程與運輸排程。全域層次的主要任務是產生每個位置(location)中需要被製造的中間產品(intermediate product)之需求量，在不影響其他廠區的同時，提供足夠的彈性使得區域排程(指的是單一位置)能夠反應區域性的干擾事件，他藉由加入緩衝時間的啟發式法則及模糊技術來最佳化機器群組的平均負荷。

Sauer et al. (2000)則是進一步的將各廠間的運輸排程問題納入多廠規劃的求解中，藉由將運輸排程問題轉換成類似於生產排程的問題，之後和其餘各廠一起規劃求解，並認為多廠規劃的目的在於產生一組穩健的排程，也就是此排程要使各單廠的排程人員有足夠的彈性去處理各自的變異，並且不會影響到其他工廠。

Vercellis (1999) 考量生產規劃中的主生產排程(master production planning, MPS)以及產能配置(capacity allocation)問題求解每一廠區具相同生產流程之多廠間生產問題。將需求量分配至不同區域生產，利用 LP 方式解決具相同生產流程之多廠區生產規劃問題，並考慮設備設置時間與各廠區成本因素，求得最低成本的訂單需求分配量。

Guinet (2001)認為多廠生產規劃的目的在於找出「誰該生產」與「什麼時候產品和工件必須生產」這兩個問題的答案，提出以二階(two-level)式生產管理手法管理多廠生產系統，指出多廠規劃問題可以被建構成受限於變動和固定成本的流動問題 (flow problem)，或者是以提早和延遲總和最小化為目標的非相關平行機台排程問題。而其在考量產能限制和機台設置特性，以及生產、設置和運輸等成本最小化的情形下，提出了一個「主要-對偶線性規劃法」來求解多廠規劃問題。

郭乃蓁 (2000)提出晶圓製造廠多廠間的訂單抵換機制，期使晶圓製造業者在面臨顧客提出的插換單要求時，能依據各廠之生產特性及產能，作最合適的調度，基於平衡負荷的理念，以關鍵資源為監控重點，首先透過「產能推估模組」，依據既定的排程資訊與插入訂單的交期，進行其達交性的粗估。其次，在訂單分派模組中的篩選階段，挑選出最適抵換方案，期能維持系統績效。最後，在抵換推估模組中，依據各廠訂單異動的結果，進行重排程及異動之績效評估。

2.2.4 跨廠相關文獻

因為本論文主要研究跨廠排程，所以本節將針對表 2.2 中跨廠文獻，作更進一步的比較，如表 2.3。



表 2.3 跨廠文獻分析

	張美滿 (2002)	Toba (2005)	Wu and Chang (2007)	Wu et al. (2008)	劉謹銘(2009)
當機情境	不考慮當機	不考慮當機	過去當機資料 推估	平均當機率	不當機
決策	三階段規劃 模式 (Route)	動態、即時 跨廠規劃 (Route)	每週定期「跨廠 產能交易」 (Route)	跨廠區「途 程規劃」 (Route)	跨廠區「排程規 劃」(Route & Sequencing)
派工法則	--	--	--	--	EDD
跨廠方式	瓶頸站產能 負荷程度	動態進行	產能交易	1 切點,切點 未定	2 切點,切點已 定
目標	Total Cost	Cycle time	Throughput	Throughput	Slack(寬裕度)
生產型態	Job shop & Flow Shop	Reentrant Job Shop	Reentrant Job Shop	Reentrant Job Shop	Flow Shop

1. 僅討論工件途程之跨廠文獻：

張美滿(2002) 針對多廠區規劃問題，在廠與廠間歷史銷售量及既有的資源設備下，建立三階段規劃模式，以生產總成本最小為目標。第一階段業務依歷史銷售量統一接單，管理人員以機器產能限制，並考量生產成本、製造成本、延遲成本、外包成本等成本因子之總成本最小下，求出各廠區所需生產的最佳訂單數量。第二階段求出各廠區各工作站的產能負荷量與瓶頸工作站。第三階段針對瓶頸工作站加以平衡，取得工作站跨廠生產量、外包量與產能利用率，之後便依據最佳跨廠產能結果安排生產細部規劃與排程規劃。此論文考量各廠區產能限制與各廠區不同的生產成本；外包成本等，非常符合各企業規劃現況，但並未考量緊急訂單。

Toba (2005) 提出一項負荷平衡法來平衡在多個半導體晶圓廠中所有產品的加工作業。對於半導體多廠區生產規劃分析研究中，以動態、即時作跨廠途程規劃，將作業(operation)分成多個作業集合，每個「作業集合」完工之後，需即時決定下一個「作業集合」的生產工廠，目標是最小化 cycle time。後續藉由模擬實驗來驗證此方法比普通的方法更能改善多個具有獨自製造能力的晶圓廠中的負荷。其能有效的減少在每個廠中每個加工步驟的等候時間及前置時間。

Wu and Chang (2007) 對於晶圓廠之短期產能交易決策模式分析研究中，利用單項/多項做短期產能交易決策，每週按「原訂生產計畫」與「現在機台狀況」，來預估各廠區每一工作站的下週產能利用率。當各廠產能利用率失衡時，低利用率之工作站將販賣產能給他廠高利用率工作站，此研究之決策為決定每週最佳的產能交易組合，以最大化兩廠長期利潤總和為目標。

Wu and Chen (2008) 對於半導體雙廠區產能相互支援的途程規劃分析研究中，提出兩項決策，決定跨廠位置與每個產品的途程比例，運用現性規劃求解跨廠切點，然後再運用基因演算法求解途程比例。此研究透過途程的可切割性作跨廠生產規劃決策，充份利用機台產能，在目標 CT 下最大化整體產出。

2. 同時討論工件途程與工件排序之跨廠文獻：

劉謹銘(2009) 針對跨廠提出一套方法，此研究認為過去並未有人同時討論過工件途程與工件排序這兩項決策，所以運用基因演算法提出新的基因表達法來同時求取工件的此兩項資訊。在加工途程決策方面，以兩廠負荷時間差距最小化作為分廠點；在加工順序決策方面，以最早交期最先加工(EDD)的單一派工法則來進行工件排序。為了因應多變的市場需求，希望排程能具充分的穩定性，選擇最小化變異係數(CV)為其績效指標。

由表 2.2 及表 2.3 中得知，張美滿(2002)、Toba (2005)、Wu and Chang (2007)、Wu and Chen (2008)此四篇文獻都只作了工件途程(Route Assignment)的分配，而劉謹銘(2009)才有同時對工件途程與工件排序作相關的研究，但是，劉氏的文獻中，僅針對單一派工法則來進行工件排序，而 Dabbas et al.(2003)與 Lin et al.(2005)在其研究中，都曾驗證過單一派工法則在某些情境或是績效指標下，改善率是有限的。所以本論文後續將針對此模型，提出一套組合派工法，讓不同情境或績效指標下，都能以此方法求得最適宜的派工法則

2.3 混合實驗設計

混合實驗設計首先由 Scheffe 於 1958 年提出，他認為所謂混合實驗，指的是在一個實驗中，若實驗的品質特性(反應變數)與各成分(因子)間含量的比例有關，則此實驗稱為混合實驗。並定義實驗設計中各因子佔總成份比例必須為 0 到 1 之間，而且所有因子比例加總必須為 1。假設獨立變數 X_i 為混合實驗中之成份，則 X_i 有下列特性：

$$\begin{aligned} 0 \leq X_i \leq 1 \quad i=1 \sim p \\ \sum_{i=1}^p X_i = 1 \end{aligned} \quad (2.1)$$

由於混合實驗中具有 $\sum_{i=1}^p X_i = 1$ 的限制式，導致實驗可行解區域將減少而造成與一般實驗設計有所差異。在過去二因子及三因子實驗設計中，解空間的範圍如圖 2.4 所示，分別為一個二維平面與立方體；但若加上混合實驗的 $\sum_{i=1}^p X_i = 1$ 限制，其解空間便降低一維，如圖 2.5 所示為一條線與一個三角形的範圍。

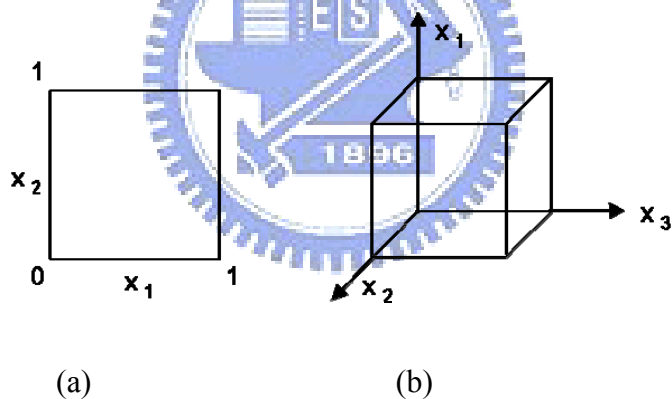


圖 2.4 實驗設計求解空間示意圖 (a)二因子(b)三因子

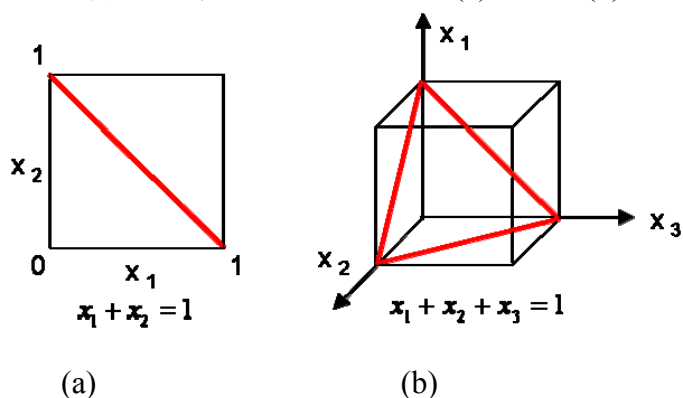


圖 2.5 混合實驗求解空間示意圖 (a)二因子(b)三因子

混合實驗設計一般若無成分比例限制時會使用單體實驗設計。所謂成分比例無限制的單體實驗，指的是組成品質特性(反應變數)的各成分(因子)並沒有上下限的限制，或是成份皆落於某個區域，可以透過轉換將其值轉換成 0 到 1 之間。若該實驗的成份之比例介於 0 至 1 之間時，可使所有成份比例於可行解範圍內呈現規則形狀，在此情況下，便能透過單體實驗來進行規劃。一般的單體實驗方法，包括單體晶格設計(simplex lattice design)與單體質心設計(simplex centroid design)，單體晶格設計是將各實驗點均勻的分佈在實驗可行區域內，而單體質心設計則是在單體晶格設計的實驗點中，在增加一個中心點作為實驗點(圖 2.6)。

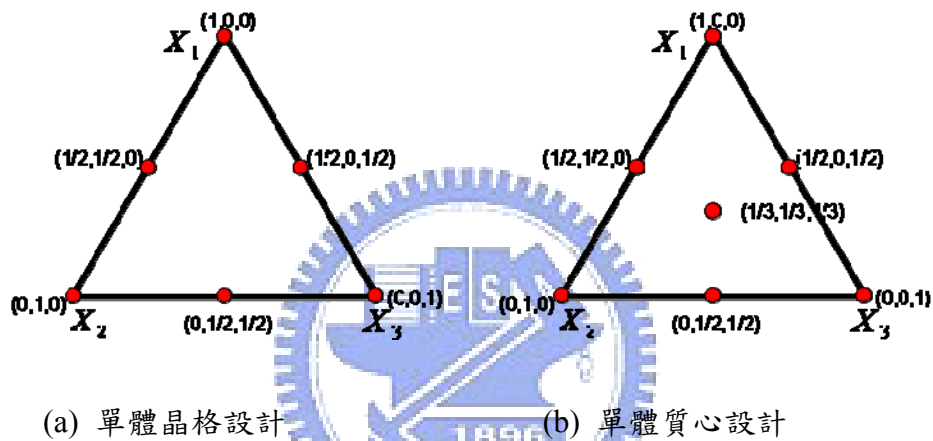


圖 2.6 單體實驗 (a)單體晶格設計(b)單體質心設計

但基於實驗點過少可能造成的實驗誤差，所以由單體晶格延伸出擴充單體晶格設計(Augmented Simplex-Lattice Design)(如圖 2.7)，由單體晶格可行域中，進行端點與中間點的連線來增加實驗點。

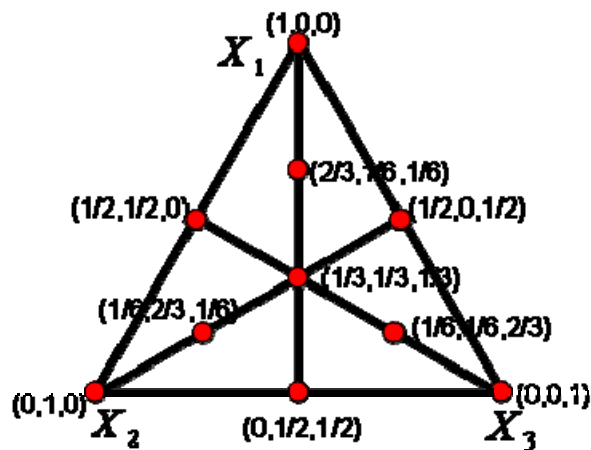


圖 2.7 擴充單體晶格設計

進行實驗設計後，通常會以反應曲面法(Response Surface Method)來尋找最佳配方，最佳化過程如下：

- (1) 依據實驗設計的實驗點進行實驗。
- (2) 由實驗設計的結果建構出反應曲面方程式。
- (2) 檢定此反應曲面方程式的正確性與解釋程度。
- (3) 依據績效導向，由方程式中找出最佳配方。

通過這四個步驟可以確保適配出來的模型符合實際資料，且能由此模型得到最佳配方。其中反應曲面模型之定義如下：

$$\eta = \beta_0 + \sum_{i=1}^q \beta_i x_i + \sum_{i<j}^q \beta_{ij} x_i x_j + \sum_{i<j<k} \sum \beta_{ijk} x_i x_j x_k + \dots \quad (2.2)$$

由於混合模型與反應曲面模型中所使用之多項式是不相同的，要讓混合實驗使用，需要將反應曲面模型之多項式中的截距項皆化簡為0。所以通常會使用以下幾個模型：

線性(Linear) 模型：

$$\eta = \sum_{i=1}^q \beta_i x_i \quad (2.3)$$

二次項(Quadratic) 模型：

$$\eta = \sum_{i=1}^q \beta_i x_i + \sum_{i<j}^q \beta_{ij} x_i x_j \quad (2.4)$$

完全立方(Full Cubic) 模型：

$$\eta = \sum_{i=1}^q \beta_i x_i + \sum_{i<j}^q \beta_{ij} x_i x_j + \sum_{i<j}^q \delta_{ij} x_i x_j (x_i - x_j) + \sum_{i<j<k} \sum \beta_{ijk} x_i x_j x_k \quad (2.5)$$

特殊立方(Special Cubic) 模型：

$$\eta = \sum_{i=1}^q \beta_i x_i + \sum_{i<j}^q \beta_{ij} x_i x_j + \sum_{i<j<k} \sum \beta_{ijk} x_i x_j x_k \quad (2.6)$$

本研究使用三種派工法則(成分)來進行混合實驗，混合實驗的實驗點採用擴充單體晶格設計的十個實驗點，透過反應曲面最佳化來求取每個派工法的最佳權重值，這部份將在第三章研究方法中，做更詳細的介紹。

2.4 基因演算法

基因演算法 (Genetic Algorithm ; GA) 為 Holland 於 1975 年所提出的一種隨機多點平行式全區域搜尋法則。是基於「物競天擇」及「遺傳法則」的概念，所謂的「物競天擇」是指 Gas 會在尋優過程中去蕪存菁，踢除較差的成員，保留較優良的成員。而「物競天擇」的操作方式則是模仿遺傳學複製、交配及突變等三大法則，故稱之。

基因演算法其搜尋過程在一個特定的基因編碼 (Gene encoding) 的規則下，以特定的演算法或隨機產生起始的族群 (Population)，再依據定義的目標函數 (Objective function) 計算個體的適應值 (Fitness)，個體的適應值愈高代表適應能力愈強，其衍生為子代的機率愈大。經過多代繁衍所得的最高適應值之染色體即為最佳解。常見的運算子即為之前提到的複製 (Reproduction)、交配 (Crossover) 及突變 (Mutation) 三種，每個運算子重覆循環執行，如此一代一代進化過程中，保存最好的染色體，直到系統達到預定的演化次數或所設定的誤差容許度內為止。整個基因演算法的解題流程如圖 2.8 所示。

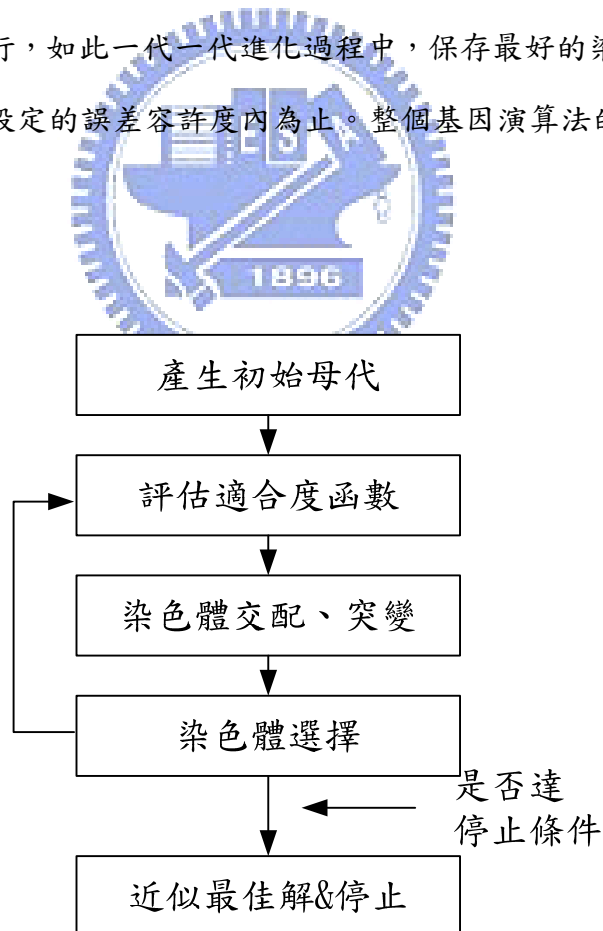


圖 2.8 基因演算法的求解流程

以下為基因演算法各名詞之介紹：

染色體與基因

在執行基因演算法前，必須透過編碼的程序，將決策變數轉成可以代表可行解的字串，此字串稱為染色體，在其中的每個字元稱為基因，一般是以一個基因代表一個決策變數，基因值即代表決策變數值。

族群

在基因演算法的過程中可以同時存在多個染色體，數個染色體形成的群體稱為族群。

適合度函數

評估所有染色體在族群中之表現好壞，表現好的染色體即表示適應力較佳，因此存活機率也較高，反之較差的染色體適應力較差，因此存活機率也較低。

在求解過程中，必須將染色體的表現轉換成適合度函數值，在望大的情境中，會將染色體的目標值當成其適合度函數；在望小的情境中，因為染色體的目標值越小適應函數值越高，通常以染色體目標值的倒數或是取族群中最大目標值減去該染色體目標值當作適合度函數；若為望目的情境，染色體的目標值越接近設定的目標值適合度函數值越高，故先求算染色體目標值與設定值之差距，再用望小問題方法來求其適合度函數。最後選取族群中適合度函數值最高的染色體當做下一代。

選擇策略

針對交配池中的所有染色體，依據其適應函數值來選擇下一代。選擇的方式目前有很多種：俄羅斯輪盤法(Roulette wheel selection)、競賽法(Tournament selection)。

所謂的俄羅斯輪盤法，意思是說每個染色體將被選中來進行繁衍後代之機率，為該染色體之適合度值佔族群每一染色體之適合度值總和的比例。這方法的缺點為適合度函數值較好的染色體容易支配整個族群的染色體的選擇，容易過早陷入局部最佳解。

競賽法是隨機選取兩個或以上的染色體，在望小的問題中，選擇最小適合度函數值的染色體，這個方法可以避免過早陷入局部最佳解。

交配

交配是兩個被隨機選取的染色體彼此交換其內部的基因，藉由這個方法試圖找到較好的染色體，一般排程的交配有單點交配（one-point crossover）、線性順序交配(LOX: linear Order Crossover Operator)、部分相應交配(PMX: Partially matched crossover Operator)、 NABEL Operator 等

突變

突變通常會配合交配一同使用，避免基因演算法陷入局部最佳化 (local optimum)。一般常用的突變有 Swap (交換)、Inverse (反轉)及 Insert (插入)(Wang and Zheng, 2003)。

至於基因演算法在本論文中的各步驟流程以及相關之參數設定（如族群大小、突變率、交配率以及終止條件），將於第三章中詳細描述。

2.5 小結

過去多廠文獻中，大多文獻都是研究不跨廠的居多，而且大部分都只做工件途程的研究，所以本研究主要的特色除了針對極少人探討的跨廠問題進行研究外，還同時討論了工件途程與工件排序問題，後續更運用權重來組合不同派工法，找出各種情境下最適合的派工法則。接下來的章節將針對本研究所使用的求解方法做詳細的介紹。

第三章 研究方法

本章主要在說明如何運用基因演算法與組合派工法，求解雙流線型工廠排程模式，並針對本研究的兩項決策，進行相關的參數設定。首先進行問題的描述與假設，其次介紹如何決定工件的加工途程與加工順序，以及適合度函數，最後介紹基因演算法的所有流程。

3.1 研究問題

如前所述，本論文所研究的問題為雙流線型工廠排程(圖 3.1)，所謂的雙流線型工廠，指的是兩個生產功能相近，且為流線型(Flow Shop)生產方式的廠，能夠透過彼此工件間的跨廠加工，達到產能的互相協調。為了便於進行排程規劃的探討，故將其簡化如下：

- 兩廠，各有三個 Stage。
- 各個 Stage 機台功能相似。
- 以實際工廠生產線可進行跨廠的位置進行切割，切出三部份。
- 兩廠皆為 Flow Shop 生產方式。
- 每個 Stage 都可做互相支援。

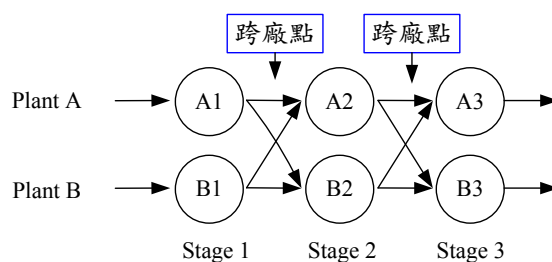


圖 3.1 雙流線型(Dual Flow Shop)工廠排程

雙流線型工廠排程面臨的排程問題有兩點，第一點為客戶下單後，中央必須決定兩廠訂單配置問題；第二點為決定廠際間與各廠工件排序問題。透過本論文整體排程規劃後，希望能藉此解決客戶臨時訂單變動問題。

在績效指標的決定方面，第一章已經說明過，本研究將採用變異係數(CV； $CV = \text{標準差} / \text{平均數}$)作為最終的目標式，藉此達到兩個目的：(1) 每個工件都具有足夠緩衝時間；(2) 滿足客戶訂單交期。為了達到第兩個目的，必須使寬裕度的標準差很小、平均數很大，這樣才能滿足每個工件都具有足夠緩衝時間的目的，而且工件也能不延誤交期。因此，透過最小化寬裕度的變異係數(CV)，來同時達到最小化標準差與最大化平均數兩項指標。

3.2 研究假設

本研究對於跨廠的情境可以分作三部份，分別為 Route、Machine、Job，以下將針對各項進行情境的定義與假設。



1. Route

- 共有 A、B 兩廠
- 廠間可作跨廠支援加工，跨廠運輸時間需考慮且固定
- 每個 Job 都須經三道加工流程(Flow Shop)

2. Machine

- 各廠各 Stage 各有一機台
- 機台不停機不當機
- 無 setup time

3. Job

- 訂單即為工單
- 工廠起始階段沒有 WIP (空廠)
- 每個 Job 有各自的 Due date

- Job 批量不可切割
- 一有跨廠 Job 即進行跨廠(不集批運送)

本論文為雙流線型工廠排程所以一開始在 Route 的假設中，便將模型定義為兩間工廠，而且廠與廠之間彼此可以進行不定期的跨廠支援，為了將跨廠模式加以簡化，本論文設定工廠起始階段沒有 WIP (空廠)，以實際工廠生產線可進行跨廠的位置進行切割，切出三部份，所以最後僅針對前段製程(第一站)、中段製程(第二站)與後段製程(第三站)，分成縮簡的三個站來研究，每站都僅有一部加工機台，因為屬於流線型生產模式，每個工件也都必須經過三道加工流程(Flow Shop)。

針對跨廠部分的設定，因為要與單廠平行機台有所差異，又希望能更貼近事實，所以在進行跨廠時，需考慮跨廠時間，而且不管是在哪站進行跨廠，其跨廠時間都會是一樣的，當一有要跨廠的工件時，即進行跨廠(不集批運送)。

對於機台的設定，由於本研究屬於預排排程，所以就不去考量機台停機、當機問題，而且不考慮設置時間(setup time)；至於工件的設定，每個工件都具有各自的交期日，可以是相同的，也可以是不同，而且工件是不可切割的。

針對以上機台與工件的假設下，將在後續說明如何建構一個同時考量加工途程與加工順序，以解決多廠區複雜的排程問題。

3.3 問題複雜度

如同先前一直提及的，本論文屬於多廠排程問題，所以須進行兩項決策的求解：加工途程(Route Assignment)、加工順序(Job Sequencing)。

要透過整數規劃求解上述兩項決策時，必須同時考量加工途程與加工順序來進行模式的建構，而本研究必須再加上交期的限制，在進行整數規劃時，須考量到時間點的安排，所以複雜度非常高，求解是很困難的。以加工途程來看，其求解空間的算法為 $F^{m \times n}$ ， F 代表工廠數， m 代表站別， n 代表工件數，以本研究來看為兩廠、三站形式，當有 40 個待加工工件時，其加工途程的求解空間便為 $2^{3 \times 40}$ 這麼多的可能。而在實際業界裡，工

件一定不只有 40 個，站別也一定會超出三站，每個工件都按照這樣的方式來計算加工途程的話，結果會是相當驚人的，後續就更不用提還要再加上加工順序的所有的排列結果。

此類求解空間大且複雜，無法用整數貴化模型求解之問題，過去的文獻將其定義為 NP-hard 的問題 (Garey and Johnson, 1979)，若要藉由整數規劃模型來求取最佳解，其解題所花費的時間，可不是一天兩天就可以完成的。然而，對於業界而言，時間就是金錢，若要為了求得最佳解而花費個一兩個月來進行解題，當你求出解時，工件也全都超過原本答應給客戶的交期了。所以為了能以最快的速度進行求解，又希望所求的解不至於太差，大家開始退而求其次，交由一些演算法來解決。過去有許多篇文獻都在針對此類 NP-hard 的問題提出不同的演算法，或是將不同的演算法套用在不同的問題中求解。本研究也不例外，後續的解題將運用 Holland (1975) 所提出的基因演算法，來進行本研究的求解，並運用組合派工法進行加工順序的排列，希望能以最快的速度來求出最佳近似解。



3.4 雙流線型工廠模式建構

本節分為四個部份來介紹雙流線型工廠模式的建構流程：(1)染色體設計；(2)染色體解讀；(3)適合度函數；(4)基因演算法求解流程。

在第一部份將針對基因演算法中的染色體進行介紹，第二部份說明本章的重點—加工途程與加工順序在染色體中如何被解讀；第三部份針對基因演算法中的適合度函數做介紹；最後第四部份說明基因演算法求解流程與相關設定。

以下針對上述三個部份，進行相關符號的設定：

i ：工件代號， $i=1, 2, \dots, n$

s ：加工順序， $s=1, 2, \dots, o$

j ：Stage 代號， $j=1, 2, \dots, m$

f ：工廠別， $f=1, 2$ 其中 $f=1$ ，代表 A 廠； $f=2$ ，代表 B 廠

$P_{i,s,j,f}$: 工件 i 在工廠 f 、Stage j ，加工順序 s 的製造時間

$ST_{i,s,j,f}$: 工件 i 在工廠 f 、Stage j ，加工順序 s 的開始生產時間

$CT_{i,s,j,f}$: 工件 i 在工廠 f 、Stage j ，加工順序 s 的完工時間

TS : 運輸時間

d_i : 工件 i 的 Due Date

σ_s : 寬裕度的標準差

\bar{X}_s : 寬裕度的平均數

CV : 寬裕度的變異係數， $CV = \frac{\sigma_s}{\bar{X}_s}$

3.4.1 染色體設計

本節介紹基因演算法中用來進行演化的染色體設計方式，所謂的染色體 (chromosomes)，指的是在執行基因演算法前，必須透過編碼的程序，將決策變數轉成可以代表可行解的字串，而此字串稱為染色體。

在本研究中，染色體就代表著一組待加工的工件，如圖 3.2，為染色體表達方式，其排列的方式便是工件加工時的順序。染色體中的基因格便代表著一個工件，彼此的順序不固定，染色體的長度由工件數量來決定，數量越多染色體越長。舉例來說，第一個基因格編號為 J5 便代表著工件 5。

染色體

J5	J3	J7	J1	J6	J4	J8	J2
----	----	----	----	----	----	----	----

圖 3.2 染色體設計方式

後續進行基因演算法的時候，只要透過這條染色體的演化，運用下一節的解讀方式，便能同時代表工件加工途程與加工順序的意義。

3.4.2 染色體解讀

在過去研究中，染色體只要透過不同的解讀方式便可以呈現不同的意義，而在本論文中，透過不同解讀方式可以同時呈現本研究的兩個決策方向：加工途程與加工順序。圖3.3代表的是染色體的兩種解讀方式，本研究的貢獻之一就是只要透過一條染色體，就能同時針對加工途程(Route Assignment)與加工順序(Job Sequencing)做解讀。

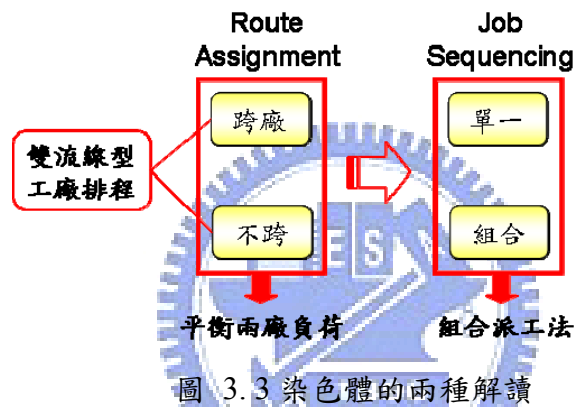


圖 3.3 染色體的兩種解讀

在加工途程的部份，使用平衡兩廠加工時間負荷的方式進行分廠，而且同時針對跨廠與不跨廠進行求解，最後將結果最好的作為最後的加工途程方式，所以這樣的加工途程解讀下，可以為工廠提出最佳的跨廠建議，如果跨廠績效值較佳，便選擇跨廠；反之，便選擇不跨廠。

在加工順序方面，使用組合派工法，將多種不同的單一派工法進行組合，透過混合實驗與反應曲面最佳化找出最適合的權重值，而此權重值便反映了應該使用何種派工法。當權重值偏向單一派工法，則建議加工順序方面使用單一派工法來進行排程；若偏向多種單一法則的組合權重，則建議使用組合過後的新權重。由此，只要不管是任何工件組合，組合派工法都可以給予最佳的派工方式建議。

以下將針對此兩種解讀分別進行詳細的介紹。

3.4.3 加工途程

所謂的加工途程，指的是每個工件在進行加工時所會經過的路徑。在本研究中，因為涉及跨廠觀念，而且每一廠都是 Flow Shop 的生產方式，所以可以使用另一種觀點來進行說明：所謂的加工途程，指的就是決定工件在每一站的加工位置，也就是決定每一站中，工件的分廠位置。

工件分廠的方式，主要依循一個觀念「分廠後兩廠各自的總負荷時間能最接近」，兩個能互相進行跨廠支援的廠，如果隨意的進行分廠，可能會造成某一廠負荷過重，當負荷過重時可以透過跨廠將工件交由他廠來做，但是因為工件的跨廠必須考量運輸時間，可能造成最後完工產品交期的延誤，喪失訂單的可能。因此工件分廠的方式是必須被慎重考量的，本研究希望讓工件分完廠後，兩廠的負荷量能夠相同，所以將平衡負荷量的方式作為分廠的準則。在決定機台負荷量時，本論文認為所謂負荷量也可以視為機台運作的程度，當一台機台一天工作 24 小時才能如期完工，相較於另一台一天只運作 20 小時便可完工，兩者相比，可以明顯看出前一部機台負荷量較大，所以本研究將負荷量以兩廠各站之機台的加工時間多寡來替代。因此，本研究針對工件分廠模式可以建構如下：

Stage 1 :

$$\text{Min} \left| \sum_{s=1}^{k_1} P_{i,s,1,1} - \sum_{s=1}^{o-k_1} P_{i,s,1,2} \right| \quad \text{for } i=1 \sim n \quad (3.1)$$

Stage 2 :

$$\text{Min} \left| \left(\sum_{s=1}^{k_1} P_{i,s,1,1} + \sum_{s=1}^{k_2} P_{i,s,2,1} \right) - \left(\sum_{s=1}^{o-k_1} P_{i,s,1,2} + \sum_{s=1}^{o-k_2} P_{i,s,2,2} \right) \right| \quad \text{for } i=1 \sim n \quad (3.2)$$

Stage m :

$$\text{Min} \left| \left(\sum_{s=1}^{k_1} P_{i,s,1,1} + \sum_{s=1}^{k_2} P_{i,s,2,1} + \cdots + \sum_{s=1}^{k_m} P_{i,s,m,1} \right) - \left(\sum_{s=1}^{o-k_1} P_{i,s,1,2} + \sum_{s=1}^{o-k_2} P_{i,s,2,2} + \cdots + \sum_{s=1}^{o-k_m} P_{i,s,m,2} \right) \right| \quad \text{for } i=1 \sim n \quad (3.3)$$

在進行分廠之前，會先將所有的工件*i*都以隨機的方式重排，重排的的目的是為了讓後續的分廠能嘗試多種不同的組合，而這部份由 3.3 節可得知其組合很多，所以本研究將採用基因演算法來進行此部分的排列。

工件在重排過後便可進到分廠的部份，方程式(3.1)到(3.3)各自呈現 Stage 1 到 Stage *m* 的分廠模式，在方程式中， $P_{i,s,j,f}$ 表示工件*i*在工廠*f*第*j*部機台，加工順序*s*的製造時間，所以 $\sum_{s=1}^{k_1} P_{i,s,1,1}$ 代表的是由隨機排列的工件組合中，在工廠 A、Stage 1 中的加工時間累積至第 k_1 個工件(k_1 個隨機排列的工件)，如圖 3.4 中的前半段，中間那條序為染色體，也就是待加工的工件序，J3 即為工件編號 3；而 $\sum_{s=1}^{o-k_1} P_{i,s,1,2}$ 即代表在工廠 B、Stage 1 中的加工時間，由圖 3.4 中的切點開始算起，累積至第 $o-k_1$ 個工件。由於本論文分廠所依循的觀念是「分廠後兩廠各自的總負荷時間能最接近」，為了讓分廠後兩廠的負荷時間差不多，所以在(3.1)式中，便將兩廠的總加工時間進行相減，並取最小值，因為只希望最小化兩廠總加工時間的差距，不管相減之後的正負值，所以便將此差距取絕對值，再進行最小化的動作。由圖 3.4 來看即工件由左側(A 廠)開始累加加工時間，並隨時與右側(B 廠)相比較，找到兩者差距最小的點即為 Stage 1 的切點(k_1)。

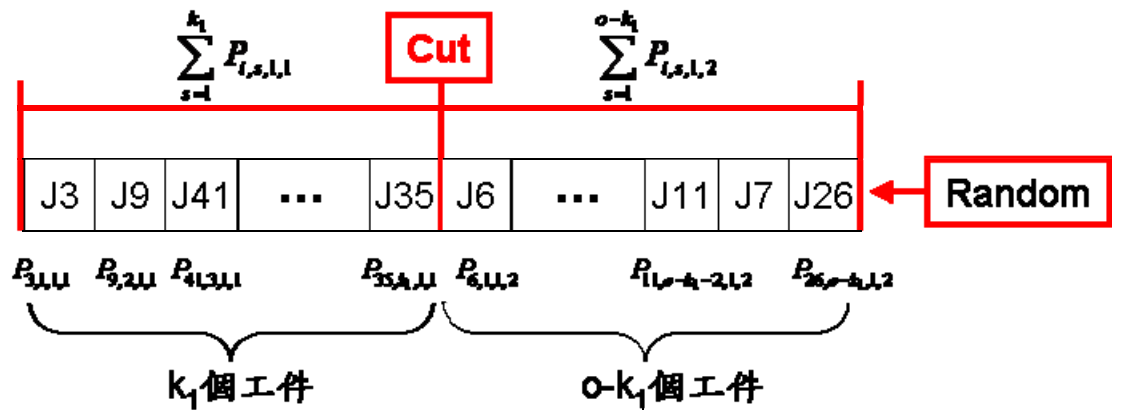


圖 3.4 Stage 1 分廠示意圖

由上述觀點，透過(3.1)式便可求出所有工件在 Stage 1 的分廠情形，也就是能夠求出 k_1 之值，因為希望能以全廠的觀點來進行分廠考量，所以做完 Stage 1 的分廠之後，接著可以進行 Stage 2 求解(3.2)式中的 k_2 值，求解時會將 Stage 1 的加工時間再拿來與 Stage 2 相加總，然後才能進行 k_2 的求取，一階一階的算出每一 Stage 的分廠位置，等到加總至 Stage j ，便完成了兩廠每一 Stage 的分廠。如(3.3)式子前半段括號中的

$\sum_{s=1}^{k_1} P_{i,s,1,1} + \sum_{s=1}^{k_2} P_{i,s,2,1} + \dots + \sum_{s=1}^{k_m} P_{i,s,m,1}$ ，此式即為整個 A 廠在 Stage j 中的總加工時間，而後半段括號中的 $\sum_{s=1}^{o-k_1} P_{i,s,1,2} + \sum_{s=1}^{o-k_2} P_{i,s,2,2} + \dots + \sum_{s=1}^{o-k_m} P_{i,s,m,2}$ 為整個 B 廠的總加工時間，兩者相減取最小值，便能讓兩廠負荷時間最為接近。

本研究在基因演算法中，會分兩支線同時求解跨廠與不跨廠的分廠模式，針對跨廠的部份，採取上述分廠方式；不跨廠，則三站皆採用第一站的分廠方式，由於透過上述方程式來進行跨廠求解，可能會遇到三站切點相同的情形，此時便會將其列為不跨廠的解。由此，本研究便可以同時計算跨廠與不跨廠之績效，再透過基因演算法篩選出最佳的結果，作為工廠加工途程之建議。

加工途程案例說明

圖 3.5 以八個工件、三個 Stage 的例子來做說明，工件 J1 至 J8 做隨機排列後，希望由此來決定三個 Stage 的分廠情形：

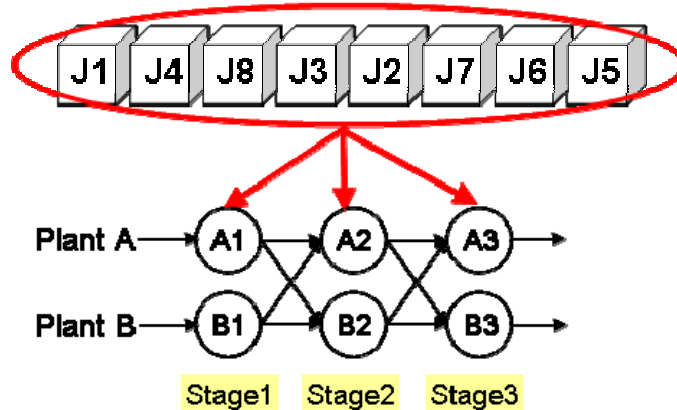


圖 3.5 8 個 Job 的分廠示意圖

作法：表 3.1 為工件 1 至 8 隨機排列後 Stage 1 的加工時間與累積加工時間表，表中第一列為工件序，第二列與第五列為工件各自在 A 廠與 B 廠 Stage 1 中的加工時間，也就是方程式中的 $P_{i,s,j,f}$ ，第三列與第四列為 A 廠與 B 廠的累積加工時間，即為方程式 $\sum_{s=1}^{k_i} P_{i,s,1,1}$ ，第三列由左邊開始累計，第四列由右邊開始累計。此表的目的是為了找到一個最好的切點，能夠讓 A 廠(第三列)與 B 廠(第四列)所累計的加工時間最為接近，也就是希望達到(3.1)式中的結果。

表 3.1 Stage 1 加工時間與累積加工時間表

Job	1	4	8	3	2	7	6	5
Fab A	1.2	1.5	1.6	1.6	1.5	1.7	1.6	1.3
累積 A	1.2	2.7	4.3	5.9	7.4	9.1	10.7	12
累積 B	10.5	9.5	8.2	6.8	5.3	4	2.5	1.1
Fab B	1	1.3	1.4	1.5	1.3	1.5	1.4	1.1

表 3.2 切割點於 J1、J4 中間

Job	1	4	8	3	2	7	6	5
Fab A	1.2	1.5	1.6	1.6	1.5	1.7	1.6	1.3
累積 A	1.2	2.7	4.3	5.9	7.4	9.1	10.7	12
累積 B	10.5	9.5	8.2	6.8	5.3	4	2.5	1.1
Fab B	1	1.3	1.4	1.5	1.3	1.5	1.4	1.1

表 3.3 切割點於 J3、J2 中間

Job	1	4	8	3	2	7	6	5
Fab A	1.2	1.5	1.6	1.6	1.5	1.7	1.6	1.3
累積 A	1.2	2.7	4.3	5.9	7.4	9.1	10.7	12
累積 B	10.5	9.5	8.2	6.8	5.3	4	2.5	1.1
Fab B	1	1.3	1.4	1.5	1.3	1.5	1.4	1.1

表 3.2 首先測試切於 J1 與 J4 間兩廠之差距，發現 A 廠累計值為 1.2，而 B 廠累計值為 9.5，所以此切點讓兩廠 Stage 1 加工時間差了 $9.5-1.2=8.3$ ；繼續嘗試下個切點，

直到表 3.3 切於 J3 與 J2 之間時，發現 A 廠累計為 5.9，B 廠累計為 5.3，兩者差距為 $5.9-5.3=0.6$ ，是所有切點中差距最小的，所以便選擇此點作為 Stage 1 的分廠點。

找完 Stage 1 的切點後，再來就是依照同樣的步驟去找出 Stage 2 的分割點，表 3.4 為 Stage 2 相關資訊，因為要尋找能使整廠加工時間差距最小的切點，所以第二站在考量累積加工時間時要將第一站的累積時間也考量進來，如表 3.4 第三列，A 廠的 J1 累積加工時間為 Stage 1 中 A 廠最佳切點的累積時間 5.9，加上 Stage 2 在 A 廠的加工時間 2.1，所以第三列的第一格為 $5.9+2.1=8$ ，其他的欄位都與 Stage 1 作法相同，由方程式 (2) 可以求出兩廠在 Stage 2 的最佳分廠點在 J8 與 J3 之間(表 3.5)，經由此步驟計算下去便可將三個 Stage 的分廠點都求出，表 3.6 即為三個 Stage 分廠結果。而圖 3.3 為染色體三站加工途程示意圖。

表 3.4 Stage 2 加工時間與累積加工時間表

Job	1	4	8	3	2	7	6	5
Fab A	2.1	1.8	1.9	1.9	2.1	2	1.9	1.9
累積 A	8	9.8	11.7	13.6	15.7	17.2	19.6	21.5
累積 B	17.9	16.3	14.8	13.3	11.8	10.2	8.6	6.9
Fab B	1.6	1.5	1.5	1.5	1.6	1.6	1.7	1.6

表 3.5 Stage 2 最佳切割點

Job	1	4	8	3	2	7	6	5
Fab A	2.1	1.8	1.9	1.9	2.1	2	1.9	1.9
累積 A	8	9.8	11.7	13.6	15.7	17.2	19.6	21.5
累積 B	17.9	16.3	14.8	13.3	11.8	10.2	8.6	6.9
Fab B	1.6	1.5	1.5	1.5	1.6	1.6	1.7	1.6

表 3.6 分廠結果

	Stage 1	Stage 2	Stage 3
A 廠	1,4,8,3	1,4,8	1,4,8,3
B 廠	2,7,6,5	3,2,7,6,5	2,7,8,3

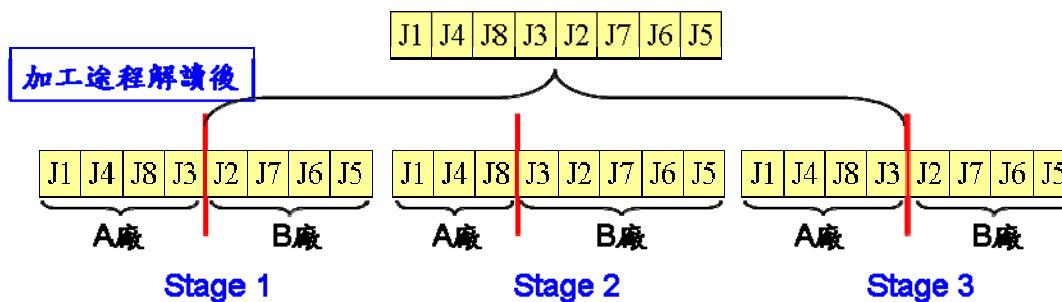


圖 3.6 染色體三站加工途程示意圖

3.4.4 加工順序

所謂的加工途程，指的是工件進到各站時的先後加工順序。在本節將針對本研究的重點——組合派工法，進行詳細的介紹。

一、組合派工法之定義

組合派工法指的是將多種不同的派工法則（如 EDD、SPT、Slack Time... 等）依某種方法進行組合，藉此整合成一項新的派工準則。而整合後的派工準則，其特性是可以包含所有單一的派工法或是重新創造出一項新的規則。

本論文的績效指標為最小化寬裕度的變異係數(CV)，將針對這點選定與績效指標有關的三種派工準則來進行組合，藉此分析兩廠應該如何來訂定彼此間的生產排程，為了將三項決策準則整合成單一準則，必須分別對此三項決策準則給予一個權重值，而不同的權重組合將導致不同的 CV 值，因此本章擬找出三項決策準則的最佳權重組合(optimal weighting portfolio)，期能最小化所有工件間績效值。

後續更藉由此研究來指出績效指標好壞會受到許多因素的影響，單單僅用一項派工準則來進行排程的規劃，並不夠具有說服力。為加強驗證這一點，本研究後續將針對單一派工法則與組合派工法則兩種模式進行不同情境的實驗比較，藉此強調組合派工法相較於單一準則所能適用的情境更廣，並且對於排程而言，能夠最小化 CV 的績效值。

二、組合派工法研究架構

組合派工法的主要架構分作三個部份：(1) 單一派工準則的選定；(2) 數值轉換；(3) 設計實驗尋找最佳權重組合。找出最佳權重組合之後，便可將所選定的派工準則依照此權重來組合成一個新的派工準則，接著便可以將此新的派工準則運用在跨廠排程工件排序的部份，強化本研究之績效。

以下將針對上述三點，做更詳細的介紹。

(1) 派工準則的選定

進行派工準則的選定，其所憑藉的依據是此派工準則對於績效指標是否會有所影響，如果會造成影響，則應該選取，如果不會造成影響，那麼此派工準則便沒有被選取的必要。

所以在本研究下，績效指標為寬裕度的 CV 值， $CV = \frac{\sigma_s}{X_s}$ ，寬裕度 = 完工時間-交期。會影響此績效值的派工法則很多，本研究找出三種差異度很大的派工法則來進行組合派工法的分析與研究：



1. Earliest Due Date (EDD)

Due Date 越早的工件，讓他能越先被加工。換句話說就是緊急的先做，所以具有較早交期的工件，也能較早完工。

舉例來說，圖 3.4 代表的是三個工件(J1、J2、J3)的相關資訊，上方為工件交期，下方為加工時間。依照 EDD 法則來排，排法為 J1→J2→J3，績效指標

$$CV_1 = \frac{1.16}{1.67} = 0.69 \quad ; \quad \text{若不按交期長短排列，例如 } J3 \rightarrow J2 \rightarrow J1, \text{ 則 } CV_3 = \frac{4.51}{1.33} = 3.38 \quad ,$$

按照 EDD 排列的績效值較佳。由此可以發現 EDD 派工法則對於績效指標 CV 是會有影響的。

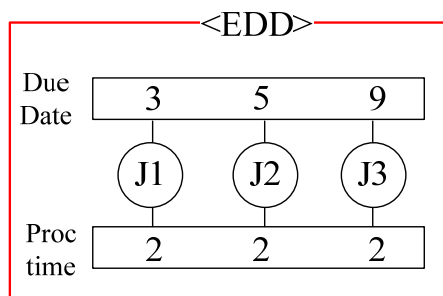


圖 3.7 EDD 派工法說明案例

2. Shortest Processing Time (SPT)

加工時間越短的工件越先加工。此派工法則讓加工時間短的先做，所帶來的好處是能讓後面的工件不必在其後方等候太久，節省 Queuing time，較不會延誤到後方工件的完工時間，平均數變大。如圖 3.5，最短加工時間之工件為 J1，所以 J1→J2→J3，績效為 $CV_1 = \frac{4.04}{4.33} = 0.93$ ；反之，績效為 $CV_3 = \frac{2.08}{1.67} = 1.25$ ，按照 SPT 排列的績效值較佳。由此可以發現 SPT 派工法則對於績效指標 CV 也是會有影響的。

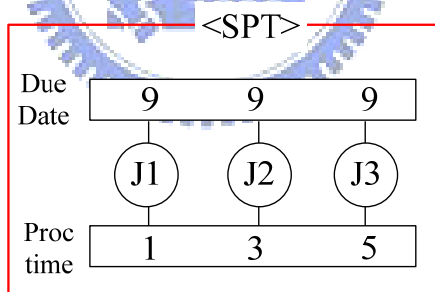


圖 3.8 SPT 派工法說明案例

本研究所選用的 SPT 並非如過去所使用的當站加工時間最小最先做，而是採用第一站加工時間最小的最先加工。原因是因為原本的 SPT 是屬於動態的，每一站要進新工件加工時都會判斷一次工件在當站的加工時間長短，而本論文屬於預排排程，要事先決定好所有工件的順序，所以無法每站都去重排一次，所以本研究便選用第一站的加工時間作為 SPT 的派工標準。

3. Least Slack First (LSF)

寬裕時間越短之工件，越先加工。此派工法則與 EDD 有相同的概念，也就是緊急的先做，所謂的寬裕時間就是交期減去加工時間，當進行排程時如果讓寬裕時間越短的越先做，可以避免快到期的工件延誤交期，以最優先的方式讓其出貨，所以不僅顧及交期，也顧及完工時間。如圖 3.6，最短 Slack 之工件為 J3，所以 $J3 \rightarrow J2 \rightarrow J1$ ，績效為 $CV_3 = \frac{0.58}{4.33} = 0.13$ ；反之，績效為 $CV_1 = \frac{3}{6} = 0.5$ ，按照 LSF 排列的績效值較佳。由此可以發現 SPT 派工法則對於績效指標 CV 也是會有影響的。

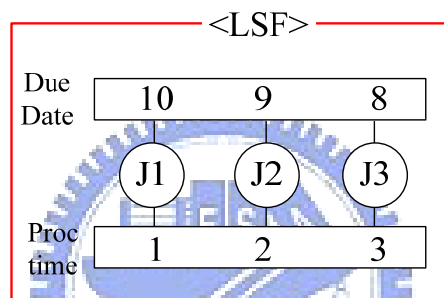


圖 3.9 LSF 派工法說明案例

(2) 數值轉換

決定好三項影響績效指標的派工法則—EDD、SPT、LSF，接下來要針對三者進行不同權重的組合，在這之前必須先進行資料的轉換，轉換的原因是由於原本的派工準則彼此之間都是相對性的指標，所以各自在單位上或是數值比例上並不相同。例如 EDD 看的是交期，而交期是以天為單位，而 SPT 看的是加工時間，所以是以小時或分鐘為單位，雖然兩者都是時間上的單位，但是其數值比例卻相差很多，以天計算的時間長度遠比分鐘計算來的高出許多，所以必須經由數值轉換，將這三個派工準則都轉成比例或單位都相同的絕對指標，才能進行權重的合併。

轉換公式

當單位或比例大小不同時，便無法用權重來進行重組，所以本研究提出運用標準化的方式來進行數值的轉換，其轉換的公式如下：

$$\text{標準化公式：} \frac{x_i - \bar{x}}{\sigma_x} \quad (3.4)$$

其中，符號假設如下：

i ：工件編號。

x_i ：工件 i 的 Due Date、Processing time 或 Slack 數值。

\bar{x} ：所有 x_i 的平均數。

σ_x ：所有 x_i 的標準差。

當數值經由標準化轉換之後，三種不同派工準則便都落於相同的基準，具有相同的平均數與標準差，所以便能透過之後的步驟繼續求算出最佳權重組合。

案例解說

運用第三章 8 個工件的案例來說明資料轉換的過程。表 3.7 與 3.8 為 8 個工件分廠後與工件的相關資訊：

表 3.7 8 個工件分廠資訊

	Stage 1	Stage 2	Stage 3
A 廠	1,4,8,3	1,4,8	1,4,8,3
B 廠	2,7,6,5	3,2,7,6,5	2,7,8,3

表 3.8 8 個工件加工時間與交期

	Stage 1		Stage 2		Stage 3		Due Date
	A	B	A	B	A	B	
1	1.2	1.0	2.1	1.6	1.6	1.6	10
2	1.5	1.3	2.1	1.6	1.6	1.6	11
3	1.6	1.5	1.9	1.5	1.5	1.5	12
4	1.5	1.3	1.8	1.5	1.7	1.7	13
5	1.3	1.1	1.9	1.6	1.3	1.3	14
6	1.6	1.4	1.9	1.7	1.7	1.7	15
7	1.7	1.5	2.0	1.6	1.6	1.6	16
8	1.6	1.4	1.9	1.5	1.4	1.4	17

因為要進行三種派工法資料的轉換，所以首先必須先找出與派工法相關的數值：(1) EDD：交期日；(2) SPT：總加工時間；(3) LSF：交期-總加工時間。

表 3.9 更進一步將上述資訊整理成一份表格。在表中的二、三、四欄記錄著表 3.7、3.8 的資訊，其中的 A、B 表示為廠別，數值則是加工時間，而五、六、七欄位記錄著上述三項派工法則的相關數值。第五欄為 SPT 的總加工時間，第六欄為 EDD 的交期日，第七欄為 LSF 所需的寬裕時間，此三項數值即為方程式(3.4)中的 x_i 。這三項數值求算出來後，因為要進行標準化動作，所以還必須求算出三者的平均數 \bar{x} 與標準差 σ_x ，因此我們可以透過表 3.9 得知 SPT 的平均數為 4.5125，標準差為 0.4155；EDD 平均數為 13.5，標準差為 2.4495；寬裕時間平均數為 8.9875，標準差為 2.5887。

表 3.9 平均數與標準差的計算

Job	Step 1	Step 2	Step 3	Sum	DD	DD-Sum
1	A 1.2	A 2.1	A 1.6	4.9	10	5.1
2	B 1.3	B 1.6	B 1.5	4.4	11	6.6
3	A 1.6	B 1.5	A 1.5	4.6	12	7.4
4	A 1.5	A 1.8	A 1.7	5	13	8
5	B 1.1	B 1.6	B 1.2	3.9	14	10.1
6	B 1.4	B 1.7	B 1.3	4.4	15	10.6
7	B 1.5	B 1.6	B 0.9	4	16	12
8	A 1.6	A 1.9	A 1.4	4.9	17	12.1
Average				4.5125	13.5	8.9875
Stdev				0.4155	2.4495	2.58868

平均數與標準差計算完成後，只要代入標準化公式即可求出三者標準化的數值(如表 3.10)。舉例來說，Job 1 要求取 SPT 的標準化數值，其求法為 $\frac{4.9-4.5125}{0.4155} = 0.2146$ ，如表 3.10 SPT 的第一格所示。

表 3.10 標準化後的數值

標準化			
Job	DD	SPT	LSF
1	-1.4289	0.2146	-1.3160
2	-1.0206	1.0730	-1.0584
3	-0.6124	0.7869	-0.6534
4	-0.2041	1.3591	-0.3589
5	0.2041	-0.9299	0.3037
6	0.6124	-0.3577	0.5982
7	1.0206	-1.2161	1.0768
8	1.4289	-0.9299	1.4081

(3) 設計實驗尋找最佳權重組合

解法說明

在完成數值標準化的步驟之後，下一步便是進行不同權重的組合，組合的公式如下：

$$priority = \alpha \times EDD + \beta \times SPT + \gamma \times LSF \quad (3.5)$$

$$\alpha + \beta + \gamma = 1 \quad (3.6)$$

$$0 \leq \alpha, \beta, \gamma \leq 1 \quad (3.7)$$

方程式(3.5)中的 EDD、SPT、LSF 三者的值便是前一節所提到的標準化過後的數值，三個數值都各自乘上一個比重，組合成新的派工準則。此派工比重 (α, β, γ) 須滿足另外一個限制條件，即(3.6)的 $0 \leq \alpha, \beta, \gamma \leq 1$ ，所以為了求出能使 CV 值最小化的組合派工法，必須在解空間 $S = \{(\alpha, \beta, \gamma) | \alpha + \beta + \gamma = 1, 0 \leq \alpha, \beta, \gamma \leq 1\}$ 中，找出一組最佳的權重組合 $(\alpha^*, \beta^*, \gamma^*)$ 。

為了搜尋這組最佳權重組合，本研究採取實驗設計(design of experiments, DOE)的方式，在眾多的解空間 S 中，找尋幾個適當的權重組合組成實驗點，然後再利用這些實驗點進行基因演算法的模擬，找出每個實驗點下的 CV 值，然後依據此實驗結果取得之權重組合與 CV 之資料，建構一個配適的反應曲面(response surface method, RSM)，求解出反應曲面中每個數值的最佳解，便是本研究所要求取的最佳權重組合 $(\alpha^*, \beta^*, \gamma^*)$ 。

為了能夠精確的找出此反應曲面的模式，必須要選擇適當的實驗點來做實驗，如此不僅能讓最佳權重組合更為確實，更能減少不必要的實驗時間。在本研究中，因為 α, β, γ 三者之間的關係是 $\alpha + \beta + \gamma = 1$ 且 $0 \leq \alpha, \beta, \gamma \leq 1$ ，因此，三個權重便可進行混合實驗設計。

本研究將採用混合設計中的擴充單體晶格設計(Augmented Simplex-Lattice Design)來進行實驗設計與分析(如圖 3.7)。經由擴充單體晶格設計可以找出 10 個實驗點來進行實驗設計(如表 3.11)，此 10 個實驗點可以建構出一個適合於混合實驗的反應曲面，最後藉由此反應曲面之配適方程式(3.8)，找出最佳權重組合 $(\alpha^*, \beta^*, \gamma^*)$ 。

$$\eta = \sum_{i=1}^q \beta_i x_i + \sum_{i < j} \sum_{i < j}^q \beta_{ij} x_i x_j + \sum_{i < j < k} \sum_{i < j < k} \beta_{ijk} x_i x_j x_k \quad (3.8)$$

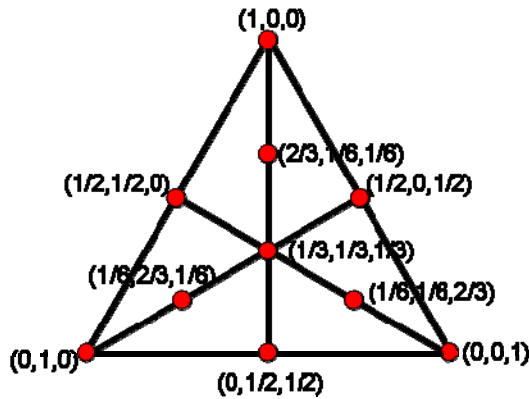


圖 3.10 Augmented Simplex-Lattice Design

表 3.11 實驗點

	x1	x2	x3
1	1	0	0
2	0	1	0
3	0	0	1
4	1/2	1/2	0
5	1/2	0	1/2
6	0	1/2	1/2
7	2/3	1/6	1/6
8	1/6	2/3	1/6
9	1/6	1/6	2/3
10	1/3	1/3	1/3

二、案例解說

繼續藉由之前 8 個工件的案例來說明。先前進行到表 3.10 標準化後的數值，本節便是繼續找出每個工件的加工優序(priority)，如表 3.12，假設我們所求出的最佳權重 $(EDD, SPT, LSF) = (\alpha, \beta, \gamma) = (0.3, 0.2, 0.5)$ ，因此表 3.12 左邊的值便是表 3.10 乘上各自權重後的值，然後再透過 $priority = \alpha \times EDD + \beta \times SPT + \gamma \times LSF$ 進行加總。最後在進行優序的求取之前，必須先評斷每個派工準則是望大、望小還是望目，其中 EDD 是交期最接近的最先加工，也就是 Due Date 越小的越先加工，為望小值；SPT 為最短加工時間先加工，也是望小值；LSF 為最短寬裕時間先加工，亦是望小值，由於三者都是望小值，因此三個數值的加總也是越小的越先加工，所以在分配每個工件的加工優序時，由表 3.12 可知，最小的 Sum = -1.0438 加工優序為 1，以此類推便可求出每個工件的加工優序。

圖 3.8 為染色體的兩項解讀示意圖。

表 3.12 加工優序計算

乘上權重後的值			Sum	Priority
EDD	SPT	LSF		
-0.4287	0.0429	-0.6580	-1.0438	1
-0.3062	0.2146	-0.5292	-0.6208	2
-0.1837	0.1574	-0.3267	-0.3530	3
-0.0612	0.2718	-0.1795	0.0311	5
0.0612	-0.1860	0.1519	0.0271	4
0.1837	-0.0715	0.2991	0.4113	6
0.3062	-0.2432	0.5384	0.6014	7
0.4287	-0.1860	0.7040	0.9467	8

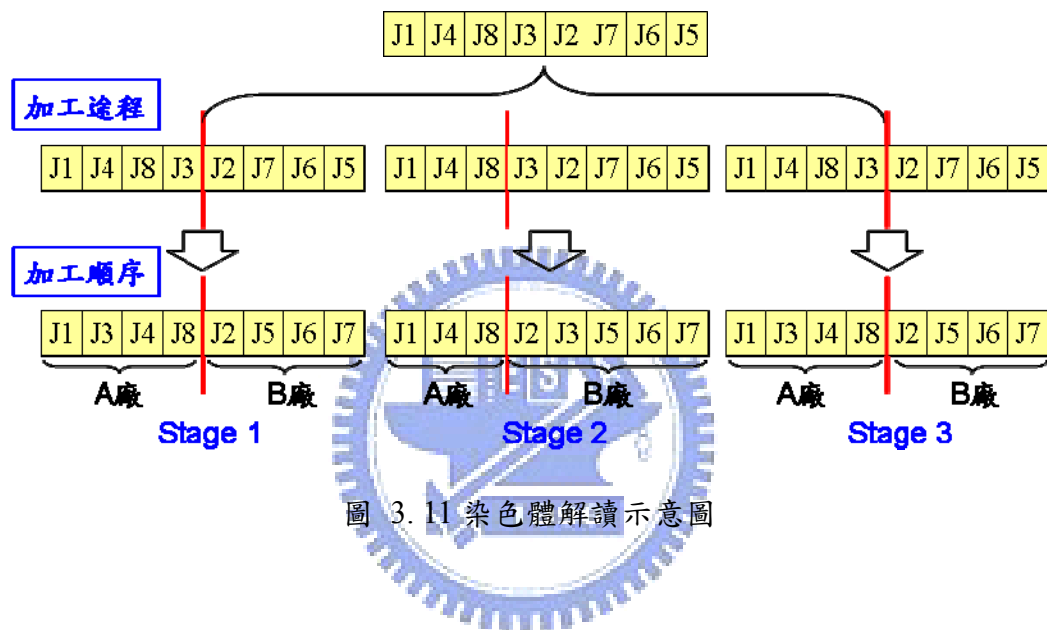


圖 3.11 染色體解讀示意圖

小結—染色體解讀

如上所述，本研究最初始產生的染色體為隨機排列，並無任何特殊意義，但是經由染色體分別作加工途程與加工順序的解讀後，同一條染色體卻能含有許多不同的意義。

以下小結本研究染色體的所有貢獻：

1. 加工途程：如圖 3.8，由加工途程解讀後，便可以得知各廠、各站加工工件為何。
2. 加工順序：如圖 3.8，由加工順序解讀後，便可得知各廠、各站工件加工的先後順序，只要工件來到便依照此順序進行加工。
3. 省時與擴充性：本研究只用一條染色體便可以表達兩廠、三站的工廠排程，對於基因演算法的演算時間來說，相較於每廠各一條染色體而言，省下了許多運算時間。

所以後續若要再將工廠或是站別加以擴充時，也只要運用一條染色體，便可以快速的找出多廠多站的生產排程。

3.4.5 適合度函數

完成工件加工途程與加工順序的解讀之後，接著要介紹基因演算法中，用來評估染色體績效值的適合度函數。

所謂的適合度函數，指的是用來評估所有染色體在族群中表現好壞之績效，表現好的染色體即表示適應力較佳，因此存活機率也較高，反之較差的染色體適應力較差，因此存活機率也較低。在求解過程中，會將染色體的表現轉換成適合度函數值，最後選取族群中適合度函數值最高的染色體當做下一代。

在本研究中的適合度函數值指的便是目標式 CV 值，在進行上述的染色體解讀之後，便可透過適合度函數 CV 值的計算，來評估此條染色體之優劣。求算 CV 值的步驟與其方程式與建構規則如下：



一、模式建構

Objective

$$\text{Min } CV_s = \frac{\sigma_s}{\bar{X}_s} \quad (3.9)$$

Fitness Function

$$ST_{i,1,1,f} = 0 \quad (3.10)$$

$$ST_{i,s,1,f} = CT_{i',s-1,1,f}, \text{ 其中 } s \neq 1 \quad (3.11)$$

$$ST_{i,1,j,f} = CT_{i,s,j-1,f'} + |f - f'| * TS, \text{ 其中 } j \neq 1 \quad (3.12)$$

$$ST_{i,s,j,f} = \text{Max}\{CT_{i,s',j-1,f'} + |f - f'| * TS, CT_{i',s-1,j,f}\} \quad (3.13)$$

$$CT_{i,s,j,f} = ST_{i,s,j,f} + P_{i,s,j,f} \quad (3.14)$$

二、建構規則

適合度函數的建構具有以下規則：

目標式：最小化寬裕度之 CV 值。(3.9)

1. Stage1 的第一個 Job 無須等候直接投入。(3.10)

→ Start time = 0

2. Stage1 其餘 Job 的 Start time 只需考慮上一個 Job 的 Complete time。(3.11)

3. Stage 2 以後，各 Stage 第一個加工 Job 之 start time，只需考慮該 Job 在前一 Stage 的 Complete time 與運輸時間。(3.12)

4. Stage 2 以後，非第一個加工 Job 之 start time，需考慮該 Job 在前一 Stage 的完工時間+運輸時間，及該 Stage 前一 Job 之完工時間，兩者取最大值。(3.13)

5. 所有 Job 的 Complete time = Start time + Processing time。(3.14)

三、案例說明

以下同樣運用先前 8 個工件的案例來說明上述方程式與建構規則，在此皆假設運輸時間為 0.05。

1. Stage1 的第一個 Job 無須等候直接投入，方程式(3.10)。

說明：

因為本研究假設為空廠開始進行排程，所以剛開始兩廠內都沒有半成品，所以當工件下來，要進行加工時，便無需等候，可以直接投入，所以開始加工時間

$ST_{i,1,1,f} = 0$ ，完工時間 $CT_{i,s,j,f} = ST_{i,s,j,f} + P_{i,s,j,f} = P_{i,s,j,f}$ 。

案例解說：

由表 3.13 可得知左側為 Stage 1，A 廠的待加工工件，順序為 J1、J3、J4、J8，右側為 B 廠的工件順序 J2、J5、J6、J7，因此第一個工件投入時(J1 與 J2)，因為是由空廠開始進行排程，再加上無跨廠所以運輸時間、到站時間與開始加工時間皆為 0，而完工時間則為開始時間加上製造時間。

表 3.13 Stage 1 首次派工情形

Stage 1									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	0	0	1.2	2	0	0	0	1.3
3					5				
4					6				
8									

$ST_{1,1,1} = 0$

$CT_{1,1,1,1} = ST_{1,1,1,1} + PT_{1,1,1,1}$

2. Stage 1 其餘 Job 的 Start time 只需考慮上一個 Job 的 Complete time，方程式(3.11)。

說明：

Stage 1 第一個工件加工完之後，下個工件的開始時間即為前一個工件的完工時間，如方程式(3.11) $ST_{i,s,1,f} = CT_{i',s-1,1,f}$ ，其 CT 的下標 i' 與 i 同樣工件的意思，但在這會用 i' 主要是為了與 ST 的下標 i 作區別，表示此兩工件是不同的，最後，方程式子中的 s 都不得為 1，因為如果 $s=1$ 就變成規則 1 的情形。

案例解說：

如表 3.14 所示，J3 要在 A 廠進行加工，必須先等到 J1 完工之後才能進入，所以 J3 的開始加工時間=J1(前一個工件)完工時間。

表 3.14 Stage 1 後續派工情形

Stage 1									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	0	0	1.2	2	0	0	0	1.3
3	0	0	1.2	2.8	5	0	0	1.3	2.4
4	0	0	2.8	4.3	6	0	0	2.4	3.8
8	0	0	4.3	5.9	7	0	0	3.8	5.3

$ST_{3,2,1,1} = CT_{1,1,1,1} = 1.2$

3. Stage 2 以後，各 Stage 第一個加工 Job 之 start time，只需考慮該 Job 在前一 Stage 的 Complete time 與運輸時間，方程式(3.12)。

說明：

工件在 Stage 1 加工完後，進入 Stage 2 時，如果是第一筆加工工件，則工件到站後不必等候馬上可進行加工，但是此工件的開始時間卻受制於前一 Stage 的完工時間 $CT_{i,s,j-1,f'}$ 與運輸時間 TS 。前一 Stage 何時完成，才能被運往下一個 Stage，又因為本研究跨廠時須考量運輸時間，所以如果工件在兩個 Stage 間有跨廠時，便必須將運輸時間加進去，所以方程式才會如(3.12)所示為，此方程式中， f 與 f' 都是代表工廠，會使用 f' 是為了區別是兩個廠 f 代表目前 Stage， f' 代表前一個 Stage，兩者可能相同也可能不同，如果 $f=f'$ 則不必考量運輸時間 TS ，否則兩者相減便會為 1，也就是開始時間必須加上運輸時間。

案例解說：

表 3.15 顯示 J1 由 Stage 1 要換到 Stage 2 的過程，當 J1 要進到 Stage2 的空機台時，他必須先在 Stage 1 完成加工，也就是要經過 1.2 的時間才能送往下一 Stage，又因為 J1 在 Stage 1 與 Stage 2 都是在 A 廠加工，不必跨廠的情況下，也就不用在加總運輸時間，所以 J1 在 Stage 2 的開始加工時間便等於 Stage 1 的完工時間 1.2。

表 3.15 Stage 2 首次派工情形

Stage 1									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	0	0	1.2	2	0	0	0	1.3
3	0	0	1.2	2.8	5	0	0	1.3	2.4
4	0	0	2.8	4.3	6	0	0	2.4	3.8
8	0	0	4.3	5.9	7	0	0	3.8	5.3

Stage 2									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	1.2	1.2	3.3	2	0	1.3	1.3	2.9
4					3				
					7				

$$ST_{1,1,2,1} = CT_{1,1,1,1} + |1-1| * 0.05$$

4. Stage 2 以後，非第一個加工 Job 之 start time，需考慮該 Job 在前一 Stage 的完工時間+運輸時間，及該 Stage 前一 Job 之完工時間，兩者取最大值，方程式(3.13)。

說明：

如果工件非此站第一個加工之工件，其開始加工時間的建構會比較複雜，此工件在衡量開始之時間時，因為我們無法得知工件何時會到達本站，所以必須考量兩種情況：第一種，如果此工件到達本站的時間比本站前一個工件的完工時間晚，表示說工件到達時機台是空的，那此工件的開始時間便只需考量他在前一 Stage 的完工時間 $CT_{i,s',j-1,f'}$ ，如果工件有經過跨廠，則必須再加上跨廠的運輸時間 TS ；第二種情形，如果此工件到達本站的時間比本站前一個工件的完工時間早，那表示說工件到達本站時，還必須等候前一個工件的完工才能開始進行加工，所以其開始加工時間便會等於前一個工件的完工時間 $CT_{i',s-1,j,f}$ 。所以方程式(3.13)必須先判斷哪種情形的時間會比較長，時間較長者便是此工件的開始時間。

案例解說：

如表 3.16，當 J3 要進到 Stage 2 進行加工時，由於必須跨廠，所以 J3 到達 Stage 2 的時間便為 Stage 1 完工時間 2.8 加上運輸時間 0.05 即為 J3 到 Stage 2 的時間 2.85，但是當 J3 到達時卻無法馬上進行加工，因為 Stage 2 中前一個工件 J2 還在進行加工，所以等到 J2 在 2.9 的時間點完工之後，J3 才得以進入機台中，所以他在 Stage 2 的開始加工時間便為 $\text{Min}\{2.8+0.05, 2.9\}=2.9$ 。

表 3.16 Stage 2 後續派工情形

Stage 1									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	0	0	1.2	2	0	0	0	1.3
3	0	0	1.2	2.8	5	0	0	1.3	2.4
4	0	0	2.8	4.3	6	0	0	2.4	3.8
8	0	0	4.3	5.9	7	0	0	3.8	5.3

加上

Stage 2									
A廠					B廠				
Job	運輸	Arrival	Start	End	Job	運輸	Arrival	Start	End
1	0	1.2	1.2	3.3	2	0	1.3	1.3	2.9
4	0	4.3	4.3	6.1	3	0.05	2.85	2.9	4.4
8	0	5.9	6.1	8	5	0	2.4	4.4	6
					6	0	3.8	6	7.7

$ST_{3,2,2,2} = \text{Max} \{CT_{3,2,1,1} + |1 - 2| * 0.05, CT_{2,1,2,2}\}$

5. 最小化寬裕度之 CV 值。(3.9)

說明：當所有工件在每一個 Stage 都經由上述的規則進行排列後，便可以得到所有工件的總完工時間 $CT_{i,s,j,f}$ ，工件在一開始顧客下單時便預先決定好了各自的交期 d_i ，由於寬裕度 = $d_i - CT_{i,s,j,f}$ ，所以要求最小化 CV 的公式

$$\text{即為 } \text{Min } CV_s = \frac{\sigma_s}{\bar{X}_s}。$$

案例說明：如表 3.17 可以很清楚的看到每個工件的完工時間與交期，最後一欄 (DD-End) 即為每個工件寬裕度的值，因此便可求出此組排程的績效值為

0.1463。

表 3.17 計算 CV

Job	End	DD	DD-End
1	4.9	10	5.1
2	4.4	11	6.6
3	6.4	12	5.6
4	8.1	13	4.9
5	7.2	14	6.8
6	9.0	15	6.0
7	10.2	16	5.8
8	9.5	17	7.5
所有 DD-End 之 CV 值			0.1463

3.4.6 基因演算法求解流程

本研究的基因演算法透過以下六個步驟進行不斷的進行重複求解，直到滿足終止條件為止。

步驟一：產生起始族群

在工件組合空間中，以亂數產生器隨機產生 N 條染色體(chromosomes)為初始群體，做為 GA 求解的起始點，在新染色體產生的途中，會進行染色體重複性檢查，以防止相同的染色體產生。

在 3.4.3 節進行分廠說明時，有提到工件必須先經過基因演算法隨機排序後再進行分廠，當時的隨機排序便是在此處與後續的交配突變進行，只要順序不同就能創造出多種不同的分廠方式。

步驟二：評估適合度函數 (Fitness Function)

計算現有族群中每一條染色體的適合度函數，此處的適合度函數即為 3.4.5 節所介紹的績效指標最小化寬裕度之 CV 值，CV 值是望小值，所以當 CV 越小，適合度函數也會越小，求解品質越佳。而先前所介紹的所有染色體解讀方式，也都是為了在此處求取 CV 值之前所做的一些前置步驟。

步驟三：演化

交配 (Crossover)

依照交配率，隨機選取兩條染色體進行交配，會產生兩條新的染色體，重複抽取直到符合交配率個數。交配的種類分為四種：單點交配(C1 Operator)、線性順序交配(LOX)、部分相應交配(PMX)、NABEL 交配(如圖 3.9)，在兩個染色體交配完後，會進行兩個新染色體重複性檢查，以防止相同的染色體產生。本研究使用線性順序交配(LOX)作為交配的方式。

LOX (Linear order crossover)



圖 3.12 線性順序交配(LOX)

突變 (Mutation)

依照突變率，隨機選取一條染色體做突變，會產生一條新的染色體，重複抽取直到符合突變率個數。本研究所使用的突變方法為 SWAP，如圖 3.10，即隨機由一條染色體中抽取兩個不重複的基因，彼此互相做位置的交換已達到突變的效果。由上述的交配與突變世代演化後，產生比母代更為優秀的子代，以形成新的群體繼續進行下個步驟。在每個染色體突變完後，也會進行染色體重複性檢查，以防止相同的染色體產生。

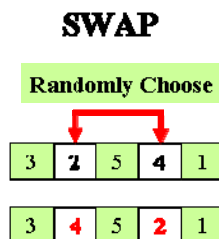


圖 3.13 SWAP 示意圖

步驟四：選擇 (Selection Strategy)

自現有染色體群體中使用選擇策略(Selection Strategy)去篩選出最適合的染色體，以適合度為評估準則，選擇適當的染色體繼續進行後續演化。過去所使用的選擇策略分為兩種，一種是俄羅斯輪盤法 (Roulette wheel selection)，另一種是競賽法(Tournament

selection)，本研究為了不讓演化提早進入區域最佳解，因此選用競賽法(Tournament selection)作為 Selection Strategy。

步驟五：測試停止條件

測試是否符合停止條件。若是則停止，完成尋優，若否則再次重複以上步驟。本研究將基因演算法的終止條件設定為：(1)當最佳解一直維持若干世代未改變時。(2)最大演化世代次數達到預設最大值。意指在基因的求解過程中，滿足上述的終止條件即可停止搜尋程序，同時可以找出最佳的排程方式。基因演算法概略流程如下：

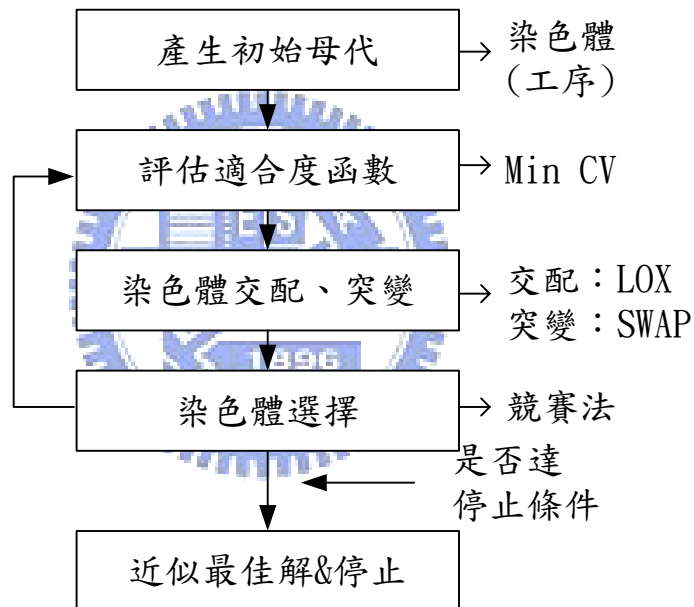


圖 3.14 基因演算法概略流程

3.5 小結

運用上述方法，便可為雙流線型工廠的排程提出排程以下建議：

1. 加工途程建議：透過加工途程的解讀，可以分析此工廠的生產型態是否適合跨廠生產，若跨廠績效值優於不跨廠，則建議進行跨廠；反之則不跨廠。

2. 加工順序建議：透過加工順序的解讀，可以決定出此工廠所有待加工工件的加工順序。只要透過混合實驗與反應曲面法求取權重值，由此權重值的特性，便可判斷工廠是適合於單一派工法或是組合式的派工法。

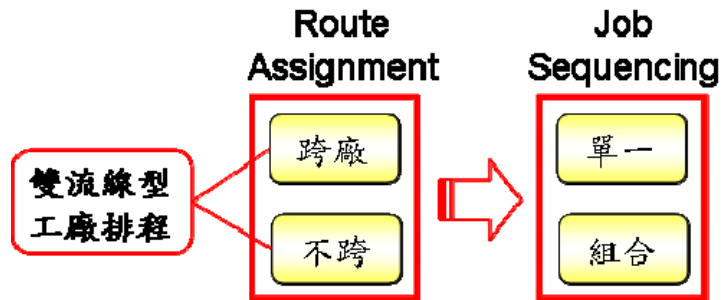


圖 3.15 本研究排程適用範圍

後續將在第四章進行不同情境下單一派工法與組合派工法的績效評比，藉此證明組合派工法在任何情境下，皆能找出最佳的派工方式。



第四章 實驗分析與比較

4.1 實驗假設

一、實驗模型設定

共有兩座具有相似機台功能的廠區，兩者可以進行相互的跨廠支援，針對兩廠的模擬資料設定如下：

1. 工廠設定

- A、B 兩廠皆為流線型工廠型態。

2. 兩廠機台設定：

- 各有 3 個 stage (3 道加工流程)，每個 stage 中只有一部機台。
- 兩廠各個 stage 機台相似。
- 兩廠機台生產效率比→A 廠：B 廠 = 1：1 與 1：3。

3. 加工時間與交期

- 皆採用 Uniform 分配。



二、基因演算法參數設定

1. 初始母代染色體數 100 條。
2. 交配率：0.8，突變率：0.2。
3. 子代數量 = 交配率*母代數量+突變率*母代數量。
4. 演算停止條件:執行的代數 = 100,000，最佳解維持代數= 1,000

4.2 實驗目的與情境

4.2.1 實驗目的

本研究想要驗證之目的有兩個：

1. 組合派工法相較於其他派工法，可提供最穩健且較佳的解。

2. 本研究之基因演算法可提供較合適的加工途程決策。

4.2.2 實驗標竿

由實驗目的來看，所謂的穩健，指的是組合派工法相較於其他派工法，能在各種情境下得到相對較佳的績效，並為工廠提供最合適的生產排程建議。

為驗證這點，本研究找出四種比較標竿如下：

- GA-EDD (最早交期最先加工)。
- GA-SPT (最短加工時間最先加工)。
- GA-LSF (最短寬裕時間最先加工)。
- GA-GA (基因演算法自然演化)。

本研究的組合派工法：

- GA-Comb (EDD、SPT、LSF 三派工法之組合)。

這四種標竿命名的方式為“加工途程-加工順序”，四種標竿的加工途程皆為 GA，指的是皆運用基因演算法進行各工件加工途程的分配，後面的代號則是派工的方法，分別為 EDD、SPT、LSF 以及 GA 自然演化法。前三項標竿都必須經由啟發式演算法進行排序；而 GA 自然演化，指的是運用基因演算法進行加工途程分配後，便交由 GA 內部的交配、突變自然演化各工件的加工順序。透過 GA-Comb 與四種標竿來進行不同情境的實驗，希望驗證 GA-Comb 在各種情境下的績效，都能有不錯的結果。

4.2.3 實驗情境

實驗情境的設計有兩個方向，分別針對各派工法適用情境與機台加工效率比兩者進行實驗，若組合派工法(GA-Comb)在以下情境皆能有不錯的績效值，則驗證了第一項目的；而透過機台加工效率之研究，可以驗證此基因演算法能夠提供較合適之加工途程決策。

以下針對四種不同的派工情境，以及兩種機台加工效率進行介紹，希望針對這四種特殊情境下，GA-Comb 皆能達到不錯的績效。

情境一：適用於 EDD 派工法之情境

資料設計原則：

- EDD 顯著：設定交期變異大。
- SPT 不顯著：設定第一站加工時間變異小。
- LSF 不顯著：設定總加工時間和交期正相關。

設計方法：

表 4.1 情境一工件加工時間與交期設計方法

	加工時間與交期設計方法	原則
交期	服從 Uniform 分配，全距 200 小時，由小到大排列。	交期變異大
第一站	服從 Uniform 分配，全距 5 小時。	第一站加工時間變異小
第二站	服從 Uniform 分配，全距 100 小時，皆由小到大排列。	總加工時間和交期正相關
第三站		

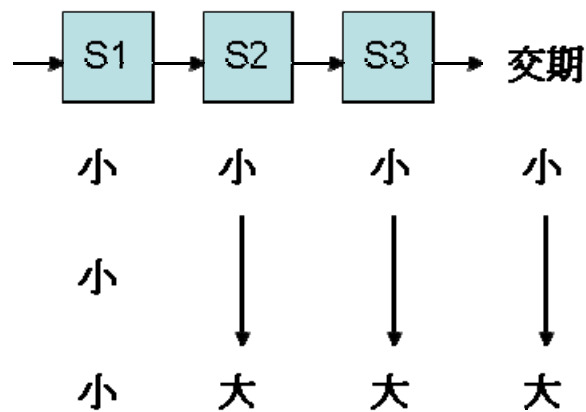


圖 4.1 情境一圖示

說明：

本研究的績效指標為最小化寬裕度的 CV 值，要讓 EDD 派工法在此情境下最好，就是要讓其 CV 值在三派工法中最小。針對這個目的，此情境的設計方向便是要讓與 EDD 最相關的交期，其各工件間能有顯著的差異，藉此才能突顯 EDD 派工法之績效。除此之外，為了讓 SPT 與 LSF 派工法不至於影響到 EDD 之績效，本研究將兩者設計成無顯著差異，針對 SPT 的部份，由於本研究的 SPT 看的是第一站的加工時間，為了讓它不顯著，所以便設定第一站加工時間變異小；針對 LSF 的部份，LSF 等於交期減去完工時間，但由於本研究是預排，所以便以總加工時間代替完工時間，為了讓 LSF 不顯著，便設定總加工時間與交期成正相關，由此讓每個工件的 Slack 都差不多，不會顯著影響績效指標。

情境二：適用於 SPT 派工法之情境

資料設計原則：

- EDD 不顯著：設定交期變異小。
- SPT 顯著：設定第一站加工時間變異大。
- LSF 不顯著：設定總加工時間和交期正相關。

設計方法：

表 4.2 情境二工件加工時間與交期設計方法

	加工時間與交期設計方法	原則
交期	服從 Uniform 分配，全距 20 小時，由大到小排列。	交期變異小
第一站	服從 Uniform 分配，全距 100 小時，由小到大排列。	第一站加工時間變異大
第二站	固定值，或是每個 Job 此兩站的加工時間總和很相近。	總加工時間和交期正相關
第三站		

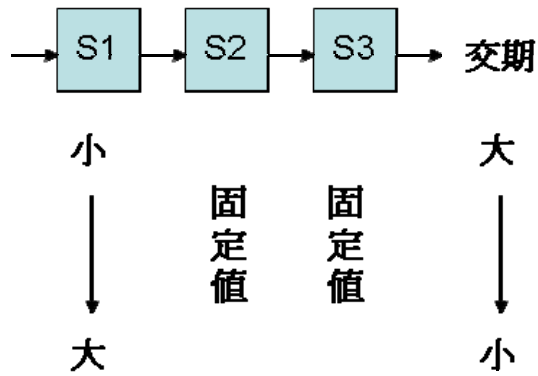


圖 4.2 情境二圖示

說明：

為了讓 SPT 派工法在 CV 績效指標下顯著，將設定第一站加工時間變異大，使得 SPT 能顯著影響績效值；要讓 EDD 不顯著，變設定與 EDD 最相關的交期變異小；要讓 LSF 也不顯著影響績效，便和情境一一樣，設定總加工時間和交期正相關。

情境三：適用於 LSF 派工法之情境

資料設計原則：

- EDD 不顯著：設定交期變異稍大，且有順序性，讓寬裕度間有差異。
- SPT 不顯著：設定第一站加工時間變異小，且有順序性。
- LSF 顯著：為了讓 LSF 有差異，且不和交期正相關，所以設定工件分成兩群，

詳細方法如表 4.3。

設計方法：

表 4.3 情境三工件加工時間與交期設計方法

	加工時間與交期設計方法	原則
交期	服從 Uniform 分配，全距 100 小時，由小到大排列。	交期變異大，有順序性
第一站	服從 Uniform 分配，全距 5 小時，由大到小排列。	第一站加工時間變異小，有順序性
第二站	分兩部分： 上半部，時間長，服從 Uniform 分配，全距 5 小時，由小到大排列。 下半部，時間短，服從 Uniform 分配，全距 5 小時，隨機排列。	讓 LSF 有差異，且不和交期正相關
第三站	分兩部分： 上半部，時間短，服從 Uniform 分配，全距 5 小時，隨機排列。 下半部，時間長，服從 Uniform 分配，全距 5 小時，由大到小排列。	

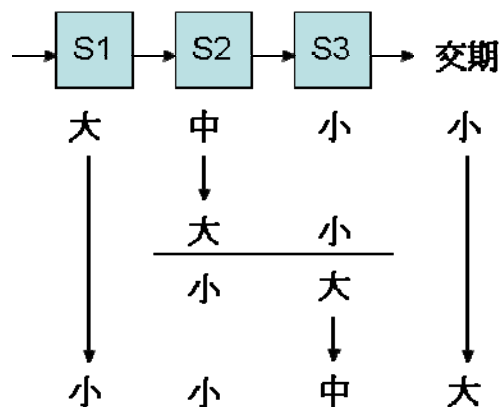


圖 4.3 情境三

說明：

LSF 的情境較難定義，由於交期對他有很大的影響性，所以要劃分開來要做點特殊設計。首先交期變異大，是為了讓寬裕度(交期-完工時間)有顯著性，設定第一站加工時間變異大，使得 SPT 能夠不顯著影響績效值，接著在二、三站的設定部份較為複雜，第二站的上半部與第三站的下半部，設定為數值較大，且具順序性，為的是讓 LSF 與 SPT 兩者做區別，當二、三站設定如表 4.3 的順序排列後，能讓 EDD 與 SPT 排序後加工順序與總加工時間相衝突，也就是會先進行時間較短之工件的加工，造成後面一些較花時間之工件無法如期完工，績效變差。

情境四：適用於組合式的派工法之情境

資料設計原則(圖 4.4)：

- EDD 不顯著：設定交期變異小
- SPT 不顯著：設定兩種工件類型，變異小
- LSF 不顯著：設定總加工時間變異小

設計方法：



表 4.4 情境四工件加工時間與交期設計方法

	加工時間與交期設計方法	原則
交期	服從 Uniform 分配，全距 20 小時。	交期變異小
第一站	分兩種 Job 類型： 第一種：三站時間小、小、大，皆服從 Uniform 分配，一、三站全距 5 小時，第二站為固定值。 第二種：三站時間大、小、小，皆服從 Uniform 分配，一、三站全距 5 小時，第二站為固定值。	兩種工件類型，變異小，而且總加工時間變異也小
第二站		
第三站		

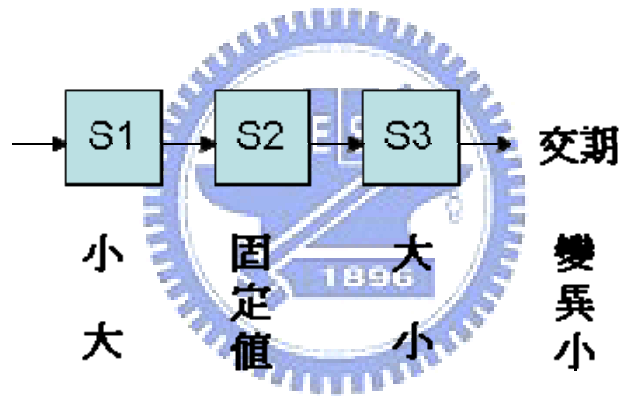


圖 4.4 情境四圖示

說明：

所謂的組合式派工法，指的是由三種單一派工法所跑出來的績效，並沒有誰的績效值較佳，所以最後使用混合實驗與反應曲面找出來的權重會平均分給三個派工法。要找出此情境，只要讓三種單一派工法彼此之間皆不顯著即可，所以要讓 EDD 不顯著，則設定交期變異小；要讓 SPT 不顯著，則設定第一站有大小工件之分，由於寬裕度各工件差不多，所以不管工件如何排，績效都不太顯著；要讓 LSF 不顯著，則是讓工件之間總加工時間變異小。

機台加工效率之實驗

在加工途程方面，第三章研究方法中有提到，在基因演算法進行演化時，同一條染色體會同時求解跨廠與不跨廠解，並將最佳的解當作本次的績效值，所以為了驗證基因演算法是否會提供較合適的途程決策，後續將針對上述所提到的四種情境，分別進行以下兩種實驗：

1. 機台加工效率協調的情境(較不易跨廠)。
2. 機台加工效率不協調的情境(較易跨廠)。

當機台加工效率協調時，代表兩廠機台加工速度一致，所以工件不管是在自己廠加工或跨廠加工，加工時間都是一樣的，由於跨廠需要計算跨廠時間，所以通常會選擇不跨廠；當加工效率不協調時，兩廠機台加工速度不一致下，可能某一廠比較慢做完，可能造成工件延誤交期，所以通常會建議工件進行跨廠。

在設計上，將加工效率協調設定為兩廠中間站加工效率比為 1:1，而不協調的情況設定為中間站兩廠機台效率為 1:3 來進行實驗。若這兩種加工效率實驗之結果，在加工效率不協調時傾向跨廠解較佳或是跨廠程度優於加工效率協調時，則驗證了本研究之基因演算法可針對不同情境提供較合適的加工途程決策。

4.3 實驗分析與比較

4.3.1 情境一

	20J	40J	60J	80J	100J
派工法	機台加工效率協調之 CV 值				
GA-EDD	0.890	0.805	1.190	0.932	0.907
GA-SPT	2.221	1.412	2.169	1.404	1.357
GA-LSF	9.038	1.823	6.712	3.849	4.070
GA-GA	0.891	0.789	1.191	0.929	0.902
GA-Comb	0.974	0.805	1.227	0.950	0.907

	20J	40J	60J	80J	100J
派工法	機台加工效率不協調之 CV 值				
GA-EDD	3.157	1.914	7.416	3.019	2.975
GA-SPT	-1.911	-38.640	-2.108	-2.639	-2.426
GA-LSF	-1.002	-1.121	-0.934	-0.908	-0.918
GA-GA	4.545	2.321	11.189	3.394	3.451
GA-Comb	3.390	1.975	7.416	3.253	3.038

表 4.5 情境一之派工法績效值(CV)

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	8.63%	0.00%	2.97%	1.94%	0.00%
GA-SPT	-127.96%	-75.40%	-76.85%	-47.79%	-49.57%
GA-LSF	-827.88%	-126.46%	-447.15%	-305.12%	-348.51%
GA-GA	8.53%	2.00%	2.95%	2.19%	0.58%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	6.89%	3.07%	0.00%	7.19%	2.06%
GA-SPT	*	*	*	*	*
GA-LSF	*	*	*	*	*
GA-GA	-34.06%	-17.55%	-50.87%	-4.33%	-13.60%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

表 4.6 情境一之派工法改善率

	(EDD, SPT, LSF)
20J	(0.7954, 0, 0.2046)
40J	(1, 0, 0)
60J	(0.8286, 0, 0.1714)
80J	(0.7589, 0, 0.2411)
100J	(1, 0, 0)

(a) 機台加工效率協調

	(EDD, SPT, LSF)
20J	(0.7954, 0, 0.2046)
40J	(1, 0, 0)
60J	(0.8286, 0, 0.1714)
80J	(0.7589, 0, 0.2411)
100J	(1, 0, 0)

(b) 機台加工效率不協調

表 4.7 情境一之最佳組合權重 (a) 效率協調 (b) 效率不協調

表 4.5 與表 4.6 前四格為比較標竿：GA-EDD、GA-SPT、GA-LSF、GA-GA，最後一格為組合派工法 GA-Comb 跑出來的組合權重之改善率與名次。

由表 4.5 與表 4.6 中發現加工效率協調時，在適用 EDD 派工法之情境下，GA-EDD、GA-GA 與 GA-Comb 的改善率都不錯，也相當的接近，三者間之績效並無顯著差異；當加工效率不協調時，三者開始有顯著的差異，GA-GA 之績效值降的很多變成三者中績效最差之派工法。

表 4.7 為 GA-Comb 在五種 Job 數下，進行混合實驗及反應曲面所求得的最佳組合權重。由表中可以明顯的發現，在 EDD 的適合情境下所求到的權重，五種 Job 數的值皆為 EDD 比重佔最高。理論上來看，在情境一的工件組合下，權重結果應該都要為(1, 0, 0)，之所以會出現其他權重值的原因是由於本研究使用反應曲面法去求取最佳解，由於反應曲面是運用方程式去近似最佳解，所以才會有些許誤差導致權重並非全都是(1, 0, 0)。



4.3.2 情境二

	20J	40J	60J	80J	100J
派工法	機台加工效率協調之 CV 值				
GA-EDD	10.516	27.958	10.976	23.387	21.309
GA-SPT	1.185	1.086	1.052	0.962	1.043
GA-LSF	11.666	44.311	11.973	26.637	31.186
GA-GA	1.186	1.086	1.051	0.961	1.043
GA-Comb	1.185	1.086	1.052	0.962	1.043

	20J	40J	60J	80J	100J
派工法	機台加工效率不協調之 CV 值				
GA-EDD	12.127	37.260	11.303	24.062	24.238
GA-SPT	1.188	1.089	1.053	0.963	1.044
GA-LSF	14.933	57.940	13.062	33.559	32.554
GA-GA	1.206	1.096	1.057	0.965	1.046
GA-Comb	1.188	1.089	1.053	0.963	1.044

表 4.8 情境二之派工法績效值(CV)

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-787.03%	-2474.11%	-943.49%	-2330.55%	-1942.94%
GA-SPT	0.00%	0.00%	0.00%	0.00%	0.00%
GA-LSF	-884.06%	-3979.63%	-1038.25%	-2668.28%	-2889.87%
GA-GA	-0.02%	0.01%	0.05%	0.13%	0.04%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-920.53%	-3322.62%	-972.92%	-2398.54%	-2221.70%
GA-SPT	0.00%	0.00%	0.00%	0.00%	0.00%
GA-LSF	-1156.70%	-5222.29%	-1139.89%	-3384.67%	-3018.27%
GA-GA	-1.45%	-0.65%	-0.36%	-0.17%	-0.16%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

表 4.9 情境二之派工法改善率

	(EDD, SPT, LSF)		(EDD, SPT, LSF)
20J	(0, 1, 0)	20J	(0, 1, 0)
40J	(0, 1, 0)	40J	(0, 1, 0)
60J	(0, 1, 0)	60J	(0, 1, 0)
80J	(0, 1, 0)	80J	(0, 1, 0)
100J	(0, 1, 0)	100J	(0, 1, 0)

(a) 機台加工效率協調

(b) 機台加工效率不協調

表 4.10 情境二之最佳組合權重 (a) 效率協調 (b) 效率不協調

由表 4.8 及表 4.9 可以發現，GA-SPT、GA-GA、GA-Comb 三者所求出來的績效與改善率，不管加工效率是否協調，結果幾乎是一樣的。而表 4.10 顯示在適用於 SPT 情境下，GA-Comb 的最佳權重全都落在 SPT 權重為 1 的情形，也就是說，GA-Comb 在情境二中，皆能找到最適合的派工法。

4.3.3 情境三

	20J	40J	60J	80J	100J
派工法	機台加工效率協調之 CV 值				
GA-EDD	0.560	0.411	0.573	0.668	0.625
GA-SPT	0.621	0.403	0.534	0.605	0.562
GA-LSF	0.439	0.334	0.457	0.531	0.505
GA-GA	0.455	0.333	0.457	0.518	0.494
GA-Comb	0.437	0.334	0.460	0.531	0.505

	20J	40J	60J	80J	100J
派工法	機台加工效率不協調之 CV 值				
GA-EDD	0.625	0.412	0.660	0.659	0.723
GA-SPT	0.772	0.357	0.926	0.910	0.994
GA-LSF	0.448	0.322	0.467	0.474	0.535
GA-GA	0.648	0.397	0.557	0.540	0.590
GA-Comb	0.439	0.308	0.493	0.537	0.526

表 4.11 情境三之派工法績效值(CV)

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-27.96%	-23.26%	-24.73%	-25.72%	-23.65%
GA-SPT	-41.97%	-20.77%	-16.14%	-13.85%	-11.17%
GA-LSF	-0.32%	0.00%	0.63%	0.00%	0.00%
GA-GA	-3.98%	0.10%	0.64%	2.38%	2.19%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-42.15%	-33.68%	-33.99%	-22.78%	-37.44%
GA-SPT	-75.79%	-15.94%	-88.00%	-69.38%	-88.80%
GA-LSF	-1.91%	-4.56%	5.16%	11.72%	-1.64%
GA-GA	-47.39%	-28.86%	-13.08%	-0.63%	-12.05%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

表 4.12 情境三之派工法改善率

	(EDD, SPT, LSF)
20J	(0, 0.1657, 0.8343)
40J	(0, 0, 1)
60J	(0, 0.0913, 0.9072)
80J	(0, 0, 1)
100J	(0, 0, 1)

(a) 機台加工效率協調

	(EDD, SPT, LSF)
20J	(0, 0.0928, 0.9072)
40J	(0.2982, 0.4510, 0.2508)
60J	(0.296, 0.4745, 0.2295)
80J	(0.2795, 0.4866, 0.2339)
100J	(0, 0.0976, 0.9024)

(b) 機台加工效率不協調

表 4.13 情境三之最佳組合權重 (a) 效率協調 (b) 效率不協調

在此情境下，也能看到 GA-LSF、GA-GA、GA-Comb 三者效率協調時，績效值都非常相近且解都很好；但不協調時 GA-GA 之績效明顯變差。表 4.13 也指出五種 Job 數下，效率協調時最佳組合權重都是 LSF 權重最高；效率不協調時，權重結果有所變動，但績效值卻能最接近 GA-LSF 之結果。



4.3.4 情境四

	20J	40J	60J	80J	100J
派工法	機台加工效率協調之 CV 值				
GA-EDD	2.077	1.615	1.784	1.580	1.383
GA-SPT	1.662	1.835	2.121	1.876	1.979
GA-LSF	1.867	1.910	1.528	1.610	1.491
GA-GA	1.033	0.943	0.939	0.899	0.876
GA-Comb	1.085	1.103	1.089	1.178	0.997

	20J	40J	60J	80J	100J
派工法	機台加工效率不協調之 CV 值				
GA-EDD	2.486	1.719	1.865	1.648	1.428
GA-SPT	1.720	1.881	2.076	1.877	1.975
GA-LSF	1.520	1.295	1.352	1.167	1.140
GA-GA	1.187	0.991	0.970	0.920	0.892
GA-Comb	1.145	1.088	1.069	1.103	1.035

表 4.14 情境四之派工法績效值(CV)

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-91.43%	-46.44%	-63.78%	-34.12%	-38.66%
GA-SPT	-53.14%	-66.37%	-94.74%	-59.16%	-98.46%
GA-LSF	-72.04%	-73.11%	-40.27%	-36.60%	-49.49%
GA-GA	4.84%	14.49%	13.82%	23.74%	12.20%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

	20J	40J	60J	80J	100J
派工法	相較於 GA-Comb 之改善率				
GA-EDD	-117.04%	-57.97%	-74.36%	-53.89%	-37.96%
GA-SPT	-50.19%	-72.82%	-94.15%	-75.33%	-90.90%
GA-LSF	-32.66%	-19.00%	-26.41%	-8.98%	-10.12%
GA-GA	-3.61%	8.96%	9.26%	14.07%	13.79%
GA-Comb	0.00%	0.00%	0.00%	0.00%	0.00%

表 4.15 情境四之派工法改善率

	(EDD, SPT, LSF)
20J	(0.6069, 0.3733, 0.0198)
40J	(0.3456, 0.3651, 0.2893)
60J	(0.4293, 0.4293, 0.1414)
80J	(0.3335, 0.3869, 0.2796)
100J	(0.3117, 0.3646, 0.3237)

(a) 機台加工效率協調

	(EDD, SPT, LSF)
20J	(0.6069, 0.3733, 0.0198)
40J	(0.3456, 0.3651, 0.2893)
60J	(0.4293, 0.4293, 0.1414)
80J	(0.3335, 0.3869, 0.2796)
100J	(0.3117, 0.3646, 0.3237)

(b) 機台加工效率不協調

表 4.16 情境四之最佳組合權重 (a) 效率協調 (b) 效率不協調

適合組合式派工法的情境，指的是三種單一派工法之績效相差不多，進行混合實驗以及反應曲面最佳化後的最佳權重，並不會像前三種情境一樣，都偏向某種派工權重最高(如表 4.16)，而是三者相互的組合(表 4.14 中的 GA-Comb)才能讓績效最佳。

所以由上可知，在此情境下，其他三種單一派工法皆不適用，而 GA-Comb 與 GA-GA 卻都可以得到不錯之績效值。若組合派工法之最佳權重並沒有何者較高之傾向時，只要使用最佳組合權重所組成之新的派工規則，便可求出相較於單一派工法為佳的解。

4.3.5 小結

綜合上述的分析，整理如下：

情境一：<1:1> GA-GA→GA-EDD→GA-Comb→GA-SPTG→A-LSF

<1:3> GA-EDD→GA-Comb→GA-GA→GA-SPT、GA-LSF

情境二：<1:1> GA-Comb、GA-SPT、GA-GA→GA-EDD→GA-LSF

<1:3> GA-Comb、GA-SPT→GA-GA→GA-EDD→GA-LSF

情境三：<1:1> GA-Comb、GA-LSF、GA-GA→GA-EDD→GA-SPT

<1:3> GA-Comb、GA-LSF→GA-GA→GA-EDD→GA-SPT

情境四：<1:1> GA-GA→GA-Comb→GA-SPT→GA-LSF→GA-EDD

<1:3> GA-GA→GA-Comb→GA-LSF→GA-EDD→GA-SPT

由上面的整理，可以發現在四種情境下，GA-Comb 皆可以求出與各情境適用之單一派工法較為接近之績效值，若 GA-Comb 之最佳權重落於某個單一派工法最高時，且三單一派工法間權重差異很大時，便可直接建議使用該派工法進行派工。

以下針對上述之結果，得到以下結論：

1. 目前沒有一個派工法在所有情境下全贏。

2. 組合派工法(GA-Comb)在四種情境下，皆能透過最佳權重的組合來建議最適合的派工方法。

3. GA-GA 自然演化的方法，在加工效率不協調的情境下，績效值會明顯變差。

以上驗證了一開始的第一項目的一組合派工法相較於其他派工法，可提供最穩健且較佳的解。針對第二項途程決策方面的結論，將透過下一節的分析來進行驗證。

4.4 加工途程分析

一、實驗目的

驗證本研究之基因演算法可提供較合適的加工途程決策。

二、實驗數據與分析

在加工途程方面，第三章研究方法中有提到，在基因演算法進行演化時，同一條染色體會同時求解跨廠與不跨廠解，並將最佳的解當作本次的績效值，所以為了驗證基因演算法是否會提供較合適的排程決策，將針對前一節所提到的四種情境與兩種加工效率進行跨廠與不跨廠數據分析。

除了驗證第二項目的外，由前一節得知 GA-Comb 與 GA-GA 在四種情境下，相同機台加工效率皆有不錯的解，但加工效率不同時，GA-GA 的績效會何會變差。針對這點，本研究也想透過以下實驗分析造成此種結果的原因所在。

所以，以下之實驗特別將 GA-Comb 與 GA-GA 兩者拿來做進一步的探討。

	20J	40J	60J	80J	100J
情境	跨廠相較於不跨廠之改善率				
情境一	4.64%	35.71%	0.84%	1.41%	1.26%
情境二	0.00%	0.00%	0.02%	0.02%	0.01%
情境三	9.25%	12.43%	9.84%	5.35%	6.81%
情境四	32.95%	55.80%	68.96%	54.26%	44.16%

表 4.17 GA-Comb 在加工效率協調之情境

	20J	40J	60J	80J	100J
情境	跨廠相較於不跨廠之改善率				
情境一	120.80%	89.10%	253.38%	114.16%	124.60%
情境二	1.43%	0.67%	0.43%	0.29%	0.24%
情境三	88.66%	70.61%	78.38%	80.19%	83.99%
情境四	41.25%	77.93%	83.02%	55.77%	84.63%

表 4.18 GA-Comb 在加工效率不協調之情境

	20J	40J	60J	80J	100J
情境	跨廠相較於跨廠之改善率				
情境一	-247.64%	-4.21%	-13.96%	-0.46%	0.00%
情境二	-91.58%	10770.35%	-26.86%	-15.13%	-12.58%
情境三	-312.38%	-187.85%	-340.50%	-546.04%	-473.07%
情境四	-784.96%	-2404.19%	-283.51%	-182.94%	-13503.90%

表 4.19 GA-GA 在加工效率協調之情境

	20J	40J	60J	80J	100J
情境	跨廠相較於不跨廠之改善率				
情境一	147.02%	203.17%	111.62%	155.37%	411.27%
情境二	526.23%	554.98%	582.40%	690.31%	153.40%
情境三	-242.23%	-153.36%	-209.07%	-227.59%	-260.40%
情境四	267.88%	345.97%	361.76%	395.05%	-150.64%

表 4.20 GA-GA 在加工效率不協調之情境

當機台加工效率協調時，代表兩廠機台加工速度一致，所以工件不管是在自己廠加工或跨廠加工，加工時間都是一樣的，由於跨廠需要計算跨廠時間，所以通常會選擇不跨廠；當加工效率不協調時，兩廠機台加工速度不一致下，某一廠比較慢做完時，可能造成工件延誤交期，所以通常會建議工件進行跨廠。

表 4.17 到表 4.20 整理出 GA-Comb 與 GA-GA 在機台加工效率協調與不協調情況下之跨廠改善率，所謂的跨廠改善率為跨廠績效值減去不跨廠績效值後，再除以不跨廠績效值，也就是跨廠績效相較於不跨廠績效贏的程度。由表中很明顯的可以發現，不管是

GA-Comb 或是 GA-GA 之改善率在機台加工效率不協調下，都比加工效率協調時為佳，此結果符合先前所說的，加工效率不協調時較適合於跨廠。所以驗證了實驗二的目的——本研究之基因演算法可提供合適的加工途程決策。

從數值來看，可以發現 GA-GA 在加工效率協調時，跨廠改善率都是負值，也就是說，在這種情境下，GA-GA 都會選擇不跨廠；而 GA-Comb 則較偏向於跨廠。由前一節的結果來看，在加工效率協調時，兩者績效值差不多，也就是說，當機台加工效率協調時，跨廠與不跨廠兩種方法之績效值並無顯著差異。

在加工效率不協調時，GA-GA 與 GA-Comb 之結果幾乎都偏向於跨廠，但 GA-GA 之績效值卻明顯變差，由此可以發現 GA-GA 方法似乎較適用於不跨廠的情境。

由 GA-GA 的本質來看，由於 GA-GA 是經由基因演算法進行途程決策後，排序的部份則是交由 GA 自然演化，所以只有進行分廠的動作，如果在適合跨廠的情境下，GA 會強制讓加工途程分為跨廠解，這樣的缺點會讓原本最後一個加工之工件，在下一站被分到第一個加工之機率提高。如圖 4.1，此圖為 GA-GA 三站的分廠方式，當第一站與第二站分廠位置不同時，原本工件 1 在第一站是 A 廠最後一個加工，到了第二站卻變成 B 廠第一個加工，因為不會像 EDD、SPT 等單一派工法進行重排，所以 GA-GA 在適合跨廠的情境下，績效才會變的較差。

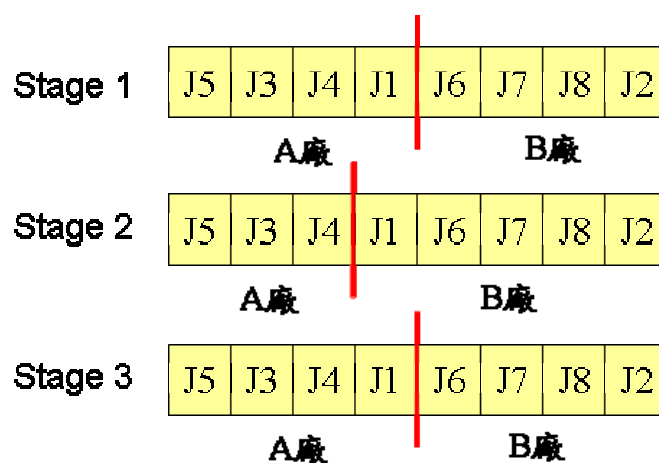


圖 4.1 GA-GA 分廠示意圖

三、小結

由本節的實驗分析，可以得到以下結論：

1. 驗證本研究之基因演算法可提供較合適的加工途程決策。
2. GA-GA 適用於不跨廠之情境。



第五章 結論與未來建議

5.1 結論

如上所述，本研究最初始產生的染色體為隨機排列，並無任何特殊意義，但是經由染色體分別作加工途程與加工順序的解讀後，同一條染色體卻能含有許多不同的意義。

以下小結本研究的所有結論：

1. 目前沒有一個派工法在所有情境下能全贏。
 - 會贏的情況是由於原始數據 Data Dependent 的關係。
2. 在所有派工法中，組合派工法可以在各種情境下，得到最穩健且較佳的結果。
3. GA-GA 適用於不跨廠的情境。
 - GA-GA 適用情境下，他的貢獻是只要做分廠，便可透過基因演算法自然演化，可搜尋的解空間更大，較能得到比排序後更佳的解。
4. GA-GA 方法，不適用於跨廠情境。
 - 越容易跨廠時，GA-GA 會變差。因為 GA-GA 只做分廠，不做排序，會造成第一站最後一個加工工件，第二站卻跑到第一個，要分析跨廠解會變差。
5. 機台效率越不協調，越容易跨廠。
 - 工件傾向往效率高之機台加工，造成機台效率不協調情境下，跨廠解績效比不跨廠佳。
6. 可同時求解加工途程與加工順序兩項決策，並為工廠提供最合適的排程建議。
 - 加工途程建議：可以分析此工廠的生產型態是否適合跨廠生產，若跨廠績效值優於不跨廠，則建議進行跨廠；反之則不跨廠。
 - 加工順序建議：可以決定出此工廠所有待加工工件的加工順序。只要透過混合實驗與反應曲面法求取權重值，由此權重值的特性，便可判斷工廠是適合於單一派工法或是組合式的派工法。
7. 本研究具擴充性。
 - 本研究的方法可擴充至多廠、多站，甚至整條供應鏈。

- 全部只需一條染色體，當規模變大時，運算時間比較不是問題。
8. 補足過去文獻不足處。
- 過去跨廠文獻僅針對 Route 做研究，本研究還特別針對 Sequencing 做深入探討。
 - 過去多廠大多做多階段生產規劃，但多階段規劃必須面臨每階段的目標式決定問題，無法同時考量同一目標值，而本研究可同時考量同一目標來進行生產規劃。

5.2 未來建議

經過上述的實驗研究之後，後續可以在進行延伸的方向有以下幾點：

1. 組合派工法進行其他單一派工法之組合，探討結果是否會相同。
2. 針對多廠或是供應鍊進行更詳細的分析。
3. 與過去多階段求解進行績效比較，評比何者績效結果較佳。
4. 將當機、良率等實際問題納入排程考量。



參考文獻

英文部分

- Adiri I and Pohoryles D (1982). Flow-shop/no-idle or no-wait scheduling to minimize the sum of completion times. *Naval Research Quarterly* **29**: 495-504.
- Allahverdi A, Guptab J ND and Aldowaisan T (1999). A review of scheduling research involving setup considerations. *OMEGA* **27**: 219-239.
- Allahverdi A , Ng CT, Cheng TCE and Kovalyov MY (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* **187**: 985–1032.
- Caroloin T, Pascal B, Dinitri G. and Gernar B (1995). Multi-site planning: non flexible production units and set-up time treatment. *Proceedings of the Emerging Technologies and Factory Automation* **3**.
- Chen JC, Fan YC, Wang JY, Lin TK, Leea SH, Wu SC and Lan YJ (1999). Capacity Planning for a Twin Fab, Semiconductor Manufacturing Conference Proceedings. *IEEE International Symposium* 317-320.
- Chen JC, Chen CW, Lin CJ and Rau H (2005). Capacity planning with capability for multiple semiconductor manufacturing fabs. *Computers & Industrial Engineering* **48**: 709–732.
- Cornell JA (1990). Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data. 2nd edition, Wiley, New York.
- Dabbas RM and Fowler JW (2003). A New Scheduling Approach Using Combined Dispatching Criteria in Wafer Fabs. *IEEE Transactions on Semiconductor Manufacturing* **16**.
- Frederix F (1996). Planning and Scheduling Multi-Site Semiconductor Production Chain: A survey of needs, current practices and integration issues. Conference on IT and Manufacturing Partnerships: Delivering the Promise, Galway, Ireland, 107-116.

- Garey MR and Johnson DS (1979). Computers and intractability: A guide to the theory of NP-completeness. New York:Freeman.
- Guinet A (2001). Multi-site planning: A transshipment problem. *International Journal of Production Economics* **74**: 21-32.
- Holland JH (1975). Adaptation in Neural and Artificial Systems. Ann Arbor, MI: Univ. Michigan Press.
- Johnson SM (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* **1**: 61–67.
- Lin JT, Wang FK and Kuo PC (2005). A parameterized-dispatching rule for a Logic IC sort in a wafer fabrication. *Production Planning & Control* **16**:426-436.
- Linn R and Zhang W (1999). Hybrid Flow Shop scheduling: A survey. *Computers and Industrial Engineering* **37**: 57–61.
- Sambasivan M and Yahya S (2005). A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research* **32**: 537–555.
- Sauer J, Suelmann G. and Appelrath HJ (1998). Multi-site scheduling with fuzzy concepts, *International Journal of Approximate Reasoning* **19**: 145-160.
- Sauer J and Appelrath HJ (2000). Integrating transportation in a multi-site scheduling environment. Proceedings of the IEEE 33rd Annual Hawaii International Conference on System Sciences, 2.
- Thierry C, Besnard P, Ghattas D and Bel G. (1995). Multi-Site Planning:Non Flexible Production Units and Set-Up Time Treatment. INRIA/IEEE Symposium on Emerging Technologies and Factory Automation.ETFA '95, Proceedings **3**: 261-269.
- Timpe CH and Kallrath J (2000). Optimal planning in large multi-site production networks. *European Journal of Operational Research* **126**: 422-435.

- Toba H, Izumi H, Hatada H and Chikushima T (2005). Dynamic load balancing among multiple fabrication lines through estimation of minimum inter-operation time. *IEEE Transactions on Semiconductor Manufacturing* **18**: 202-213.
- Vercellis C (1999). Multi-plant production planning incapacitated self-configuring two-stage serial systems. *European Journal of Operational Research* **119**: 451-460.
- Wu MC and Chang WJ (2007). A short-term capacity trading method for semiconductor fabs with partnership. *Expert Systems with Applications* **33**: 476-483.
- Wu MC, Chen CF and Shih CF (2008). Route Planning for Two Wafer Fabs with Capacity-Sharing Mechanisms. *International Journal of Production Research*. (DOI: 10.1080/00207540802172029)

中文部份

- 郭乃蓁，「晶圓製造廠多廠間」，國立交通大學，碩士論文，民國 89 年。
- 張美滿，「多廠區跨廠產能規劃之探討」，元智大學，碩士論文，民國 91 年。
- 陳建銘，「多廠區緊急訂單承接評估及跨廠產能規劃探討」，元智大學，碩士論文，民國 92 年。
- 劉謹銘，「以組合派工法求解雙流線型工廠排程」，國立交通大學，碩士論文，民國 98 年。