

東 海 大 學  
工業工程與經營資訊研究所

碩士論文

多廠整合型生產指派與排程系統之最佳化

Optimization of Multi-Site Integrated Production

Assignment and Scheduling System

研 究 生：黃俊嘉

指 導 教 授：張炳騰 博士

彭 泉 博士

中 華 民 國 九 十 四 年 六 月

# **Optimization of Multi-Site Integrated Production Assignment and Scheduling System**

By

Chun-Chia Huang

Advisor : Dr. Ping Teng Chang

Dr. Chyuan Peng

A Thesis

Submitted to the Institute of Industrial Engineering &

Enterprise Information

at Tunghai University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering & Enterprise Information

June 2005

Taichung, Taiwan, Republic of China

# 多廠整合型生產指派與排程系統之最佳化

研究生：黃俊嘉

指導教授：張炳騰 博士

彭 泉 博士

東海大學工業工程與經營資訊研究所

## 摘要

近年來隨著經濟環境與產業結構的快速變遷，市場上對於產品需求的大增，因此企業面臨訂單資源的不斷湧入，為了增加產能，企業不斷地擴廠或是合併其他產能，也使得原本簡單的訂單管理問題，隨著工廠的增加、地域的拉遠，從單廠衍生到複雜的多廠規劃(Multi-Site Planning)問題上。

在本研究中，將針對多製造廠在全面訂單管理的情況下，探討在一健全的訂單管理制度下，對於製造廠的跨廠訂單分配模式進行研究，建立一個考慮到多製造廠在接收大筆訂單時，如何依據產品的市場銷售特性、訂單交期、廠與廠間的生產排程、訂單利潤及廠的產能負荷度等等交叉複雜的關係，建立一多廠整合型生產指派與排程系統的決策模式。

而現實世界的排程問題並非單一目標可以滿足，需以多目標(multi-objective)的觀點來考慮，而且目標之間每每是彼此衝突。而目前多數關於多目標排程的研究中，僅考慮生產製造方面的績效因子，這些因子多是屬於量性因素(quantitative factors)。事實上，排程環境中所需考量的還需包含有一些關於組織營運策略上的質性因素(qualitative factors)。

本研究以混合式遺傳演算法(Hybrid-GA, HGA)求算訂單指派至各廠的生產排程與各廠的產能平衡，以決定訂單的最佳配置。進而便能依據所指派的結果作為各家製造廠進行生產計畫與排程規劃的依據。整體來說，本研究擬提出一套多廠整合型生產指派與排程系統，最後透過 Job shop 排程實證問題，驗證本研究所提方法與求解模式的可行性及分析衡量此系統帶來的效益。

**關鍵字：**多廠區、遺傳演算法、禁忌搜尋法、多目標生產規劃與排程

# Optimization of Multi-Site Integrated Production Assignment and Scheduling System

Student : Chun-Chia Huang

Advisor : Dr. Ping-Teng Chang

Dr. Chyuan Peng

Institute of Industrial Engineering & Enterprise Information  
Tunghai University

## Abstract

As economic environment and industrial structure changing rapidly, product demand frequency increases immensely as well. To accomplish the great amount of orders, industries expand their plants accordingly or integrate with others to increase their outputs. As a result, conventional order management is no longer sufficient to solve these problems, accompanied with the distance and different information of these incorporated factories. The single-site planning has developed in to multi-site planning.

This research aims to construct the decision model of integrated multi-site production assignment and scheduling problem. To support the multi-site factories with their enormous orders, based on the premise that they are under total order management, the decision model considers such complicated factors as product market features, due date, production schedule, order profit and capacity load of each plant.

There is no more a singular objective in real world scheduling system, but multi objectives that are commonly conflicting to each other. In addition, the effect factors taken account by current multi-objective scheduling research are quantitative factors. Essentially there are more qualitative factors to be considered related to organizations' operating tactics.

Our research using hybrid-genetic algorithms method for production assignment's scheduling and each plant's capacity balancing to determine optimal order's allocation. Consequently, our research proposes an optimization of multi-site integrated production assignment and scheduling system.

**keywords : multi-site, genetic algorithms, tabu search, multi-objective production planning and scheduling**

## 誌謝

碩士論文完成的期間，經歷了許多的挫折與低潮時期。雖然一路走來很辛苦，卻也讓我成長不少。回想這兩年來的研究路程，首先要感謝指導教授張炳騰博士，不僅在學業上細心教導，對於人生的諸多指引，更是讓我獲益非淺，由衷的感謝老師的教導。

在口試期間，承蒙彭泉博士、洪堯勳博士、白炳豐博士、陳琨太博士特別撥空審閱論文，提供寶貴意見與指導，使得論文內容更為完備，也體會到學海無涯的道理，在此致上最深的謝意。

在這兩年的研究過程中，感謝宏武、國禎、國平、志昇學長在課業及生活上的幫忙，同學小強、敬淳、文偉、瑋珊的相互扶持，一起走過最艱難的時光，此外亦感謝研究室學弟妹一龍廷、禎祥、國丞、鼎翰及香君的支持與鼓勵，讓我在低潮的時候依然能擁有歡笑。

最後特別要感謝父母及家人的關懷與照顧使我能在無憂無慮的環境下，順利完成學業。

感謝所有曾經關心及幫助我的人，謝謝。

黃俊嘉 謹誌於

東海大學工業工程與經營資訊研究所

民國九十四年六月

# 目錄

摘要.....	III
ABSTRACT .....	IV
誌謝.....	V
目錄.....	VI
表目錄.....	VIII
圖目錄.....	X
第一章 緒論 .....	1
1.1 研究背景與動機 .....	1
1.2 研究方向與目的 .....	1
1.3 研究範圍.....	2
1.4 研究方法與步驟 .....	3
1.5 論文架構.....	4
第二章 文獻探討 .....	5
2.1 多廠規劃.....	5
2.1.1 多廠規劃問題之描述 .....	5
2.1.2 多廠規劃應用之文獻 .....	5
2.1.3 商用軟體做法與架構 .....	6
2.2 排程問題之描述 .....	10
2.2.1 排程問題之分類.....	11
2.2.2 多目標排程.....	13
2.3 遺傳演算法 .....	17
2.3.1 常見 Job shop 排程問題的編碼表示法 .....	18
2.3.2 常見排序問題之運算子 .....	22
2.3.3 排程中遺傳演算法之應用.....	25
2.4 禁忌搜尋法 .....	27
2.4.1 禁忌搜尋簡介.....	27
2.4.2 禁忌搜尋在排程上之應用.....	28
2.5 本章小結 .....	28
第三章 多廠整合型生產指派與排程系統之架構設計 .....	29
3.1 系統架構原理 .....	29

3.2 多廠生產指派與排程模式架構.....	32
3.3 多廠機器排程法則.....	35
3.3.1 排程演算法.....	35
3.3.2 作業時間修正模式.....	39
3.4 定性因素模式架構.....	42
3.4.1 AHP 法的評估計算.....	43
3.5 定量因素模式架構.....	45
3.5.1 製距績效評估.....	45
3.5.2 交期滿足度評估.....	46
3.5.3 機器使用率評估.....	46
3.6 遺傳演算法之運作.....	47
3.6.1 編碼表示法.....	47
3.6.2 初始族群的產生.....	47
3.6.3 適應性函數之設計.....	49
3.6.4 交配運算子設計.....	50
3.6.5 突變運算子設定.....	51
3.6.6 育種選擇.....	53
3.7 混合式遺傳演算法(HYBRID GENETIC ALGORITHM)之設計.....	54
第四章 系統實證.....	57
4.1 相關基本資料之設立.....	57
4.2 定性資料評估.....	57
4.3 基本參數設定.....	59
4.4 傳統遺傳演算法(GA)、混合式遺傳演算法(HGA)比較.....	63
4.5 多廠區最適排程解.....	68
4.6 實驗設計.....	71
第五章 結論與未來研究方向.....	76
5.1 研究總結.....	76
5.2 未來研究方向.....	76
參考文獻.....	77

## 表目錄

表 2.1 排程績效目標.....	12
表 2.2 多目標排程文獻整理.....	14
表 2.3 遺傳演算法之優缺點歸納表.....	26
表 3.1 定性與定量因素表.....	30
表 3.2 AHP 評估尺度.....	43
表 3.3 各因素之權重制訂.....	44
表 3.4 各方案之評估值.....	44
表 3.5 各方案之加權排序.....	44
表 3.6 訂單平均基因順序.....	50
表 3.7 懲罰函數之計算.....	50
表 3.8 各訂單作業之基因.....	55
表 3.9 $S_1$ 的鄰域.....	55
表 3.10 $S_2$ 的鄰域.....	55
表 3.11 $S_3$ 的鄰域.....	56
表 3.12 $S_4$ 的鄰域.....	56
表 4.1 訂單資料.....	58
表 4.2 定性與定量之權重值.....	58
表 4.3 定性因素之權重值.....	58
表 4.4 定量因素之權重值.....	59
表 4.5 遺傳演算法中參數之設定值.....	59
表 4.6 混合式遺傳演算法中參數之設定值.....	59
表 4.7 定性因素-各訂單利潤之評估值.....	60
表 4.8 定性因素-各訂單顧客歷史交易之評估值.....	60
表 4.9 定性因素-各訂單市場考量之評估值.....	61
表 4.10 定性因素-各訂單顧客潛在訂單之評估值.....	61
表 4.11 各訂單定性因素之評估值.....	62
表 4.12 機台使用率(GA).....	63
表 4.13 機台使用率(HGA).....	63
表 4.14 零件訂單交期滿足度(GA).....	63
表 4.15 零件訂單交期滿足度(HGA).....	63
表 4.16 最適排程結果(GA).....	64
表 4.17 最適排程結果(HGA).....	65
表 4.18 各零件訂單指派順序(GA).....	66
表 4.19 各零件訂單指派順序(HGA).....	67
表 4.20 10 張零件訂單(以相同運算代數比較).....	71
表 4.21 20 張零件訂單(以相同運算代數比較).....	71
表 4.22 30 張零件訂單(以相同運算代數比較).....	72
表 4.23 10 張零件訂單(以相同運算搜尋時間比較).....	73



表 4.24 20 張零件訂單(以相同運算搜尋時間比較).....	73
表 4.25 30 張零件訂單(以相同運算搜尋時間比較).....	74

## 圖目錄

圖 1.1 論文進行流程圖 .....	4
圖 2.1 MANUGISTICS CBMP 的資料結構之概念圖 .....	7
圖 2.2 ADEXA SCP 的資料結構之概念圖 .....	8
圖 2.3 ADEXA SCP 啟發式法則的產銷平衡規劃 .....	9
圖 2.4 ADEXA SCP 之網路排程的排程邏輯 .....	10
圖 2.5 多目標排程文獻矩陣示意圖 .....	16
圖 2.6 一般遺傳演算法運作流程 .....	18
圖 2.7 分支圖 .....	21
圖 3.1 多廠區零件訂單分配示意圖 .....	29
圖 3.2 系統架構圖 .....	31
圖 3.3 多廠零工式生產系統示意圖 .....	35
圖 3.4 作業的排程流程圖 .....	38
圖 3.5 原始排程甘特圖 .....	39
圖 3.6 作業的時間修正流程圖 .....	41
圖 3.7 修正排程甘特圖 .....	42
圖 3.8 交期之模糊隸屬函數 .....	46
圖 3.9 編碼表示法示意圖 .....	47
圖 3.10 上半部基因初始族群示意圖 .....	48
圖 3.11 下半部基因初始族群示意圖 .....	48
圖 3.12 交配示意圖 .....	52
圖 3.13 突變示意圖 .....	52
圖 3.14 混合式 GA 運作流程圖 .....	56
圖 4.1 廠一, 二, 三整體最適適應函數值趨勢圖 .....	63
圖 4.2 廠一最適排程(GA) .....	68
圖 4.3 廠二最適排程(GA) .....	68
圖 4.4 廠三最適排程(GA) .....	69
圖 4.5 廠一最適排程(HGA) .....	69
圖 4.6 廠二最適排程(HGA) .....	70
圖 4.7 廠三最適排程(HGA) .....	70
圖 4.8 不同零件訂單數目效能比較(以相同運算代數比較) .....	74
圖 4.9 不同零件訂單數目效能比較(以相同運算搜尋時間比較) .....	75

# 第一章 緒論

## 1.1 研究背景與動機

經營企業的目的不外乎追求最大的利潤及最小的成本付出。而企業型態又依其特性分為服務業與製造業。所謂服務業是以付出勞務提供令顧客滿意的服務以獲取利潤，而製造業則是以生產製造有形之產品交付顧客使用以獲取報酬。

在製造業中，產品的生產數量往往取決於顧客之需要。市場需求旺盛時，會有需求大於供給的情況。結果會造成工廠面臨產能無法負荷市場需求的情形。在這種狀況下，企業若無法在數量與交期上滿足市場需求時，會造成訂單的流失與錯失獲利的契機。因此，企業會考慮以增建新工廠、擴充生產線或是將部份訂單外包給其他企業的措施，以避免因本身的產能不足而流失訂單。

而為了因應目前產業間，多廠生產方式的發展趨勢，許多大廠因而提出一多廠間的生產規劃模式，主要以企業整體的角度進行資源的規劃、訂單承接與分配。在單一廠的訂單管理問題中，考量的為廠內的產能是否能滿足訂單的要求，但是擴及多廠的訂單指派時，雖大多採全面的訂單管理系統，但是將面對的不僅是單廠的問題，也需考量廠與廠之間的產能平衡、產品組合、各廠生產之特性、各廠製造成本等複雜性。企業在成立之初，可能因本身資金不足或是市場需求不大，而只有單一間生產工廠。但是，隨者市場上產品需求量的增加，造成生產日益繁重，企業在滿足顧客需求以賺取更多利潤的誘因之下，會將產能擴充，如擴建生產線或是工廠以滿足市場需求。因此，分配顧客訂單給各個工廠的問題，就顯得格外重要，如何將顧客所下的訂單，依照各工廠不同的產能、不同的產品生產種類，做出整體的規劃，如期的完成交予顧客，是企業關注的焦點。本研究研擬解決此跨廠區的訂單分配問題並提出適用的訂單指派與生產排程模式，提供企業解決之道。

## 1.2 研究方向與目的

由於企業因應訂單的不斷湧入，在既有的工廠中，會因為產能不足，無法承接訂單的所有訂購數量。在考量整體利益後，必然會籌措新資源(增廠或外包)以擴大產能，提昇企業承接單的能力，以免錯失訂單承接的機

會，影響企業的永續經營目標與利益。

因此，在本研究中，將針對多製造廠在整合型的全面訂單管理情況下，探討在一健全的訂單管理制度下，對於製造廠的跨廠訂單分配模式進行研究，建立一個考慮到多製造廠在接收大筆訂單時，如何依據產品的市場銷售特性、訂單交期、廠與廠間的生產排程、訂單利潤及廠的產能負荷度等等交叉複雜的關係，進而建立一多廠整合型生產指派與排程系統的決策模式。希望能協助生產管理者快速地做出訂單指派與生產排程的正確決策。

在此多廠訂單分配模式下，企業製造的產品，可在多個工廠中生產。換言之，在本研究中其目的在於衡量將  $n$  筆零件訂單指派到  $m$  家製造廠中，建立一訂單指派模式，之後以混合式遺傳演算法求算訂單指派至各廠的生產排程與各廠的產能平衡及最大化訂單交期滿足率為目的，決定訂單的最佳配置。進而便能依據所指派的結果作為各家製造廠進行生產計畫與排程規劃的依據。各製造廠便可依此資訊安排或修正生產規劃與訂單的配置。

### 1.3 研究範圍

在本研究中是以擁有多家製造廠，分佈於各地方的總公司來檢視整個訂單指派與生產排程規劃，所以在建構中是以總公司所能掌握的資訊為主，其中所涵蓋的資訊包括：

- (1) 總公司的資訊：訂單的數量、產品價格、交期等資訊。
- (2) 製造廠的資訊：如產能負荷、製程技術、品質等資訊。

對於總公司的假設為接收訂單的唯一窗口，並不具有製造能力，而在將訂單統一指派到製造廠時，由製造廠生產並完成出貨。其實，在總公司所能掌握的資訊之外，私底下各製造廠間也可能存在著其關連的外包網路，如指派給 A 製造廠的訂單，為了將產能用於製造對其更有利的後接訂單會將前接訂單轉給 B 製造廠，或是外包給生產體系外的外包工廠。

在本研究中所提之多廠間的生產排程是以零工式(Job shop)生產類型的排程問題做為研究範圍，在一般的排程問題中以零工式類型最為複雜。傳統常見的流程式(Flow shop)生產類型因為市場競爭激烈與生產技術的進步，造成產品生命週期的縮短與產品變異性的提高，導致產品愈趨向於多樣化，流程式的大批量生產方式已經無法滿足市場的多樣化需求，多樣少

量的製造方式已經是一個無法避免的趨勢，亦逐漸取代了傳統的大量生產方式，零工式生產方式最大的優點即是依照顧客所需產品之加工途程來進行安排，也就是說，零工式生產可以提供整個製造環境應對市場變化的彈性。因此，本研究以零工式生產做為本研究在製造廠中的生產方式。

在限制條件方面，因為本系統是在多製造廠間的規劃下進行訂單指派與廠內的生產預排。所以系統所存在的限制條件很多，亦對系統做了一些假設，整體而言包括：

1. 訂單交期的限制
2. 不考慮訂單預測性問題
3. 不考慮外包
4. 各廠內的生產排程方式與控制系統採用相同方式
5. 不考慮任何非預期狀況及機台當機之事件
6. 生產途程的限制
7. 廠內機台的限制
8. 不允許緊急插單及抽單
9. 不考慮整備成本及其他成本因素的考量

等都是系統中的限制條件及假設。

除此之外，關於排程問題的求解方法方面，隨著電腦計算能力的增強，許多的求解方法如搜尋演算法、人工智慧的工具都已發展的相當成熟，然若要將其應用於實務界，始終會存有障礙，排程問題尤其是如此，有關排程的研究為數眾多，但是在一般工廠中，仍多以簡單的派工法則做為排程規劃的依據。所以本研究希提出結合基因傳演算法與禁忌搜尋法概念的運算模式，期發展一能應用於實務界的訂單指派與生產排程系統的雛形。

## 1.4 研究方法與步驟

本研究主要是透過電腦實驗的方式進行研究，實驗的對象為一假設的多廠 Job shop 排程問題，以此排程問題對本研究所提的遺傳演算法與混合式遺傳演算法在多目標衡量下分為兩部分各進行實驗，最後再依據實驗數據與結果作分析討論。本研究主要可分為三個步驟：

1. 多目標排程架構之設計

確認排程環境中的多重目標因素，並將其區分為定性與定量二種類型。利用 AHP 法評估定性因素，同時制定量性與質性因素的權重。依據選擇的因素設計演算過程中之多目標函數。建構以遺傳演算法與混合式遺傳演算法為基礎的多目標排程系統。

## 2. 多階多廠機器模式之建立

在此部分中，對加工作業採取批量分割，並局部調整子批量加工時間，提出多階多廠機器的作業排程演算法。

## 3. 電腦實驗

在此階段中，以步驟 1、2 中所設計的遺傳演算法與混合式遺傳演算法，進行模擬實驗，藉由實驗數據的分析以驗證本研究之系統架構。

# 1.5 論文架構

本研究論文的內容共分五章：第一章說明本研究內容的背景、動機、目的、範圍、方法與步驟等相關內容；第二章則針對本研究所涉及相關文獻加以探討，包括多廠規劃、階層分析比較法、遺傳演算法、零工式排程等相關文獻；第三章則根據本論文之目的與文獻探討所得之啟發，提出一多廠整合型生產指派與排程系統架構，詳細闡述系統架構中各項機制之功能設計與運作方式；第四章為論文中系統之實驗結果與分析；第五章則根據本研究所得之結果，說明研究結論與未來發展方向。本論文的進行流程如圖 1.1。

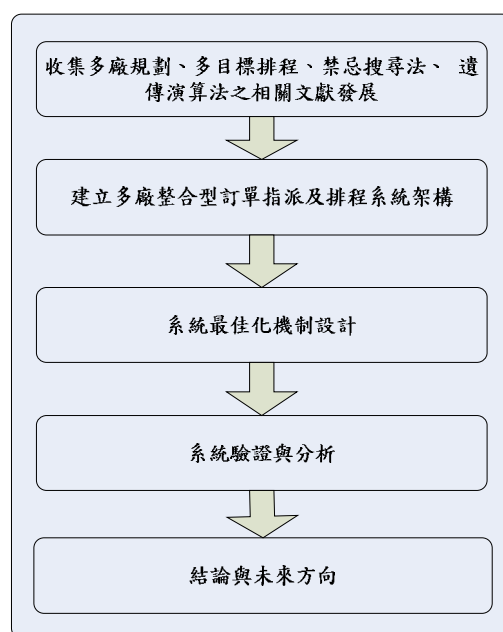


圖 1.1 論文進行流程圖

## 第二章 文獻探討

### 2.1 多廠規劃

#### 2.1.1 多廠規劃問題之描述

近年來隨著經濟環境與產業結構的改變，過去以單一製造工廠為基礎的生產規劃模式，已漸漸無法符合產業界的需求。企業藉由增廠或是整併其它工廠滿足龐大的訂單需求，因而一套良好的多廠生產規劃模式是不可或缺的因素。

#### 2.1.2 多廠規劃應用之文獻

Thierry, et al. (1995)[63]指出多廠區問題主要決定產品在不同廠區間的生產以及不同時段間的運送問題，作者並認為這是限制條件滿足(constraint satisfaction)問題，應同時考量並滿足先前關係、資源限制、交期以及運送限制(transportation constraint)。Bullinger et al. (1997)[10]提出以物件導向(object-oriented)模式規劃多廠區生產，由於以物件導向為方法，也易於廣泛的應用在不同領域。Sauer et al. (1998)[59]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)之需求量及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，它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。Pirkul and Jayaraman (1998)[55]提出分配網路(distribution network)策略性設計問題，並且探討多樣、多階運籌探討運送與配送的問題，作者以有限產能以及倉儲供應鏈規劃問題為例，提出一套以Lagrangian relaxation為啟發法則之整數規劃(mixed integer programming)模式。Rous et al. (1999) [58]提出一套整合性的模式，並應用在多廠區的排程以及規劃，主要是訂出一套可行解，包含各機台上批量的順序問題以及各廠間的排程先後次序問題。Vercellis (1999) [65]考量主生產規劃(master production planning, MPS)以及產能配置(capacity allocation)問題求解一多廠間生產問題。Timpe and Kalltrah (2000)[64]提出以時間為基礎之線性整數規劃模式，涵蓋相關整個多廠生產網路之供應鏈管理，作者將生產(production)，配送(distribution)，行銷(marketing)，工廠產地(production site)以及銷售點(sales point)納入考量。Guinet (2001) [27]提出以二階

(two-level)式生產管理手法管理多廠生產系統，分為全域(global)的多廠生產規劃以及區域(local)的多現場排程。Sambasivan and Schmidt (2002)[60]提出一啟發式法則解決多廠多階的廠內運送(inter-plant transfers)問題。Moon et al. (2002)[46]以多廠間整合型規劃排程(integrated process planning and scheduling)模式為概念，並考量替代機台選擇與順序，利用基因演算法(GA)求解最小化總延遲時間(total tardiness time)。Leung et al. (2003)[39]提出一多目標模式解決生產規劃問題，求解最大化訂單利潤，並加入生產懲罰(production penalties)變數，以及最小化人力(work force)變動。Gnoni et al. (2003)[25]對於多廠製造系統之批量(lot sizing)及排程問題，以產能限制、不確定多產品及多區間需求(multi-period demand)加以考量並求解。作者提出一混合模式(hybrid modeling)方法，並加入模擬方法(simulation model)求解。Levis and Papageorgiou (2004)[38]提出一系統化數學規劃模式，求解一製藥業之多廠(multi-site)產能規劃問題。Jolayemi and Olorunniwo (2004)[31]提出一確定性(deterministic)模式解決多廠(multi-site)與多倉儲間(multi-warehouse)之生產規劃與運送問題，並求解最大化利潤。

### 2.1.3 商用軟體做法與架構

以下我們彙整二種現有SCP軟體與模擬導向APS，探討一般商用軟體解決供應鏈規劃或多廠區生產規劃的架構與概念：

#### (1)Manugistics CBMP

CBMP是Manugistics的一個模組，其功能主要是產生滿足需求的生產計劃。它同步考量物料供給(material availability)、配送產能(distribution capacity)及生產產能(production capacity)等限制條件。它可以根據公司的企業策略(business strategy)設定限制條件，例如顧客訂單中計算安全存貨及消耗預測訂單的法則。主要考量的限制條件如下：物料供給(material availability)、產能(production capacity)、進料存貨能力(in handling capacity)、出貨存貨能力(out handling capacity)、與運輸通路產能(lane capacity)。CBMP 也提供即時可允諾交貨數量查詢(Real-Time Available-To-Bromise；RTATP+)的功能，其考量關鍵資源及物料的供給能力來決定品項(item)的達交數量，其結構如圖2.1：



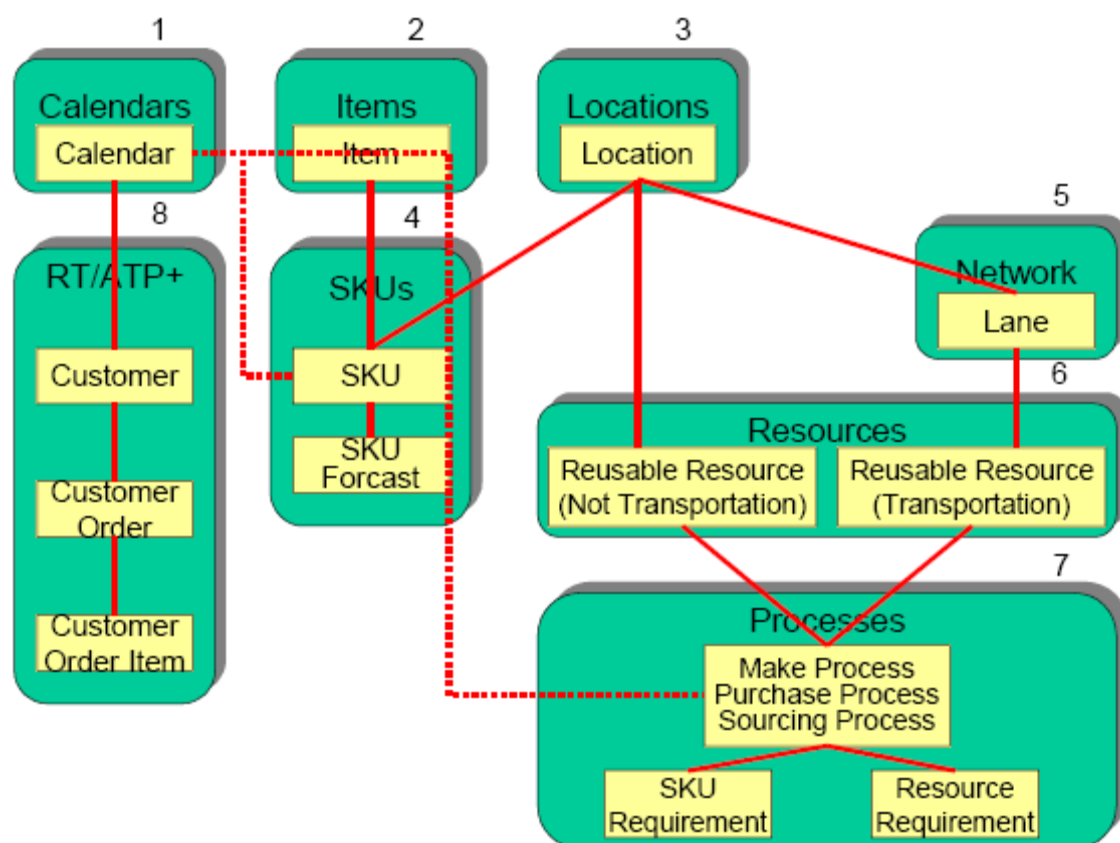


圖 2.1 Manugistics CBMP 的資料結構之概念圖

CBMP藉由設定優先度模式(priority scheme)來決定那個訂單能最先被滿足。其優先度模式主要考量以下因素的權重：收益、邊際利潤、品項、顧客、安全存貨、預測、及交期。而決定訂單的排序方式如下：

1. 現有品項需求中，優先度較高較優先處理；
2. 新的品項需求能以現有多筆品項需求同時滿足；
3. 根據顧客與品項之優先度的排序結果來重新規劃；
4. 如果優先度較高的訂單在重新規劃的過程中仍然延遲或未被滿足，則系統會發出例外訊息；
5. 避免優先度高的訂單佔用資源必須使用固定訂單的設定；

滿足需求的方式，首先以現有的存貨來滿足其儲存單位的需求。如果無法滿足的話，它將會指定製造、外包及採購流程來要求生產、運輸及購買該儲存單位。CBMP會優先考量優先度最高的流程，當流程能完全或部份滿足需求訂單(顧客訂單、儲存單位的預測訂單及安全存貨單)時，它將根據流程來建立供給訂單(製造訂單、外包訂單或採購訂單)。它會一直規劃至需求被滿足，或是該儲存單位的流程都被指定過了。在建立供給訂單的同

時，系統會要求設定物料的供給狀況及生產和運輸的產能狀況之限制條件。它也能設定預測調整及安全存貨計算的法則。

## (2)Adexa SCP

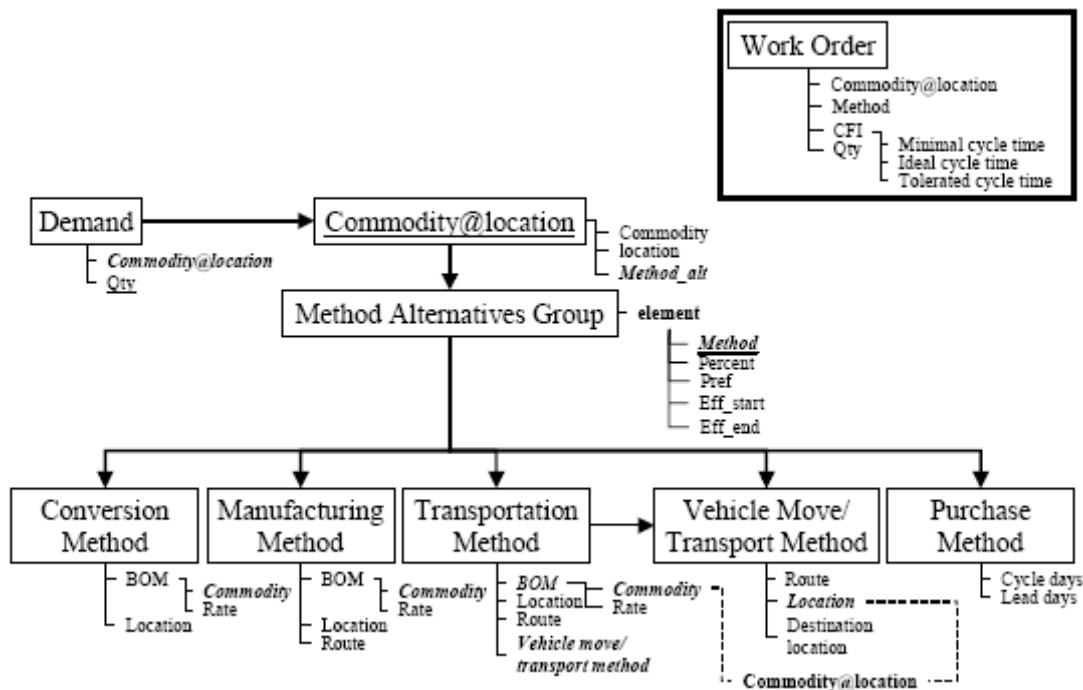


圖 2.2 Adexa SCP 的資料結構之概念圖

Adexa SCP的規劃方式分為兩大部分：(1)產銷平衡(balancing)規劃；(2)供應網路排程(scheduling)，其架構如圖2.2所示。其產銷平衡規劃使用兩種方法：

- 1.線性規劃：其中包含了Inner、CPLEX、XPRESS、External四種法則可供選擇。
- 2.啟發式法則：根據優先度(preference)及外包比率(outsourcing rate)，並考慮等級需求(grading requirement)及工單大小限制(work order sizing constraint)做規劃。

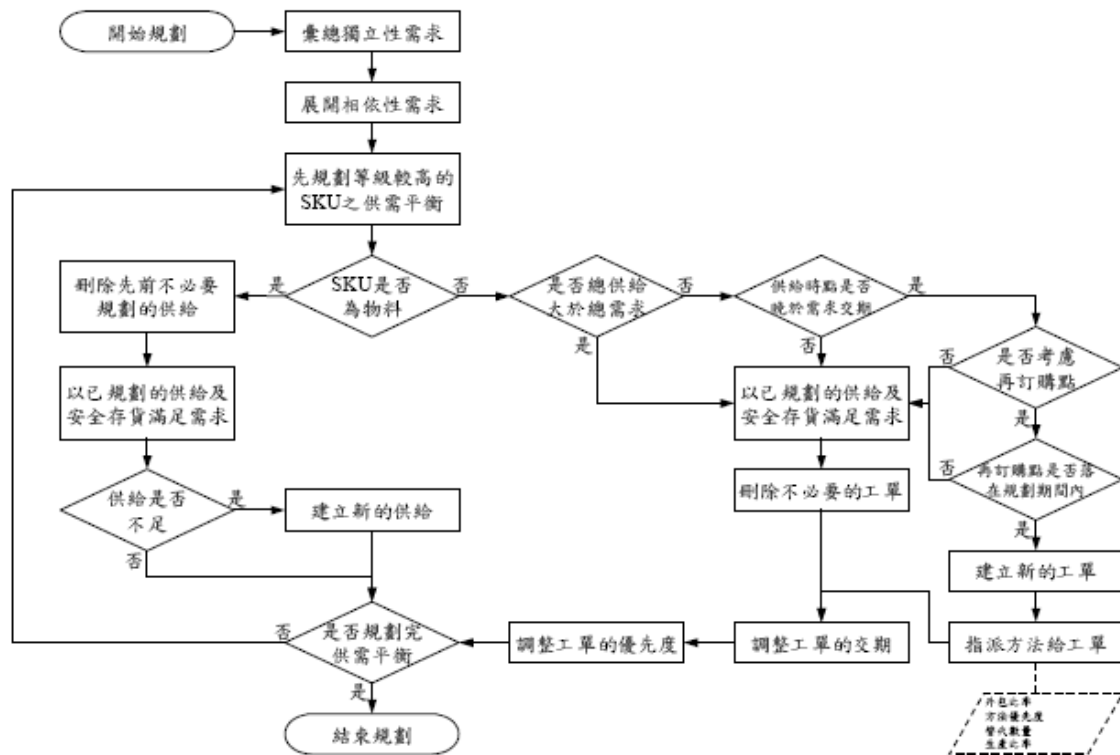


圖 2.3 Adexa SCP 啟發式法則的產銷平衡規劃

產銷平衡的規劃邏輯圖2.3，主要是針對每個儲存單位(Stock Keeping Unit；SKU)以現有的供給存貨來滿足需求。若是供給過多，則是刪除多餘的工單(workorder)；若是供給不足，則是建立新的工單，此一階段同時決定每個工單的生產方法(method)。

Adexa 網路排程的邏輯，圖2.4是根據產銷平衡規劃方式所規劃的工單需求，以產品生產週期為目標來決定各個工單的開工及完工時間。結合 Adexa SCP的規劃方式是以供應鏈網路(supply chain network)中關鍵要徑(critical path)上的工單先行規劃，再規劃非關鍵要徑上的工單，若是在非關鍵要徑上的工單無法決定其合理的開工及完工時間，則將此工單列入關鍵要徑上的工單再重新規劃。決定各個工單的開工及完工時間之規劃邏輯，以前推(forward)方式由上游的工單以最早開工時間為基準，規劃至下游的工單。若無法得到合理的解，則再以後推(backward)方式由下游的工單以交期為基準，規劃至下游的工單。若無法再行得到合理解，則以前推(forward)方式規劃至有合理解為止。

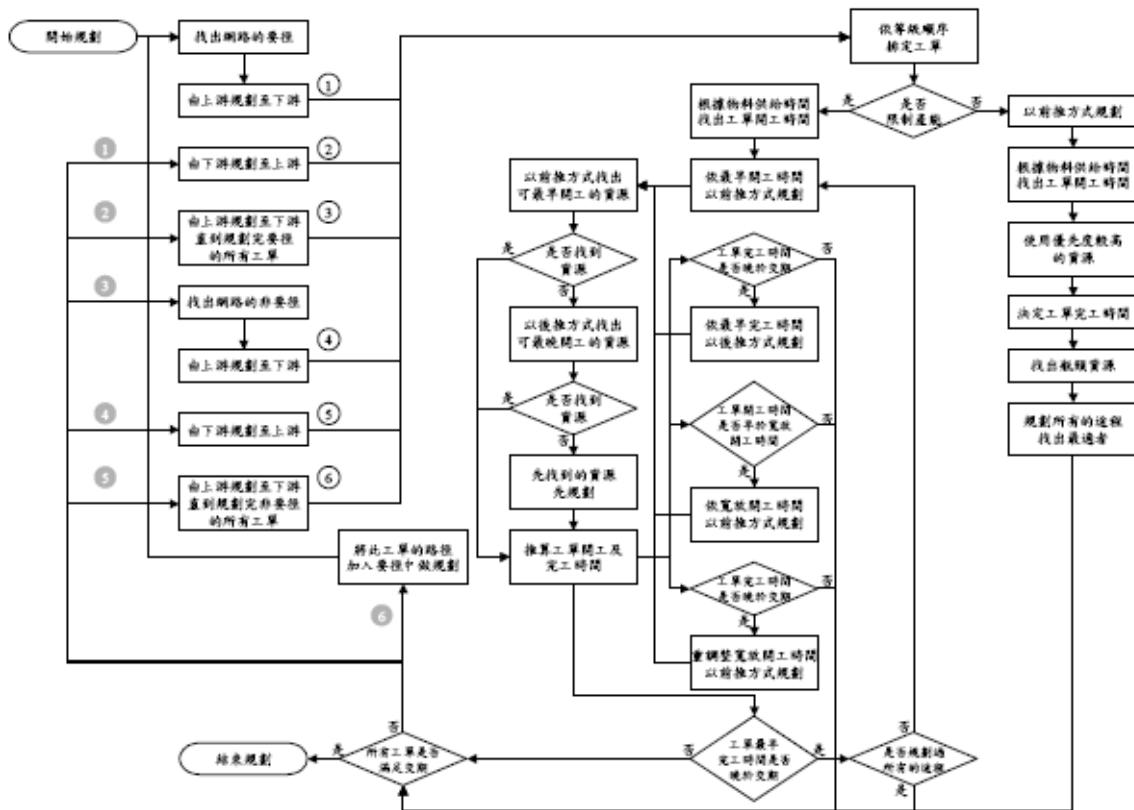


圖 2.4 Adexa SCP 之網路排程的排程邏輯

Adexa SCP 的產銷平衡規劃是在考量供應鏈上的成本最小化之前提下，決定每個儲存單位(commodity/location)的生產方式(method)之外包/生產比率(outsourcing rate/percent)。決定外包/生產的比率所考量的成本如下：

1. 違反安全存貨的成本是每單位低於存貨水準的成本。
2. 持有成本是每單位可存貨空間所造成的成本。
3. 存貨成本是倉儲費用。
4. 單位成本係指生產成本。
5. 低於使用率的成本是資源沒有滿載之情況下所造成的處罰成本。
6. 收入是賣給顧客的品項價格。
7. 延遲成本是顧客訂單交期延遲所造成收入損失的成本。
8. 卸貨成本是將物料送至需要的位置的總成本。

## 2.2 排程問題之描述

所謂排程是在有限資源的限制下，將訂單、或生產計畫轉換成生產活動的過程，考慮生產活動中各個時間點的最佳配置。換言之，排程問題是決定所有工作在機器上加工的起始時間以及加工順序，所以排程問題乃是

同時考慮資源限制和執行限制之最佳化問題。

### 2.2.1 排程問題之分類

在[1]指出排程問題可以依下列四項資訊加以分類

- 1.工作及作業之特性：在排程問題中，若訂單數目及作業內容均為固定且為已知，則此種排程問題稱為靜態(static)排程問題；反之，若訂單數目或作業內容會隨時間之變動而改變，則成為動態(dynamic)排程問題。此外，若訂單在機器上的加工時間為固定，則稱此類排程問題具有確定性(deterministic)；反之，若訂單在機器上的加工時間不固定，為某種機率分配，則稱此類排程問題具有隨機性(stochastic)。
- 2.以工廠中的機器數目及型態可分為
  - (1)單機單階：一台機器，且訂單只有一個作業。
  - (2)多機多階：有多部相同的機器，每張訂單在機器需進行多個作業。
  - (3)多階流程式工廠(Flow shop)：所謂流程式工廠是一多階作業的加工程序，若是所有訂單的機器途程均相同，則此流程型態稱之為淨(Pure)流程式工廠。但若是某些訂單可以跳過某些作業，則稱為非淨(Unpure)流程式工廠。
  - (4)多階零工式工廠(Job shop)：相較於流程式工廠，在零工式的生產環境每一訂單各有其加工順序。
- 3.排程之技術限制：此類限制規則定義了訂單受限於機器的特性，例如
  - (1)一台機器一次僅能加工一個訂單。
  - (2)訂單沒有一定的加工途程。
- 4.衡量標準：排程品質之衡量大致可分為兩類
  - (1)尋求成本最小：常用的衡量標準有生產的固定成本、整備成本、變動成本、存貨成本、延遲成本、與缺貨成本等。運用此類衡量標準必須制訂成本函數始能以成本衡量，而成本函數的設定則較具主觀性。
  - (2)尋求績效最佳：因衡量績效目標不同所產生排程解亦不盡相同，常見的就期滿足、最小化總製距時間、最小化平均流程時間、最小化最大延遲時間...等，Mellor[43]提出 27 種排程可能考慮的目標，如表 2.1。

表 2.1 排程績效目標

1. 設備閒置時間最小化(minimum idle facility investment)
2. 在製品數量最小化(minimum in-process inventory)
3. 設備整備成本最小化(minimum facility set-up cost)
4. 平穩化勞動量(day to day stability of work force)
5. 交期滿足(adherence to promised shipping date)
6. 產出率最大化(maximum output(product rate))
7. 物料持有成本最小化(minimum materials-handling cost)
8. 訂單優先次序的滿足(adherence to arbitrary job priorities), such as arise in dealing with preferred customers, emergency repair parts, etc.
9. 加工合理性(technological feasibility)
10. 變換生產時的敏感度(sensitivity to possible production change)
11. 彈性(general flexibility)
12. (non-dependence on unreliable process)
13. 保留生產力予緊急插單(reverse capacity for rush order)
14. 廠內運送排程最佳化(optimal in-plant transportation schedule)
15. 運送成本最小化(minimum shipping cost)
16. 預期成本最小化(minimum total expected costs, primarily in theoretical investigations)
17. 設備使用率最大化(maximum weighted facility utilization)
18. 人力使用率最大化(maximum utilization of manpower)
19. 勞工績效(optimal assignment of various labor grades)
20. 原物料存貨最小化(minimum raw material in inventory)
21. 最終成品最小化(minimum finished product inventory)
22. 存貨投資最小化(minimum investment in inventories)
23. (minimum obsolescence and deterioration of product)
24. 特定產品製距最小(shortest make-span for certain products)
25. 製距最小化(minimum overall fabrication span)

26. 損失風險最小化(minimum risk of excessive losses)

27. 預防價格的改變(anticipated changes in price)

資料來源：Mellor[43]

### 2.2.2 多目標排程

於現實的排程環境中，生管排程人員所需考量的因素非常的多，Mellor (1966)[43]摘要了 27 項指標，而這些因素可分為定性因素與定量因素。一般所考慮的定量因素有：交期、機器使用率、製距，...等。現實環境中還存有另外的定性因子，例如：公司策略因素、顧客的歷史交易、或是該訂單所需物料的情形，...等。而目前多數關於排程的研究，僅只考慮定量方面的因素，很少提及定性的因素，且其所發展的排程演算法多是以單一目標作為排程績效衡量的基礎，但一般製造環境的生產排程問題並非單一目標可以滿足，皆需以多目標的觀點來考慮。

關於多目標排程的文獻中(Itoh et al., 1993[30]；Daniels, 1994[16]；Neppalli et al., 1996[50]；Murata et al., 1996[48]；Ishibuchi and Murata, 1998[28]；Kim et al., 1998[34]；Min et al., 1998[44]；Chang and Lo, 2001)[11]整理如表 2.2，其所關注的目標多是針對製造現場的生產績效指標，如製距、流程時間、機器使用率、最小延遲時間...等，Baker (1984)[7]指出有關這些排程的績效指標大致可以分成兩種類型：現場時間績效(shop time performance)與交期滿足績效(due date performance)。而 Brown (1997)[9]則提出在整個多目標的環境中，對於在規劃生產排程時應考量市場(marketing)與生產(production)兩大類型的因素。在求解方面，Kim et al. (1998)[34]；Min et al. (1998)[44]是以類神經收集現場狀態搭配派工法則來訓練、建構一排程系統。Ishibuchi and Murata (1998)[28]；Murata et al. (1996)[48]；Neppalli et al. (1996)[50]的研究中採用遺傳演算法來進行搜尋求解，其著重於演算法中各運算子的設計使得求解過程能更有效率，Ishibuchi and Murata (1998)[28]；Murata et al. (1994)[47]為了避免限定演算的搜尋方向，在其研究中的權重是以隨機方式產生。Chou and Lee (1999)[14]；Rajendran and Ziegler (1999)[56]；Danneberg et al. (1999)[16]；Rajendran and Ziegler (2003)[57]；Allahverdi (2003)[4]；Lin and Liao (2003)[42]各利用不同法則的啟發式演算法求解流程式排程的問題。Chang and Lo (2001)[11]以零工式排程環境，以 GA 和 TS 啟發式演算法解決多目標排程問題，此多目標函數結

合了質性與量性指標，成功解決多目標排程問題。Lee (2001)[37]求解具有最小化交期懲罰函數、存貨成本以及加工成本為績效目標之多機雙階製造系統的排程問題，其利用許多種求解方法並比較其優劣，而其中以雙階段的禁忌搜尋法之求解表現較佳。Esquivel et al. (2002)[17]利用優先順序列表決定最小化多目標函數中製距、提早完成時間及加權完成時間中各項權重，並以提升轉移演算法.(Enhance devolutionary algorithms)求解。Chang et al. (2002)[12]針對多目標流程式排程問題，運用 Gradual-Priority Weighting (GPW)制訂各項績效目標權重，並使用遺傳演算法求解。Framinan et al. (2002)[20]根據製距及流程時間最小化作為績效目標，且使用 posteriori heuristic 制訂權重，以 NEH(Nawaz Enscore Ham, NEH)啟發式解法求解。Ghrayeb (2003)[21]利用遺傳演算法以模糊的觀點出發，求解模糊製距(FM)不確定性及積分值最小化作為績效目標的排程問題。Ishibuchi et al. (2003)[29]以製距、最大延遲最小化為目標，運用局部搜尋多目標遺傳演算法(MOGLS) 及多目標演化尋優法求解。

表 2.2 多目標排程文獻整理

作者	排程型態	求解方法	考量之目標	權重制訂
Itoh et al.(1993) [30]	零工式排程	TLAS	最小化平均延遲時間 最小化平均流程時間	設定為 0.5
Daniels(1994) [16]	單機排程	派工法則	最小化總流程時間 最小化最大延遲 最小化延遲數目	依照各目標成本制訂
Murata et al. (1996)[48]	流程式排程	遺傳演算法	最小化製距 最小化總延遲 最小化總流程時間	隨機方式決定
Neppalli et al. (1996)[50]	流程式排程	遺傳演算法	最小化製距 最小化總流程時間	研究中採用二種方法將族群分為兩個子族群，分為依照單一目標進行運作，權重設定為 1:1



Kim et al. (1998)[34] Min et al. (1998)[44]	彈性製造系統	競爭式類神經訓練機器、工件、儲位以及搬運車選擇之派工法則搭配組合	最小化總流程時間 最小化平均延遲時間 最小化平均流程時間 加工機器平均使用率 搬運車平均使用率 最小化製距 系統平均工單數 平均在製品	依照各單一目標設定水準值，再予以評估。
Chou and Lee(1999)[14]	流程式排程	啟發式演算法	加權總流程時間 製距	自訂權重，相加為 1
Rajendran and Ziegler(1999)[56]	流程式排程	啟發式演算法	加權總流程時間 加權延遲工作	直接加總
Chang and Lo (2001)[11]	零工式排程	GA/TS	作業順序 (job-sequence) 最小化製程時間 交期滿足 機器利用率	層級分析法 (AHP) 訂定權重
Lee(2001)[37]	多機雙階製造系統	雙階段禁忌搜尋法；單階段禁忌搜尋法；遺傳演算法；模擬退火法；鄰近空間搜尋法	交期懲罰函數 存貨成本 加工成本	權重設定為 1：1：1
Kacem et al.(2002)[32]	彈性零工式排程	AL/CGA	最大機器利用率 總機器利用率 製距	權重設定為 1：1：1
Esquivel et al.(2002)[17]	零工式排程	提升轉移演算法 (Enhance devolutionary algorithms)	製距 提早完成時間 加權完成時間	優先順序列表決定
Chang et al.(2002)[12]	流程式排程	GA	製距 總流程時間 總延遲 最大延遲	運用 gradual-priority weighting (GPW) 制訂權重
Framinan et al.(2002)[20]	流程式排程	NEH 啟發式解法	製距 流程時間	以 posteriori heuristic 制訂權重

Ghrayeb (2003)[21]	零工式排程	遺傳演算法	模糊製距的不確定性 模糊製距的積分值 Minimizing the integral value and the uncertainty of the fuzzy makespan (FM)	自訂權重 $l, w$
Ishibuchi et al.(2003)[29]	流程式排程	局部搜尋多目標遺傳演算法(MOGLS) 多目標演化尋優法	製距 最大延遲	pseudoweight vector
Rajendran and Ziegler(2003) [57]	流程式排程	啟發式演算法	加權流程時間 加權延遲工作	加總
Allahverdi(2003) [4]	流程式排程	啟發式演算法	加權製距和 平均流程時間	設定為 0.1,0.9
Lin and Liao(2003)[42]	流程式排程	啟發式演算法	加權最大延遲	根據各成本設定權重
Allahverdi(2004) [5]	流程式排程	DCH/SAH/MNEH/PH	製距 最大延遲	實驗三組： 0.25 : 0.75 0.5 : 0.5 0.75 : 0.25
Birman and Mosheiov (2004) [8]	流程式排程	Johnson 演算法	最大提早完成 最大延遲 交期成本	自訂權重

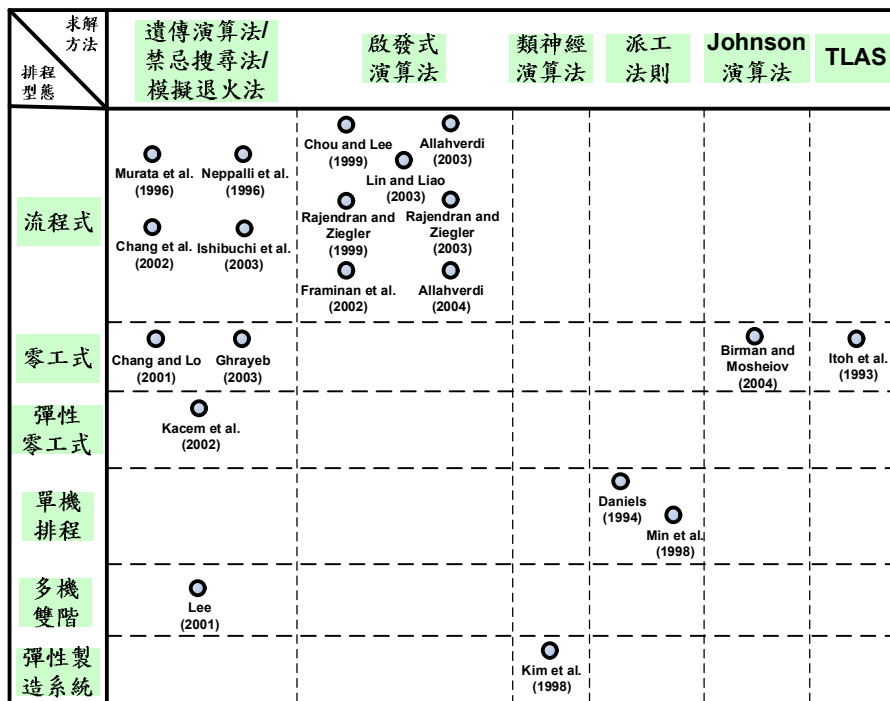


圖 2.5 多目標排程文獻矩陣示意圖

## 2.3 遺傳演算法

遺傳演算法(Genetic algorithm; GA)為 Holland 於 1975 年所提出，其主要的想法是發展一人工系統(artificial system)模擬自然生態運作的方式，藉由演算法的自行淘汰與部分交換來求解最佳化的問題。Goldberg[26]提到遺傳演算法是以自然選擇與遺傳技術為基礎的搜尋過程。Michalewicz[45]更明確指出遺傳演算法是由五個基礎步驟所構成：

1. 以基因型態表示問題的特徵或解答：遺傳演算法對問題的表示方式是将不同的問題特徵（或變數）分別以一個或一組基因表示，其中一基因是二元(binary)數字，而如同一個體是由數個基因所構成，問題的解答也就是由不同的問題特徵所組成。

2. 創造任意數目的初始解答：在遺傳演算法開始運作之前，需先產生一些初始的解答做為初始的狀態，亦即讓電腦創造一些「數位個體」，形成原始族群(initial population)再進行演算搜尋。至於產生這些個體的方式，分為隨機或是特定的方式，Forgaty[19]則提出產生初始狀態的方法會影響演算法的搜尋績效。

3. 評估功能的建立-適應性函數(fitness function)的設計：若一個體的適應能力愈高，代表此一個體在環境中存活下來的機會較高，則愈有可能繁殖下一代。相對於適應函數的是在解決問題時之目標函數，藉由目標函數來評判解答接近預設目標的程度，若目標函數值愈高則代表此解答愈接近目標，也愈有機會讓遺傳演算法進一步搜尋到更好的解答個體。

4. 使用基因運算子(genetic operator)產生子代：最常見的基因運算子

有以下三個。

(i)複製(reproduction)：此運算子的功能決定哪些個體可以存活至下一代，而根據「適者生存」的原則，適應函數值高的個體應具有較高的機率被選中複製而存留至下一族群。

(ii)交配(crossover)：此運算子的功能是透過交換個體間的基因，以重組個體的基因組合，來擴展搜尋空間。Murata[47]以 10 種不同的交配方式對於流程式排程問題進行電腦模擬測試，以決定不同交換方式的績效優劣。

(iii)突變(mutation)：此運算子的功能是藉由隨機改變個體內的基因，引入新的個體型式，增加新的搜尋空間。突變的發生是隨機的，以使得在求解

的過程中能夠搜尋新的領域，避免掉入局部最佳解(local optimal)。Murata[47]以 4 種不同的突變方式來進行試驗。

5. 參數的設定：在遺傳演算法的運算過程中，有許多的參數必須事先設定，如交配機率、突變機率、個體數及族群數等。參數的設定會影響搜尋的績效，Forgaty[19]曾對不同的突變機率與遺傳演算的績效進行評估，在特定的原始族群中，不同的突變機率可以改善搜尋績效。一般遺傳演算法運作的流程大致如圖 2.1。

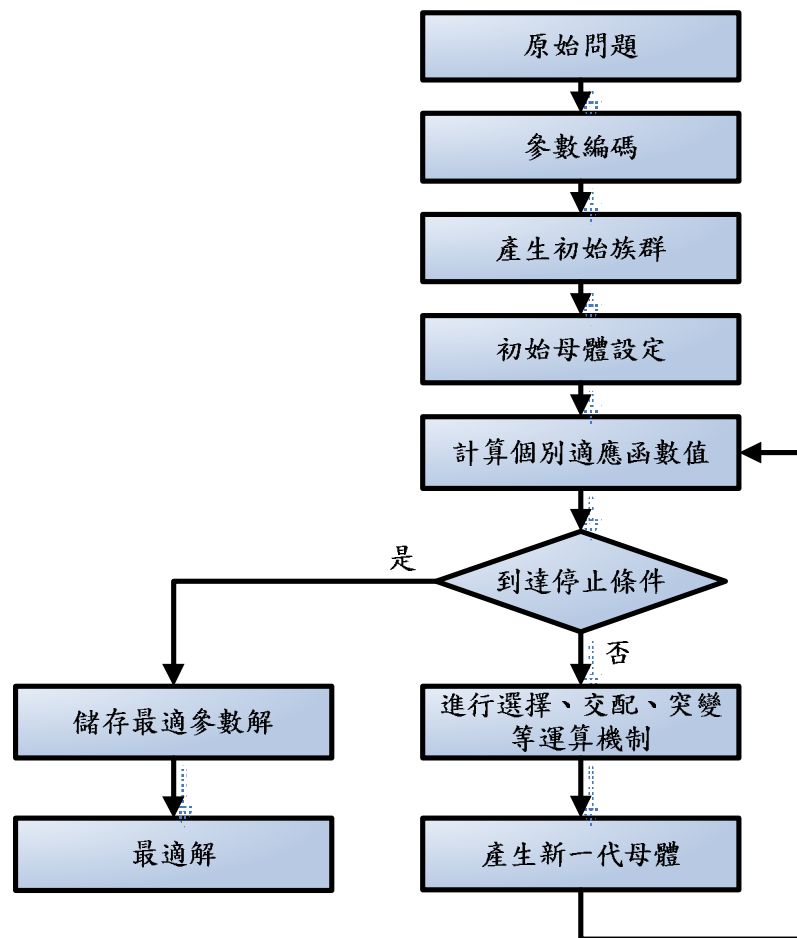


圖 2.6 一般遺傳演算法運作流程

### 2.3.1 常見 Job shop 排程問題的編碼表示法

在 Cheng[13]的研究報告中，曾調查使用遺傳演算法求解 JSP 時，常見的表示方法。說明如下：

#### 1. Operation-based representation

此類型的編碼方式是將一個排程編碼成作業的順序，而染色體中的每

一個基因代表一作業。亦即以一個  $m$  部機器與  $n$  個訂單的 Job shop 排程問題(Job-shop Scheduling Problem; JSP)而言，它的每一染色體包含有  $m \times n$  個基因。但是，由於訂單本身的作業有一定的加工先後順序限制，經過交配後所產生的基因順序不能保證能獲得一可行的排程，所以這種編碼方式必須加入修正調整，研究中的修正方式是將同一訂單中的作業在染色體中皆以同一個代號表示。

在這個表示法中，如果問題大小是  $n$  個訂單與  $m$  部機器，染色體會含有  $n \times m$  個基因，而染色體中每個訂單只會出現  $m$  次，每個基因是表示工作內的相對步驟，如此一來使得每一染色體所產生的排列皆是可行解。以一  $3 \times 3$  的問題為例，如果染色體中的基因排列是  $[2\ 1\ 1\ 1\ 2\ 3\ 2\ 3\ 3]$ ，經過解碼轉換後為  $[O_{21}\ O_{11}\ O_{12}\ O_{13}\ O_{22}\ O_{31}\ O_{23}\ O_{32}\ O_{33}]$ ， $O_{ij}$  表示第  $i$  張訂單的第  $j$  個作業，最後在排程時依照作業加工順序來進行排程規劃。

## 2. Job-based representation

此類型只有將訂單做編碼，得到一個訂單的優先順序，排程時是先安排訂單優先次序為 1 的訂單之所有作業，之後，依訂單的優先順序完成整個排程。此法的原則是先決定訂單優先順序，再按其作業程序來安排所有作業。

但是，這種方式與 shifting bottleneck heuristic 有類似的假設：當每一機器的排程是最佳化的時候整個排程即是最佳解，所以這種編碼方式並不能保證可以包含最佳解於其解答空間之內。

## 3. Preference list-based representation

以一個  $m$  部機器與  $n$  個訂單的 JSP 而言，此法的表達方式：每一染色體由  $m$  個子染色體 (subchromosome) 所構成，每一子染色體是一個長度為  $n$  的符號字串，每一個符號代表一個機器的相關作業。子染色體並不是該機器上作業的加工順序，而是該機器的喜好表列 (preference list)。整個排程是先找出每部機器的第一喜好作業，再依據每一訂單作業順序限制，判斷機器的第一喜好作業是否可以排入排程之中，若不能則保留。否則找出機器的下一喜好作業，持續至完成所有作業。這種方法只能排出非延遲 (non-delay) 的排程，可能無法將最佳解含入其中。

## 4. Job pair relation-based representation

此法是利用一個二元矩陣來，該矩陣是表示二個訂單在機器上的先後

關係。其二元的關係定義如下

$X_{ijm} = 1$ ；如果在機器  $m$  上，訂單  $i$  處理時間比訂單  $j$  早。  
0；其它。

因此一個  $3 \times 3$  的問題染色體如果是  $[0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0]$  其二元矩陣如下：

訂單(1, 2)在機器( $m_1, m_2, m_3$ ):  $X_{121} \quad X_{122} \quad X_{123} = 0 \quad 1 \quad 0$

訂單(1, 3)在機器( $m_1, m_2, m_3$ ):  $X_{131} \quad X_{132} \quad X_{133} = 1 \quad 0 \quad 1$

訂單(2, 3)在機器( $m_1, m_2, m_3$ ):  $X_{231} \quad X_{232} \quad X_{233} = 1 \quad 1 \quad 0$

然後再依照這個二元矩陣完成整個排程。這個方法最大的缺點是隨著問題變大，會產生愈多不合理的染色體，必須加入其它的調整法則來修正。

### 5. Priority rule-based representation

這種方法中每一染色體代表派工法則的順序，基因演算法是用來尋求出一個較好的派工法則之順序。以一個  $m$  部機器與  $n$  個訂單的 JSP 為例，一個染色體是一個  $n \times m$  的字串  $(p_1, p_2, \dots, p_{nm})$ ， $p_i$  表示第  $i$  個循環所使用的派工法則，整個步驟如下：首先列出每個訂單的第一個作業做為可選擇作業，選取加工時間最少的作業，若加工此作業的機器還需要加工這循環中的其它作業，則使用該循環的派工法則決定何者應被選取，進入下一循環。下一循環開始時須將上一循環中加工被選取的訂單之下一作業納入本循環的可選擇作業。每一循環選取一個作業，直到完成整個排程。使用 Priority rule-based 編碼方式所得的排程解，其缺點是品質很不穩定。

### 6. Disjunctive graph-based representation

這個方法也可以視為是 Job pair relation-based representation 的一種。排程的問題可以利用分支圖（如圖 2.2）來表達， $G = (N, A, E)$ ： $N$  代表節點來表示所有的作業， $A$  代表用來連結同一訂單相連的作業（實線部分）， $E$  用來連結同一機器上的作業（虛線部分）。圖中箭頭方向代表作業之間的優先次序， $N$  與  $A$  都是確定的，而為了確保機器上的作業不會有加工順序衝突的現象發生，每個機器的作業節點不可以行成一循環(acyclic)，用  $e_{ij}$  來表示  $E$  中的每一個箭頭。

$e_{ij} = 1$ ，當箭頭的方向是由節點  $i$  至節點  $j$ ，亦即先加工作業  $i$  再加工作業  $j$ 。

$e_{ij} = 0$ ，當箭頭的方向是由節點  $j$  至節點  $i$ ，亦即先加工作業  $j$  再加工作業  $i$ 。

利用分支圖來做編碼時，在  $n \times m$  的 JSP 中其染色體為一包含有  $n \times m$  個

基因  $e_{ij}$ 。以圖 2.2 為例，其基因如下：

$$\begin{aligned} & [ e_{15} \quad e_{19} \quad e_{59} \quad e_{24} \quad e_{28} \quad e_{48} \quad e_{36} \quad e_{37} \quad e_{67} ] \\ & = [ 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 ] \end{aligned}$$

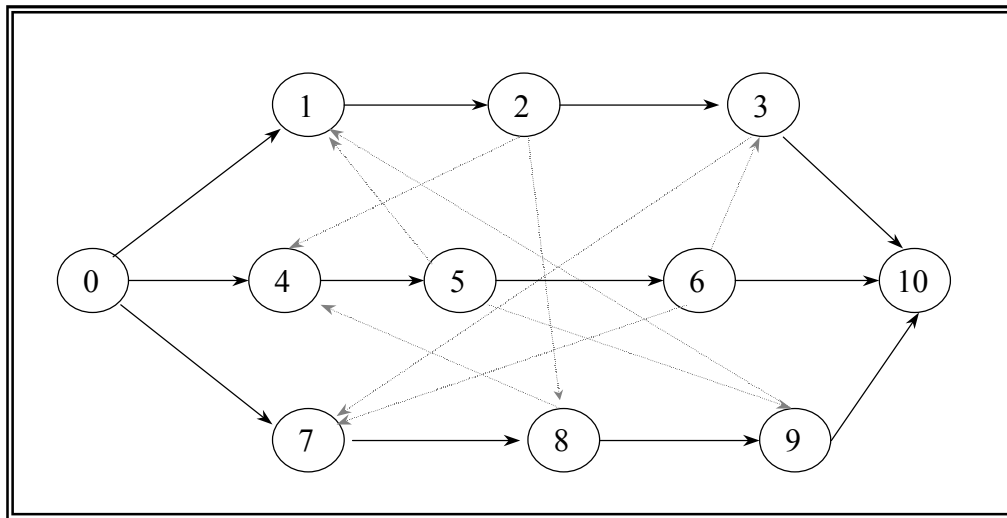


圖 2.7 分支圖

## 7. Random key representation

在 Random key representation 中，所有的解答被編碼為一串隨機數 (random key)，用這些數字作為排序的依據。如果問題是  $n$  個訂單和  $m$  個機器，每個基因 (隨機數) 被視為兩個部分，一個是整數部分，其集合為  $\{1, 2, \dots, m\}$  (代表在哪一個機器執行，因此數字 1 至  $m$  的出現次數一定是  $n$  個)，另一部份是介於  $0 \sim 1$  的小數，在解碼的時候要將同機器數字來做比較，後面小數部分的大小代表作業的優先次序。假設在一  $3 \times 3$  的問題中有一染色體如下：

$$[ 1.34 \quad 1.09 \quad 1.88 \quad 2.66 \quad 2.91 \quad 2.01 \quad 3.23 \quad 3.21 \quad 3.44 ]$$

在機器 1 上的小數部分按照小到大排列順序為  $[ 2, 1, 3 ]$ ，而在機器 2 上的順序為  $[ 3, 1, 2 ]$ ，機器 3 上的順序為  $[ 2, 1, 3 ]$ ，因此將這個染色體轉換成實際機器順序下：

$$[ O_{21} \quad O_{11} \quad O_{31} \quad O_{32} \quad O_{12} \quad O_{22} \quad O_{23} \quad O_{13} \quad O_{33} ]$$

使用這種方式來轉換的排程很容易違反訂單加工途程的限制，必須搭配一些特別的解碼方式。另外，在產生初始族群、交配、突變時所產生的染色體整數部分必須符合數目的限制。

上述之表達法可以分為二個類型：直接式與間接式。所謂的直接式是將一個排程解編碼成一染色體，在透過基因演算法的運算，得到一較佳的

排程。而間接式的染色體則是以優先順序法則做為編碼依據，如 Priority rule-based representation 中，其是要透過基因演算法找出較佳的派工法則之順序。

### 2.3.2 常見排序問題之運算子

使用遺傳演算法求解時，依據問題特性的不同，必須設計適當運算子，讓整個演算過程能順利的進行。在解決排序問題時，一個染色體是由許多“不重覆”的數字基因所構成的字串。在這個限制之下，如果以傳統簡易遺傳演算法中所提的運算子來搜尋，很有可能會產生一個個體中有兩個相同的數字基因。為了避免發生重覆的情況，必須要設計新的運算子，來產生合理的子代。在[18]提及關於遺傳演算法中交配運算子可以分為以下幾種類型：

1. 單點交配(one-point crossover)：根據育種選擇策略選擇的二個母體，隨機產生二個切點進行交配。

父代 1 : [ 7 3 | 7 6 1 3 ] → 子代 1 : [ 7 3 | 4 5 2 2 ]

父代 2 : [ 1 7 | 4 5 2 2 ] → 子代 2 : [ 1 7 | 7 6 1 3 ]

子代 1 中第一部份的基因是繼承父代 1 基因而來，第二部分繼承父代 2，子代 2 則是繼承父代 1 第二部份的基因與父代 2 第一部份的基因。

2. 雙點交配(two-point crossover)：隨機產生二個切點，交換父代彼此的基因。

父代 1 : [ 7 3 | 7 6 | 1 3 ] → 子代 1 : [ 7 3 | 4 5 | 1 3 ]

父代 2 : [ 1 7 | 4 5 | 2 2 ] → 子代 2 : [ 1 7 | 7 6 | 2 2 ]

3. N 點交配(N-point crossover)：隨機產生 3 至 n 個切點數，決定父代交換奇數或是偶數部分的基因。以下為例：產生 3 個切點 (1、2、4) 交換奇數部分的基因，產生二子代。

父代 1 : [ 7 | 3 | 7 6 | 1 3 ] → 子代 1 : [ 7 | 7 | 7 6 | 2 2 ]

父代 2 : [ 1 | 7 | 4 5 | 2 2 ] → 子代 2 : [ 1 | 3 | 4 5 | 1 3 ]

4. 均於交配(uniform crossover)：每一基因的位置皆產生一個 0~1 的亂數，假如亂數 > 0.5，則該位置的基因不交換。舉列而言，在下列中的中產生的亂數依序為 0.2, 0.7, 0.9, 0.4, 0.6, 0.1，產生的子代如下：

父代 1 : [ 7 3 7 6 1 3 ] → 子代 1 : [ 7 7 4 6 2 3 ]

父代 2 : [ 1 7 4 5 2 2 ] → 子代 2 : [ 1 3 7 5 1 2 ]



在[1]的研究中，將一些常用在排序問題的運算子，整理如下。

### 1.PMX (Partially Matched Crossover)

(1)隨機產生兩個切點。

個體 A : [ 9 8 4 | 5 7 6 | 1 3 2 ]

個體 B : [ 8 7 1 | 2 3 6 | 9 5 4 ]

(2)將個體 A 與個體 B 在兩切點中的基因互調。

個體 A : [ 9 8 4 | 2 3 6 | 1 3 2 ]

個體 B : [ 8 7 1 | 5 7 6 | 9 5 4 ]

(3)將個體 A 位於切點之外重覆的基因與個體 B 位於切點之外重覆的基因互調。

個體 A : [ 9 8 4 | 2 3 6 | 1 7 5 ]

個體 B : [ 8 3 1 | 5 7 6 | 9 2 4 ]

### 2. LOX(Linear Order Crossover)

(1)隨機產生兩個切點。

個體 A : [ 9 8 4 | 5 7 6 | 1 3 2 ]

個體 B : [ 8 7 1 | 2 3 6 | 9 5 4 ]

(2)將個體 A 切點中的所有位元，在個體 B 中以\*代替；B 者亦然。

個體 A : [ 9 8 4 | 5 7 \* | 1 \* \* ]

個體 B : [ 8 \* 1 | 2 3 \* | 9 \* 4 ]

(3)將\*往中間移動，使得兩個切點中的位元皆為\*。

個體 A : [ 9 8 4 | \* \* \* | 5 7 1 ]

個體 B : [ 8 1 2 | \* \* \* | 3 9 4 ]

(4)將原本個體 A、B 切點中的位元互調。

個體 A : [ 9 8 4 | 2 3 6 | 5 7 1 ]

個體 B : [ 8 1 2 | 5 6 7 | 3 9 4 ]

### 3. SX(Simple Crossover)

(1)隨機產生 1 個切點。

個體 A : [ 9 8 4 | 5 7 6 1 3 2 ]

個體 B : [ 8 7 1 | 2 3 6 9 5 4 ]

(2)保留切點左邊的基因，切點右邊的位元以其在另一個體的順序填入。

個體 A : [ 9 8 4 | 7 1 2 3 6 5 ]

個體 B : [ 8 7 1 | 9 4 5 6 3 2 ]

#### 4. RX(Random Crossover)

(1)隨機產生兩個切點。

個體 A : [ 9 8 4 | 5 7 6 | 1 3 2 ]

個體 B : [ 8 7 1 | 2 3 6 | 9 5 4 ]

(2)切點兩旁的基因保留不變，兩切點中的基因以隨機產生。

個體 A : [ 9 8 4 | 6 5 7 | 1 3 2 ]

個體 B : [ 8 7 1 | 3 6 2 | 9 5 4 ]

#### 5. CX(Cycle Crossover)

(1)在個體 A 中任選一個基因，假設選到 9，其相對在個體 B 為 1，將其標示起來。

個體 A : [ 9 8 2 1 7 4 5 6 3 ]

個體 B : [ 1 2 3 4 5 6 7 8 9 ]

(2)個體 A 中基因為 1 的位元在個體 B 中是 4，將其標示起來。

個體 A : [ 9 8 2 1 7 4 5 6 3 ]

個體 B : [ 1 2 3 4 5 6 7 8 9 ]

(3)個體 A 中基因為 4 的位元在個體 B 中是 6，將其標示起來。

個體 A : [ 9 8 2 1 7 4 5 6 3 ]

個體 B : [ 1 2 3 4 5 6 7 8 9 ]

(4)個體 A 中基因為 6 的位元在個體 B 中是 8，將其標示起來。

個體 A : [ 9 8 2 1 7 4 5 6 3 ]

個體 B : [ 1 2 3 4 5 6 7 8 9 ]

(5)重覆以上的步驟，直到最後的標示回到 9。

個體 A : [ 9 8 2 1 7 4 5 6 3 ]

個體 B : [ 1 2 3 4 5 6 7 8 9 ]

(6)將個體 A、B 中沒有被標示的基因互換。

個體 A : [ 9 8 2 1 5 4 5 6 3 ]

個體 B : [ 1 2 3 4 7 6 7 8 9 ]

#### 6. OM(Order-based Mutation) :

任選兩個基因，將其互調即可。

[ 9 8 4 5 7 6 1 3 2 ] → [ 9 8 4 1 7 6 5 3 2 ]

#### 7. PM(Position based Mutation)

任選兩個基因，假設為 5、1，將 7、6、1 往前移，再將 5 填入原本基因 7 的位置即可。

[ 9 8 4 5 7 6 1 3 2 ] → [ 9 8 4 7 6 1 5 3 2 ]

### 2.3.3 排程中遺傳演算法之應用

排程問題在生產管理是相當重要的研究領域。大都利用啟發式演算法解決此類問題，達到最適或次佳解。Petty et al. (1987)[52]提出分散式的遺傳演算法，將一族群分割成較小的族群，再分別進行 GA 運作，以提高求解效率。Kim et al. (1994)[33]以 GA 為基礎發展一啟發式搜尋技術解決 Job shop 的排程問題。Croce et al. (1995)[15]利用偏好法則(preference rules)提出一套遺傳演算法，分別與瓶頸漂移法、模擬退火法以及禁忌搜尋法比較實驗。Pinedo et al. (1995)[54]指出，遺傳演算法可以被應用在結構未知的問題上。Wellman et al. (1995)[67]則提到遺傳演算法已經成功地應用在最佳化求解的問題，如排程、運輸問題、旅行者問題等。Kumar and Srinivasan (1996)[36]依據不同情境下，並且從七種不同法則中採用不同派工法則進行排程。Murta et al. (1996)[48]以製距最小為目標，評估 GA 分別結合局部搜尋法、模擬退火法在 Flow Shop 排程問題上的表現。Norman and Bean (1999)[51]利用隨機法則，發展出一套方法，並應用在汽車工業。Liaw (2000)[41]利用混合式遺傳演算法(HGA)，即傳統遺傳演算法結合禁忌搜尋法，求解開放式(open shop)排程。Yun (2002)[68]提出一套遺傳演算法(GA)結合模糊邏輯控制器(fuzzy logic controller)，求解先占式(preemptive)以及非先占式(non-preemptive)零工式(job-shop)排程。Park et al. (2003)[53]分別利用單一遺傳演算法(SGA)以及平行式遺傳演算法(PGA)，並與傳統遺傳演算法比較。

基本上遺傳演算法與其他搜尋方式比較起來，能夠以較短的時間搜尋到不錯的解答，對問題變數的多寡影響相對的比較小，有其他搜尋演算法所缺乏的優點。但是在實際應用時，遺傳演算法在不同的問題上仍有一些缺點，尚待克服。今將遺傳演算法的優缺點歸納如下表 2.3：

表 2.3 遺傳演算法之優缺點歸納表

<b>優點</b>	<p><b>1. <u>多點同步搜尋</u></b></p> <p>遺傳演算法同時考慮搜尋空間上多個點而不是單一個點，因此可以較快地獲得整體最佳解(global optimum)，同時也可以避免陷入區域最佳解(local optimum)的機會，此項特性乃是遺傳演算法的最大優點。</p> <p><b>2. <u>使用適應函數</u></b></p> <p>遺傳演算法的運作過程只使用適應函數的資訊而不需要其它輔助的資訊(例如:可微分)，因此可以使用各種型態的適應函數(例如:多目標、非線性或以知識為基礎)，並可節省計算資源避免繁複的數學運算。</p> <p><b>3. <u>機率式的搜尋</u></b></p> <p>遺傳演算法是使用機率規則(stochastic)的方式去引導搜尋方向，而不是用明確(deterministic)的規則，因此較能符合各種不同類型的最佳化問題。</p> <p><b>4. <u>基因編碼</u></b></p> <p>遺傳演算法是以參數集合之編碼進行運算而不是參數本身，因此可以跳脫搜尋空間上的限制。</p>
<b>缺點</b>	<p><b>1. <u>運算子的設計問題</u></b></p> <p>根據不同的問題，需要設計不同功能的運算子，以提高搜尋效率。若單純使用簡易遺傳演算法的運算子，搜尋速度會受到影響。</p> <p><b>2. <u>重複搜尋的問題</u></b></p> <p>因為遺傳演算法並沒有記憶功能，且其運作過程只與適應函數相關，因此往往在搜尋的過程中，重複搜尋到相同的點，增加系統搜尋的時間。</p>

## 2.4 禁忌搜尋法

### 2.4.1 禁忌搜尋簡介

禁忌搜尋法又名禁忌搜尋法為 Glover 於 1977 年所提出，是一種可以用來處理組合爆炸問題的演算法。Glove 在[22]指出禁忌搜尋法具有三個主要特性：(1)使用彈性記憶結構(flexible memory structure)，較其它固定記憶（如分枝界限法）或無記憶結構（如模擬退火法）更能充分利用歷史搜尋資訊。(2)應用記憶彈性結構的一個關聯控制機制，透過對條件的限制與釋放，來控制搜尋過程。(3)從短期到長期，合併不同時間間距的記憶功能，以實行強化與多變的搜尋策略。

Glover 在[24]中，指出禁忌搜尋法的主要組成要素有：

1. 起始解(starting solution)：由最原始的可行解中，產生一啟發解作為搜尋的起始點。
2. 鄰域解(neighbor solution)：不屬於記憶架構內的紀錄，並符合問題限制之可行解。而鄰域解的移動取法有兩種交換式變動(swap move)與插入式變動(insert move)，因為一個插入式變動是由兩次交換式變動所造成，所以插入式變動對目前的解所造成的變動程度較交換式高，但必須視問題的型態作最佳的選取。
3. 移動路徑(move)：在目前的鄰域解中，選取評估值最高者作為改善之移動路徑，若此鄰域解已在禁忌串列中，且又不符合凌駕準則選取次佳之鄰域解作為改善移動的路徑；若此鄰域解符合凌駕準則，則其不受禁忌串列的限制。
4. 記憶架構(memory structure)：記憶結構亦可稱為禁忌串列(tabu liust)，記憶結構又可分為「短期記憶」與「長期記憶」，短期的架構只記錄該次搜尋前的數次記憶，長期記憶則是記錄從開始運作以來所有的搜尋過程。記憶期數的大小必須依照問題的特性和複雜度決定，Glover[23]建議可使用魔術數字 7 作為禁忌串列的尺寸。
5. 凌駕準則(Aspiration criterion)：考慮一鄰域解，若此一移動所得之鄰域解之評估值比截至目前為止搜尋過的最佳解還要好，若其恰巧在禁忌串列的限制中，此時應破除禁忌串列的限制，接受此一移動。
6. 停止條件(stop cirterion)：終止搜尋進行的條件，一般最常是以一預設最

大循環次數，當到搜尋循環達此一次數及終止搜尋。

在長期階段使用強化性(intensification)與多樣性(diversification)將搜尋帶入新的區域以求出最佳的解。所謂的強化性與多樣性乃是指當搜尋終止後，由選取最佳解或是最後一次的可行解，清除所有禁忌串列重新開始進行搜尋，其目的是為了增加搜尋時跳脫區域最佳解的機會。短期記憶的功能是避免下一次移動時，發生重複選取的現象，而長期記憶是記錄所有移動決策及其發生過的次數，計算其頻率的記憶(frequency base memory)，使其能提供資訊來補足短期記憶的不足。

#### 2.4.2 禁忌搜尋在排程上之應用

Wesley[66]以 14 種派工法則求出起始解，再利用禁忌搜尋法求解 Job shop 排程問題。Adenso[3]利用禁忌搜尋法結合模擬退火法求解流程式排程問題，利用模擬退火法中接受較差解的機制，避免求解過程中掉入局部最佳解的情形。Liaw[40]的研究中利用禁忌搜尋法結合遺傳演算法求解流程式排程問題，以最小化製距為目標，然後利用 Tabu 搜尋法對每一個體解依其關鍵路徑(critical path)進行鄰域搜尋修正改善。

### 2.5 本章小結

從以上的文獻探討中，可以發現遺傳演算法已被廣泛的運用在求解最佳化的問題上，本研究將以此種搜尋法並結合禁忌搜尋法做為搜尋的機制。在本研究中將提出一考量策略性因素與現場績效因素的多目標訂單指派與排程的方法。此外，一般關於多廠間的訂單指派文獻中大都將訂單指派與排程分開探討，本研究的系統中將訂單指派和生產排程問題一同納入考量，系統架構的詳細機制設計，將在第三章中加以說明。

# 第三章 多廠整合型生產指派與排程系統之架構設計

## 3.1 系統架構原理

在多廠間的訂單指派與排程中，大都分為兩階段實施，在指派階段依據廠與廠間相關產能、製程技術、生產品質等因素做訂單指派動作，而後作第二階段排程動作，然而，個別最佳化，並無法滿足整體最佳化的要求，因此這種作法並無法有效求解整體系統最佳化。因此，本研究針對此問題提出一最佳化之系統架構。

在第一階段，訂單指派動作便同時考慮了排程問題，例如機台選擇、作業排程、各零件訂單之生產順序及各零件訂單分割之批數及大小，透過每代的零件訂單分割至不同衛星廠生產，造成每代零件訂單指派順序演化，如圖 3.1。本研究提出此系統架構，目的是整體系統求解最佳化，並縮短訂單交期時間，提高工廠的訂單達交率。最後在求解問題時，採用傳統遺傳演算法(GA)、混合式遺傳演算法(HGA, Hybrid-GA)求解比較。

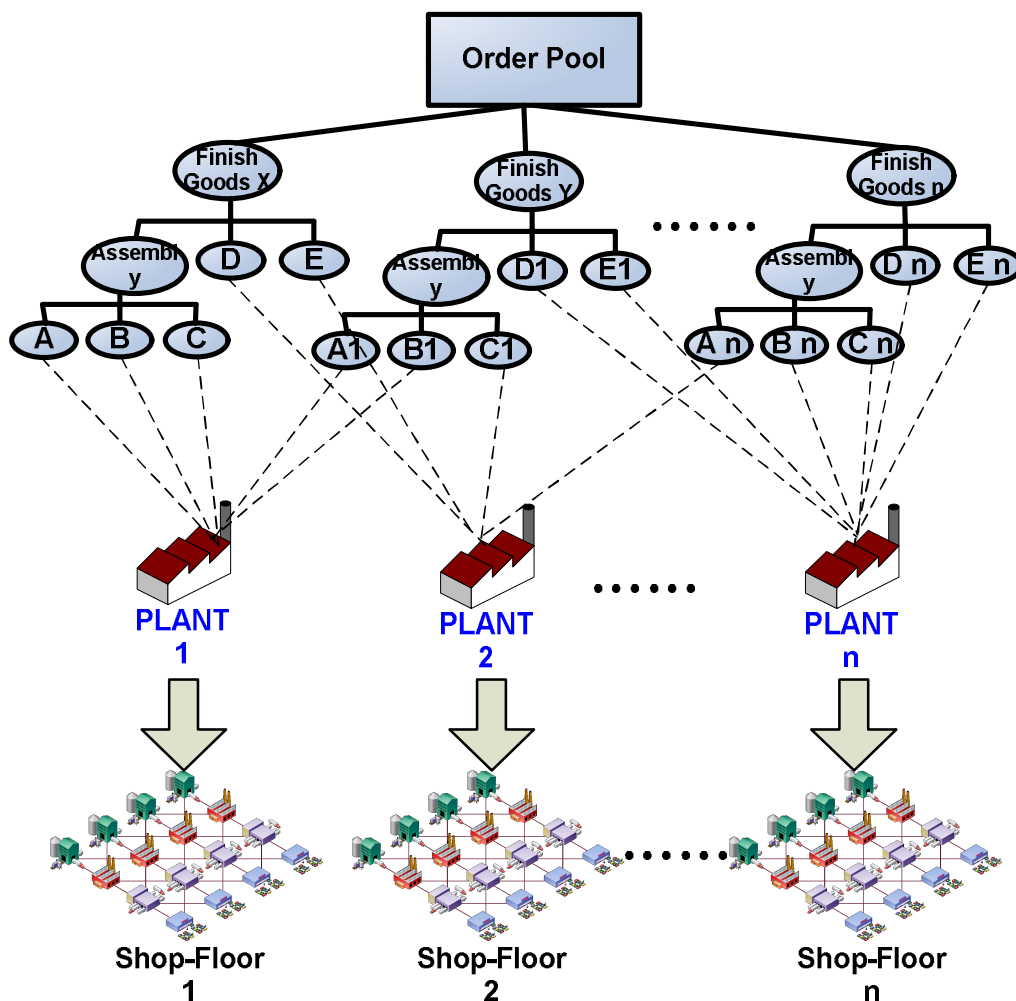


圖 3.1 多廠區零件訂單分配示意圖

一般廠內預排時，很少將廠內的生產計畫與在排程規劃時的製距、交期及機器使用率等廠內績效衡量的指標一併納入考量，因此在指派過程中易出現產能不平衡的狀態，為解決此一問題及提升多廠間的整體性生產優勢。本研究提出整合型訂單指派與生產排程評估方式，即其系統架構如圖 3.2 所示。

現實的排程環境中，在規劃階段中排程人員所需考量的因素非常的多，這些因素又可分為定性因素與定量因素（如表 3.1）。一般考慮定量的因素有：交期、現場機器使用率、製距等。然而，現實環境中還存有另外的質性因子，例如：公司策略因素、顧客的交易歷史、或是該訂單所需物料的情形，...等。而目前多數關於排程的研究，其所發展的排程演算法多是只考慮定量方面的因素，很少提及定性的因素。

表 3.1 定性與定量因素表

定性因素	定量因素
市場考量	製距時間
該訂單的顧客	交期滿足
該訂單顧客的潛在訂單	現場使用率/負荷
該訂單顧客歷史交易	訂單延遲數目
訂單的利潤	延遲時間
	等待時間

本研究試圖將定性與定量兩種因素納入排程規劃。研究中在排程的規劃階段，求解過程中為了能加強系統的效率，研究中將以遺傳演算法進行整體搜尋(global search)求解外，將利用禁忌搜尋法做為局部修正(local improvement)，藉由制定多目標適應函數(multi-objective fitness function)來處理定量方面的因素。關於定性因素方面，採用以訂單為導向的方式來考量，每一訂單依照所需考量的因素來評估，建立一訂單優先順序。再將此訂單優先順序轉為一懲罰函數納入目標函數中，做為遺傳演算法運作時的限制。

零工式生產系統的特色是由一群具有相同功能的機器設備組成工作中心(work center)或部門(department)，訂單在不同工作中心或不同部門間流動，且在各工作中心或部門內進行不同的加工程序(operation)。每項工作的製造程序具體指定於這些機器設備上，也就是製造系統的途程（routing）



安排，在本研究中對於每項工作而言每一途程皆定義為唯一的，且製造這些產品的時間假設為固定且明確的，最後每項工作的生產途程假設為已知。而這些產品的製造方式常是以接單生產 (make-to-order) 的方式進行，以反映產品在市場上的快速變化並具生產彈性。

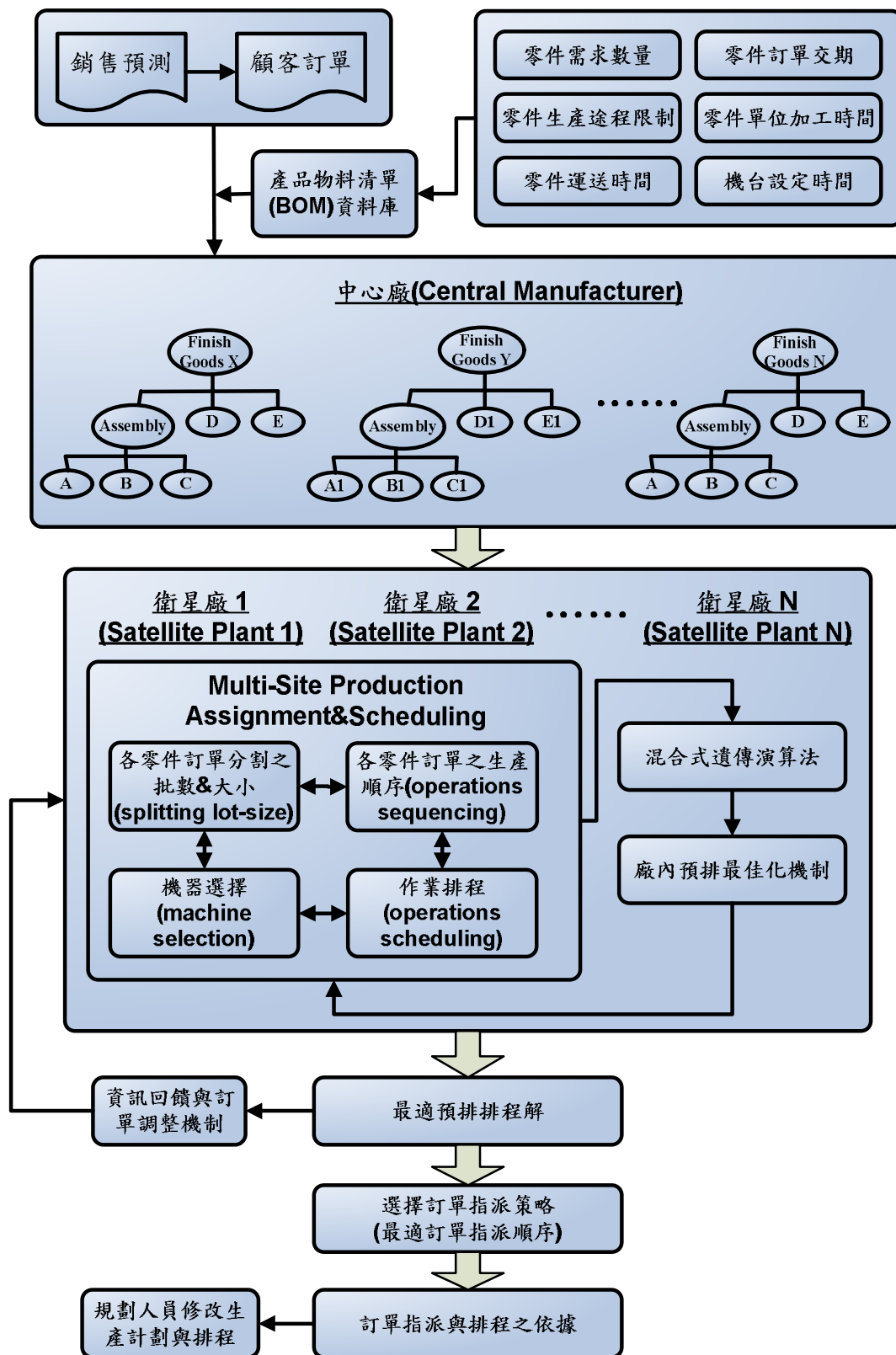


圖 3.2 系統架構圖

本研究試圖將定性與定量兩種因素同時納入多廠區排程規劃，並模式化具有多廠機器生產型態的 Job shop 類型排程問題。目前在排程問題中，機台排程問題是困難度極高的組合最佳化 NP-hard 問題，隨著問題的規模與複雜度的增加，若欲使用最佳化方法求解，遺傳演算法是發展中優異的最佳化方法。所以研究中將以遺傳演算法與混合式遺傳演算法分別進行搜尋求解，藉由制定多目標適應函數(multi-objective fitness function)來處理定量方面的因素。關於定性因素方面，採用以訂單為導向的方式來考量，每一訂單依照所需考量的因素來評估，建立一訂單優先順序。再將此訂單優先順序轉為一懲罰函數納入目標函數中，做為遺傳演算法與混合式遺傳演算法運作時的限制。

在結合多廠的訂單指派與排程規劃的規劃當中，將依循以下步驟進行系統設計及運算機制之建構。將之分述如下：

### 3.2 多廠生產指派與排程模式架構

本研究架構一套非線性(non-linear)數學模式，模擬多廠區生產指派與排程規劃之架構，透過每代的零件訂單分割至不同衛星廠生產，造成每代零件訂單指派順序演化，達到整體系統最佳化。模式基本假設如下：

#### 符號說明

$p$ ：零件訂單編號 ( $p=1\cdots P$ )

$t$ ：生產時段 ( $t=1\cdots T$ )

$i$ ：零件訂單生產階段編號 ( $i=1\cdots I$ )

$j$ ：零件訂單生產階段之廠區編號 ( $j=1\cdots J^i$ )

$m$ ：各廠區之機台編號 ( $m=1\cdots M_{j^i}$ )

#### 參數設定

$P_{(i,j)mpt}$ ：零件訂單  $p$  在生產階段  $i$ ，廠區  $j$  之現場機台  $m$ ，生產時段  $t$  之單位作業加工時間

$S_{(i,j)mpt}$ ：零件訂單  $p$  在生產階段  $i$ ，廠區  $j$  之現場機台  $m$ ，生產時段  $t$  之作業設定時間

$Q_{(i,j)mpt}$ ：零件訂單  $p$  在生產階段  $i$ ，廠區  $j$  之現場機台  $m$ ，生產時段  $t$  之加工數量

$cap_{(i,j)t}$  : 生產階段  $i$  , 廠區  $j$  , 生產時段  $t$  之生產產能

$cap_{(i,j)m}$  : 生產階段  $i$  , 廠區  $j$  之現場機台  $m$  之單機產能 (每週)

$DD_p$  : 零件訂單  $p$  之交期

$\alpha_1$  : 機台使用率之權重值

$\alpha_2$  : 交期之權重值

$\alpha_3$  : 製距之權重值

$W_1$  : 定量因素權重值

$W_2$  : 定性因素權重值

$\alpha_1 + \alpha_2 + \alpha_3 = 1, W_1 + W_2 = 1$

### 變數設定

$L_{(i,j)p}$  : 零件訂單  $p$  在生產階段  $i$  , 廠區  $j$  之單位加工分割批量

$l_{(i,j)np}$  : 零件訂單  $p$  在生產階段  $i$  , 廠區  $j$  之批量分割數 ( $l_{(i,j)np} = \frac{Q_{(i,j)pt}}{L_{(i,j)p}}$ )

( $n=1 \dots N$ )

$T_{(i,j)mpt}$  : 零件訂單  $p$  在生產階段  $i$  , 廠區  $j$  之現場機台  $m$  , 生產時段  $t$  之開始時間

$C_{(i,j)mpt}$  : 零件訂單  $p$  在生產階段  $i$  , 廠區  $j$  之現場機台  $m$  , 生產時段  $t$  之完工時間

$U(C_{(i,j)mpt})$  : 零件訂單  $p$  在生產階段  $i$  , 廠區  $j$  之現場機台  $m$  , 生產時段  $t$  之交期滿足度

### 目標函式 (Objective function)

$$MAX. W_1 \cdot \sum_{i=1}^I \left( \alpha_1 \cdot \sum_{j=1}^{J^i} \sum_{m=1}^{M_j^i} \sum_{n=1}^N \sum_{p=1}^P \left( \frac{S_{(i,j)mpt} + (L_{(i,j)p} \cdot l_{(i,j)np} \cdot P_{(i,j)mpt})}{C_{(i,j)mpt}} \cdot \frac{1}{\sum_{m=1}^{M_j^i} m} \right) + \alpha_2 \cdot \sum_{j=1}^{J^i} \sum_{m=1}^{M_j^i} \sum_{p=1}^P \left( \frac{U(C_{(i,j)mpt})}{P} \right) + \alpha_3 \cdot \sum_{j=1}^{J^i} \sum_{m=1}^{M_j^i} \sum_{p=1}^P \left( \frac{\min\{C_{(i,j)mpt} - T_{(i,j)mpt}\}}{C_{(i,j)mpt}} \right) \right) +$$

$$W_2 \cdot (1 - p(x))$$

MAX.  $W_1 \cdot (\alpha_1 \cdot (\text{平均機台使用率}) + \alpha_2 \cdot (\text{平均訂單交期滿足率}) + \alpha_3 \cdot (\text{製距績效評估})) + W_2 \cdot (1 - \text{懲罰函數})$

## 限制條件 (Constraints)

### (1) 交期限制

$$C_{(i,j)mpt} - T_{(i,j)mpt} \leq DD_p \quad \forall i, j, m, p, t \dots\dots\dots (式 3.1)$$

### (2) 批量加工時間限制

$$\{C_{(i,j)mpt} - T_{(i,j)mpt}\}_{MAX} \geq S_{(i,j)mpt} + (T_{(i,j)mpt} + L_{(i,j)p} \cdot l_{(i,j)np} \cdot P_{(i,j)mpt}) \quad \forall i, j, m, p, t \dots (式 3.2)$$

### (3) 相同作業，不同機台 (conjunctive constraints)

$$T_{(i',j')mpt} \geq S_{(i,j)mpt} + (T_{(i,j)mpt} + L_{(i,j)p} \cdot l_{(i,j)np} \cdot P_{(i,j)mpt}) \quad \forall i, j, m, p, t \dots\dots\dots (式 3.3)$$

### (4) 不同作業，相同機台 (disjunctive constraints)

$$T_{(i,j)mpt} \geq S_{(i',j')mpt} + (T_{(i',j')mpt} + L_{(i',j')p} \cdot l_{(i',j')np} \cdot P_{(i',j')mpt}) \quad \forall i, j, m, p, t \dots\dots\dots (式 3.4)$$

### (5) 產能限制

$$S_{(i,j)mpt} + (T_{(i,j)mpt} + L_{(i,j)p} \cdot l_{(i,j)np} \cdot P_{(i,j)mpt}) \leq cap_{(i,j)m} \cdot m \quad \forall i, j, m, p, t \dots\dots\dots (式 3.5)$$

### (6) 批量限制

$$L_{(i,j)p} \cdot l_{(i,j)np} \leq Q_{(i,j)mpt} \quad \forall i, j, m, p, t \dots\dots\dots (式 3.6)$$

### (7) 非負限制式

$$T_{(i,j)mpt} \geq 0 \quad \forall i, j, m, p, t \dots\dots\dots (式 3.7)$$

## 目標函式說明

本研究所設計的適應性函數由兩個部分構成：多目標函數與懲罰函數。多目標函數是針對製造現場中的績效衡量指標；懲罰函數則是利用 AHP 法所求得質性訂單順序計算轉換而來。由於是利用遺傳演算法來搜尋出較佳的作業順序，其中每一個體皆是一組作業順序，將其轉換為訂單順序，算出其與質性訂單順序的差異作為懲罰函數。

## 限制條件說明

(式 3.1)說明零件訂單之完工時間減去開始時間要小於訂單之交期。(式 3.2) 說明零件訂單加工時，受限機台批量加工時間，必須滿足其限制條件。(式 3.3) (式 3.4)說明零件訂單作業加工上的順序限制。(式 3.5)說明零件訂單加工批量必須滿足廠區的產能限制。(式 3.6)說明批量必須滿足機台的上限。(式 3.7)其它。

### 3.3 多廠機器排程法則

為了能更貼近現實環境的生產模式，本研究允許各零件訂單有一動態的批量分割機制。在加工每一作業之前，先將各零件訂單的總批量分割成數個子批，使零件訂單能同時在多廠機器加工，藉此能有效運用多廠機器，發揮其功用。

#### 3.3.1 排程演算法

本研究考慮批量分割，如圖 3.3。欲加工的零件訂單按照其途程順序在第一個工作中心加工之前，會有一動態批量分割機制（演算法的設計中會提及）均勻地分割零件訂單的總批量，而子批數也就等於此訂單加工時在工作中心需佔有的多廠機器數，子批數上限為此工作中心擁有的多廠機器數，待所有子批均加工完畢，批量分割機制再一次分割訂單，做為下一途程加工批量大小與需佔機器數的依據，批量分割的機制在每個工作中心與工作中心之間均需執行一次，直到此訂單加工完畢。

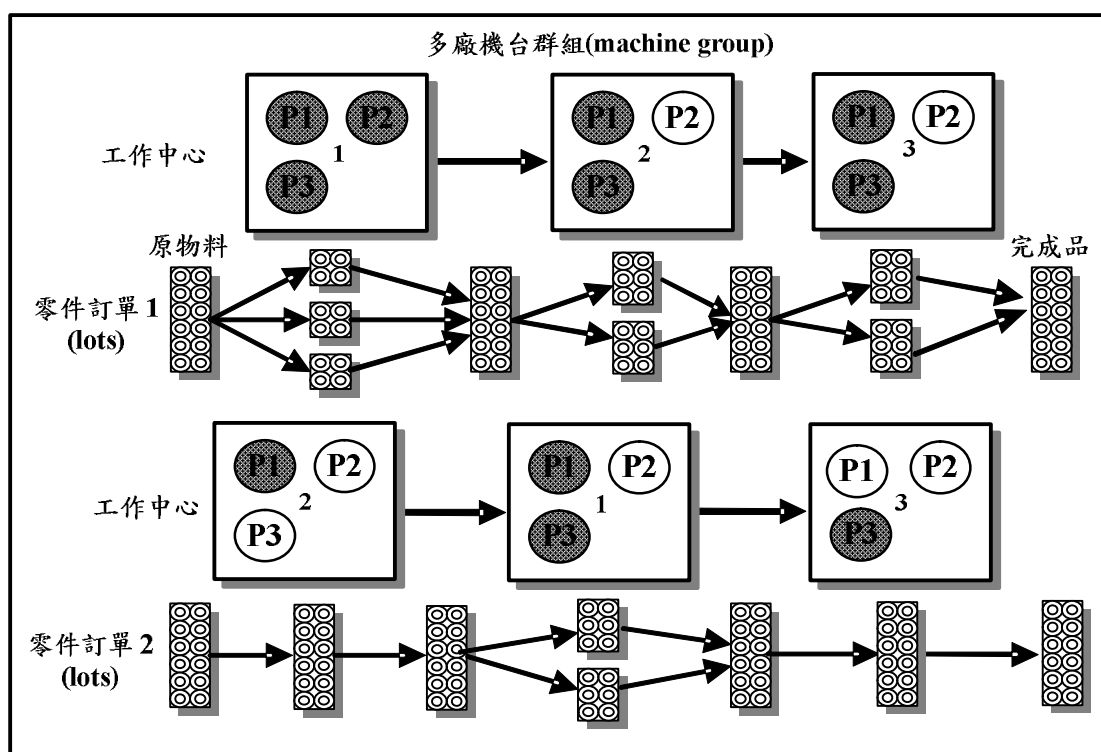


圖 3.3 多廠零工式生產系統示意圖

本研究中對於每張訂單而言，只有唯一的途程，每一作業的單位加工時間假設為固定且明確的，最後每項工作的生產途程假設為已知。這些訂單的生產方式是以接單生產 (make-to-order) 的方式進行，以反映產品在市場上的快速變化並具生產彈性。

排程演算法主要是以將產能集中的概念，以減少機器閒置時間與縮短訂單作業完成時間為目的所發展出來的。每一個作業依據由演算法所搜尋出的優先順序安排於排程計畫內，在完成實際的排程過程中，重覆進行搜尋一個最早空閒區間能夠容納欲排入之作業。傳統多階單機的排程問題，因其為單一作業獨佔機台時間，可以很容易且快速地做出判斷，但在多階多廠機器且允許批量分割的環境之下，每個批量均需對所有多廠機器進行判斷，相較於傳統情形複雜度提高許多。

本研究排程模式為零工式生產型態，並考慮為數眾多的多廠機器問題，基本假設如下：

1. 生產系統不只生產單一產品，而是生產多種不同產品。
2. 有多個工作中心，每個工作中心代表一個加工途程。
3. 當時間開始時，所有訂單皆已準備就緒。
4. 每一工作中心對每一產品有上機批量限制的下限，如果批量不足不予以開機加工。解決方法為逐一減少佔有機器的數目，直至切割出的批量大小達到下限。
5. 訂單總批量大小均超過上機批量限制。
6. 機器設置時間已包含於加工時間中。
7. 每個批量在每個工作中心只加工一次。
8. 每部機器同一時間內最多只能處理一批量。
9. 批量在加工過程中不能被中斷。
10. 在加工機器上，某些時段已有加工作業在其上加工中。
11. 需等待作業的分割批量在前一個工作中心全部加工完成後，才能到下一個工作中心進行加工，運送時間忽略不計。
12. 在具有多部機器的工作中心中，每部機器皆被視為相同機器。
13. 每個批量在每一工作中心中，只需經過其中任一部機器加工處理。

訂單經過前述批量分割機制後，便可得知欲佔有的多廠機器數（假設為  $M$  台機器）。逐一判斷所有的多廠機器，其空閒時區能否容納此批量加工時間，若否，則找尋機器中下一空閒時區，直至可被排入，最後，為了減少機器閒置時間，挑選出空閒時區最早的  $M$  台多廠機器，將批量作業分

別排入此 M 台機器排程中，作業排程流程如圖 3.4。排程演算法步驟如下：

步驟一：欲排之作業經過動態批量分割機制，將零件訂單分割為數個子批。

步驟二：判斷分割出的批量大小是否達到上機批量限制的下限。若無達到，則再執行一次步驟一。若有，則得到加工此作業需要 M 台多廠機器的資訊。

步驟三：如果欲排之批量作業是其訂單加工途程中的第一個作業，則執行步驟四，否則執行步驟八。

步驟四：若執行此批量作業之多廠加工機器目前無安排其他作業，則直接將此批量作業排入。否則執行步驟五。

步驟五：尋找多廠加工機器中的最早空閒時間區段。

步驟六：依據多廠加工機器之空閒狀態與此批量作業的開始時間，計算此批量作業的結束時間。

步驟七：判斷批量作業之結束時間與多廠加工機器之下一作業之開始時間，兩者在時間上是否有衝突，若有，則繼續尋找機器下一空閒時間區段，執行步驟六。若無衝突產生，則此批量作業完成排程。

步驟八：若執行此批量作業之多廠加工機器目前無安排其他作業，則依據途程中的前一作業的結束時間作為該批量作業的開始時間。計算此批量作業的結束時間完成此批量作業之排程。否則執行步驟九。

步驟九：以途程中的前一作業的結束時間為起點，尋找多廠加工機器中的最早空閒時間區段。

步驟十：判斷多廠加工機器之空閒時間狀態以及途程前一作業之完成時間，計算此批量作業的開始時間與批量作業的結束時間。

步驟十一：判斷批量作業的結束時間與多廠加工機器中下一作業之開始時間是否有衝突。若有，則繼續尋找機器下一空閒時間區段，執行步驟十。若無衝突產生，則執行步驟十二。

步驟十二：判斷所有的多廠加工機器是否均完成批量作業的排程。若尚有加工機器未完成批量作業排程，則尋找下一未完成排程的多廠加工機器，執行步驟三。若均完成，執行步驟十三。

步驟十三：選定批量作業開始時間最早的 M 台多廠加工機器，並將 M 台多廠加工機器中最早的批量作業開始時間作為該作業開始時

間，最晚的批量作業結束時間作為該作業結束時間，則此作業完成排程。

雖然以此種排程演算法所得的排程結果能有效地減少機器閒置時間而使得產能集中，但是相對地會造成在多部機器加工的同一直單批量加工時間誤差過大。有鑑於此，下一小節中將局部修正批量作業的加工時間。

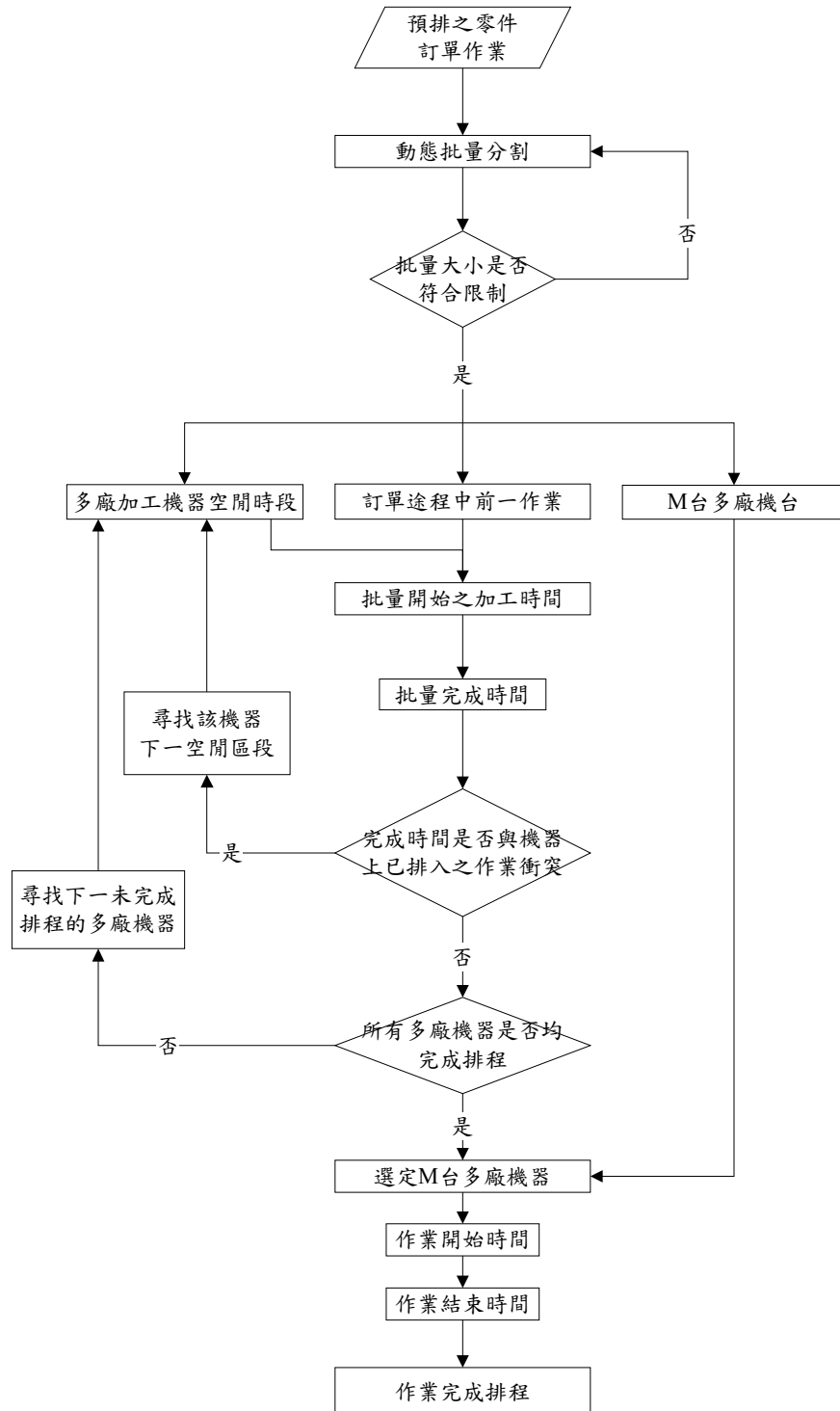


圖 3.4 作業的排程流程圖



### 3.3.2 作業時間修正模式

本研究中將生產批量採取批量分割，目的是讓各零件訂單作業可在一製程中分散至多廠機器上同時加工，使得加工作業能夠重疊，縮短總生產時間。但是上述排程演算法有可能會造成作業分割批量的加工區間相差甚遠，使得在製品(work-in-process)囤積於工作中心的時間過長，徒增許多不必要的生產成本，如圖 3.5 所示。加工某作業所需的工作中心一共有三台多廠機器 M1、M2 與 M3，而零件訂單作業欲分割成三個批量分別至不同廠機器上加工，虛線方塊代表先前在該廠機器上已排定之作業加工時間，實線方塊代表此作業所排定的加工時間。由於 M1 上已排定作業較多的關係，使得子批 1 的加工區間較子批 2 與子批 3 晚了許多，因此作業的子批 2 與子批 3 加工完畢後，需經過長時間的等待，子批 1 才能結束加工，生產批量才算全部加工完畢。

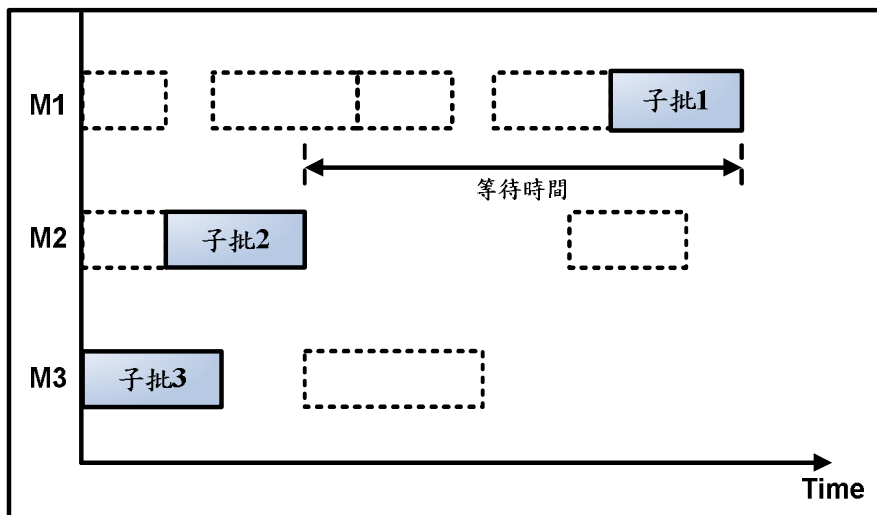


圖 3.5 原始排程甘特圖

在此例中，子批 2 與子批 3 在等待時間中均為在製品，等待時間明顯過長，使在製品管理困難，並造成同一種產品的生產不連續，間接地讓機器的操作與換模等相關步驟複雜化，這對排程來說，並不是一個最好的結果。

因此本研究在每一作業排程時間排定後，增設作業時間修正模式。以子批中最晚的加工結束時間為限制，調整各子批的加工時間，使其盡量靠近最晚的加工結束時間，讓零件訂單作業分割批量的加工區段能彼此重

疊，達到生產連續，而在製品的等待時間也能減到最低。作業修正流程如圖 3.6。修正模式的步驟如下：

- 步驟一：如果欲修正之子批是作業中加工結束時間最晚的子批，則子批的作業時間無需調整，執行步驟八。否則執行步驟二。
- 步驟二：往後尋找多廠加工機器中的下一空閒時間區段，將子批作業時間往後遞移。
- 步驟三：依據多廠加工機器之空閒狀態，計算修正後此子批作業的開始與結束時間。
- 步驟四：如果子批作業之結束時間沒有超過子批中最晚的加工結束時間，執行步驟七。否則執行步驟五。
- 步驟五：往前尋找多廠加工機器中的上一空閒時間區段，將子批作業時間往前遞移。
- 步驟六：依據多廠加工機器之空閒狀態，計算修正後此子批作業的開始與結束時間。執行步驟八
- 步驟七：如果子批作業不是現階段機器上已排入作業中的最後一個作業，則繼續尋找機器下一空閒時間區段，執行步驟二。否則執行步驟八。
- 步驟八：判斷所有子批是否均經由修正模式調整完成，若尚有子批未調整，則尋找下一未調整子批，執行步驟一。否則所有子批的加工時間均調整完畢，則此作業完成修正。

上例中作業時間修正結果如圖 3.7 所示。以子批 1 的加工結束時間為限制，將子批 2 與子批 3 的作業加工時間往後遞移，使各子批間的作業時間重疊，工作中心能連續生產同一類產品與減少在製品囤積的時間。換句話說，經由修正模式，子批 2 與子批 3 可不必太早在機器上進行加工；相較於原來無修正的情形來說，在機器上空出的較早閒置產能尚可被其餘未排入加工之作業利用，使其能提早開始加工，達成交期準確的目標。

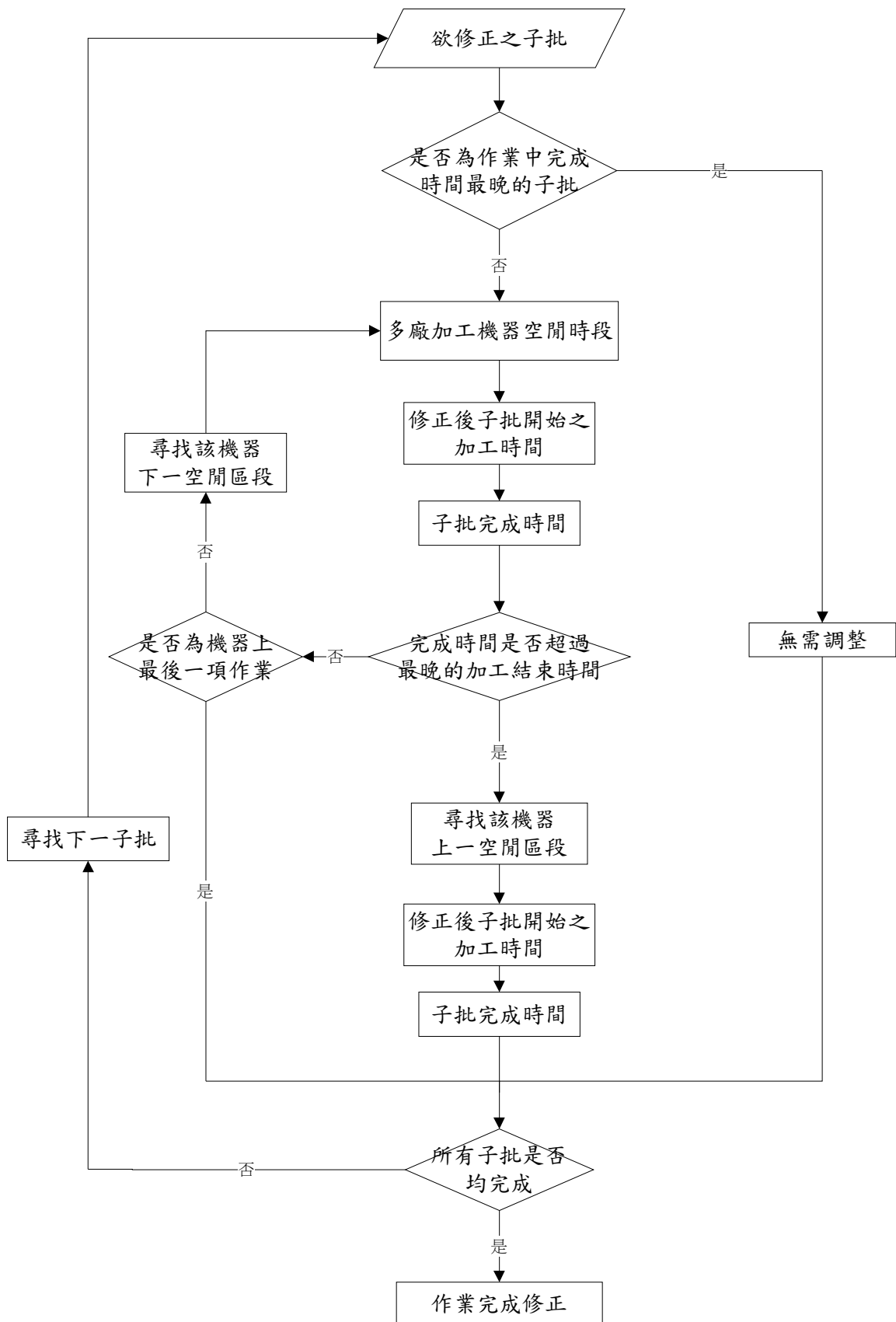


圖 3.6 作業的時間修正流程圖

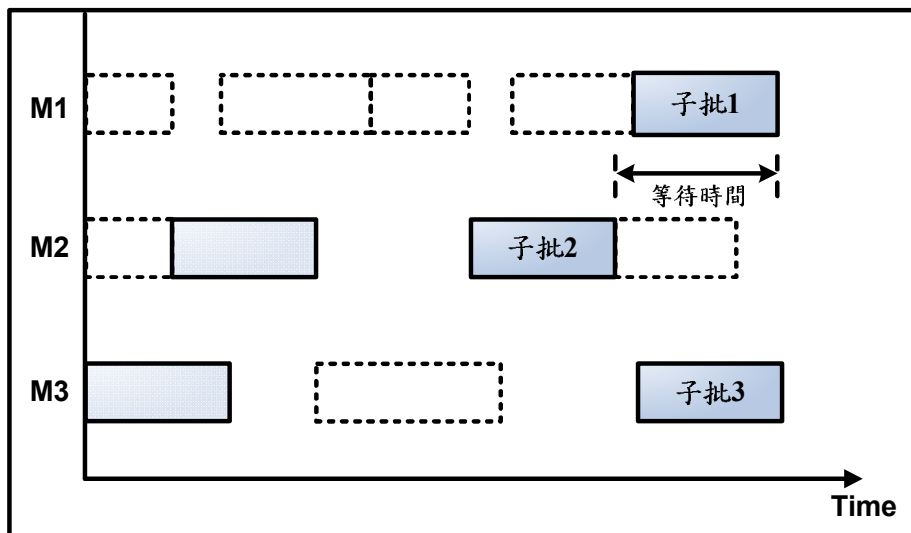


圖 3.7 修正排程甘特圖

### 3.4 定性因素模式架構

本研究中所指之定性因素乃是不能明確加以量化或是不需精確數量化的因素。以市場因素而言，該張訂單產品在市場中的定位將是主要關鍵，假若該產品的市場競爭者眾多，則產品完工的時效性顯的十分重要，因為其能使產品有更好的競爭優勢。換句話說，此訂單在市場因素之下應該擁有較優先的製造順序，但是此因素並不能直接加以量化。

訂單所屬的顧客也是屬於無法明確量化的定性因素，對公司而言，不同顧客的重要性必然會有程度上的差異，可能是因為彼此間有策略聯盟或是其它策略上的因素所造成。同樣地，每個顧客未來潛在的訂單利潤，也應納入此階段中應考量的因素之一。

此外，本研究中的定性因素包含不須被明確量化的因素，例如顧客以往的交易記錄與訂單的利潤。我們希望排程規劃人員在評估訂單製造的優先順序時，根據這些因素資料做主觀性的比較判斷。本研究中將利用 AHP 法來評估訂單在各個定性因素下的重要性，進而計算出訂單在定性因素下的優先順序。以下說明研究中 AHP 法的評估計算。

### 3.4.1 AHP 法的評估計算

關於如何設定各因素權重值的方法有很多，本研究中將利用階層式分析程序法來訂定權重值。AHP 法是由 Saaty[61]所提出，是用來評估方案或是因素間相對權重，其原理是採用配對比較法(pairwise comparison approach)來設定因素間的相對權重，將評估因素的重要性以配對的方式兩兩比較，並給予比較之分數，再加以計算各因素的權重值。

AHP 法最大的優點是在於當人們面臨多個評估因素情形下，提供決策者一客觀的方法來決定各因素的權重。同時，AHP 也可以應用於多個方案的選擇評估。AHP 法中的評估尺度如表 3.2。

表 3.2 AHP 評估尺度

評估尺度	定義
1	同等重要
3	稍重要
5	重要
7	極重要
9	絕對重要
2、4、6、8	相鄰尺度的中間值
以上數值之倒數	如上定義之相對不重要程度

舉例說明如下：假設目前有 5 張訂單，而所需考量的因素有 4 個( $F_1$ 、 $F_2$ 、 $F_3$ 、 $F_4$ )。

步驟一：將所有因素配對以表中的評估尺度來做比較。(若規劃人員評估  $F_1$  較  $F_2$  稍不重要，評估尺度為 1/3，相反的  $F_2$  較  $F_1$  稍重要，評估尺度為 3。)

步驟二：計算此評估尺度矩陣的特徵值(eigenvalue)，取其最大的特徵值( $\lambda_{\max}$ )。

步驟三：計算  $\lambda_{\max}$  對應的特徵向量(eigenvector)。

步驟四：將此特徵向量正規化(normalize)，此正規化特徵向量即為各個因素相對的權重值( $W_i$ )，如表 3.3。

表 3.3 各因素之權重制訂

因素	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	特徵值 $\lambda_{\max}=4.11$	權重 W <sub>i</sub>
F <sub>1</sub>	1	1/3	2	7	0.3773	0.24
F <sub>2</sub>	3	1	5	9	0.8976	0.58
F <sub>3</sub>	1/2	1/5	1	5	0.2192	0.14
F <sub>4</sub>	1/7	1/9	1/5	1	0.0628	0.04

步驟五：規劃人員在因素 F<sub>i</sub> 的考量下，將所有訂單配對比較評估，利用步驟一至步驟四的計算方式，求出一正規化特徵向量，此即為每一張訂單在因素 F<sub>i</sub> 下的評估值(E<sub>ij</sub>)，如表 3.4。其中 E<sub>ij</sub> 表示訂單 j 在因素 i 的評估值。

表 3.4 各方案之評估值

F <sub>i</sub>	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	特徵值 $\lambda_{\max}=5.0938$	評估值 E <sub>ij</sub>
訂單 1	1	5	2	3	3	0.7758	0.4055
訂單 2	1/5	1	1/3	1/2	1/3	0.1277	0.0668
訂單 3	1/2	3	1	3	2	0.4912	0.2568
訂單 4	1/3	2	1/3	1	1/2	0.203	0.1061
訂單 5	1/3	3	1/2	2	1	0.3153	0.1648

步驟六：計算出所有的評估值(E<sub>ij</sub>)。

步驟七：計算各訂單的加權值，並加以排序，如表 3.5。

表 3.5 各方案之加權排序

F <sub>i</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	加權值	排序
W <sub>i</sub>	0.24	0.58	0.14	0.04		
評估值	E <sub>1j</sub>	E <sub>2j</sub>	E <sub>3j</sub>	E <sub>4j</sub>		
訂單 1	0.4055	0.30	0.10	0.15	0.2913	1
訂單 2	0.0668	0.20	0.25	0.25	0.1770	3
訂單 3	0.2568	0.20	0.30	0.15	0.2256	2
訂單 4	0.1061	0.15	0.15	0.35	0.1745	4
訂單 5	0.1648	0.15	0.20	0.10	0.1586	5

此加總值排序順序就是所有訂單在定性因素考量下的優先順序。本階段利用 AHP 法評估每個訂單在質性因素下的重要性，亦即利用 AHP 所求得的質性訂單順序，作為遺傳演算法與混合式遺傳演算法中的限制。

### 3.5 定量因素模式架構

在定量因素的考量方面，將所有的定量因素轉換為單一多目標函數，各目標間的權重藉由 AHP 法則來制定。在本研究演算法搜尋求解的過程中，不單是利用一項指標來評估一個體解的優劣，相對的是強調如何在多個相衝突的目標中折衷求得一最適解，為了能正確利用經由 AHP 法則制定的權重值，在適應函數的設計中每一項目標評估值都應被正規化後再加權計算。本研究在定量因素方面將考量排程製距、訂單的交期滿足度及機器使用率三項衡量指標。

#### 3.5.1 製距績效評估

製距的評估方式為第一張訂單之起始作業到最後一張訂單的最後一個作業結束所需的時間。首先由個體解排程結果求出此次排程製距，再將製距評估值正規化，製距評估值正規化的計算方式如式 3.8

$$\frac{\min MS}{MS_x} \quad \text{式 (3.8)}$$

其中，

$\min MS$ ：到目前為止，搜尋過程中最小的製距

$MS_x$ ：個體解  $x$  的製距評估值

在搜尋求解的過程中，當找到一個體解的製距比  $\min MS$  小，則  $\min MS$  就會被取代，以上述方式來正規化製距評估值會造成同一個體在不同族代的適應函數值不相同的現象發生。舉例來說，在第  $n$  代中， $\min MS$  等於 80 且某一個體的製距為 100，製距評估值正規化後為  $80/100=0.8$ ，而如果在第  $(n+1)$  代，搜尋到一新個體的製距為 70，若前一個體有被保留至  $(n+1)$  代的族群中，則此個體在  $(n+1)$  代中的正規化製距評估值等於  $70/100=0.7$ ，會造成其適應函數值降低的現象。

雖然在演算的過程中，會發生同一個體在不同代中有不同的適應函數值。但是以相對的角度而言，在同一代中的個體適應函數值高者恆高，低者恆低，對於演算法中的機制不會造成明顯的影響。

### 3.5.2 交期滿足度評估

本研究中將交期定義於模糊集合上，如圖 3.8。個體解的交期滿足度評估方式為在該次排程結果中，先將每張訂單完成時間利用模糊隸屬函數算出交期滿足度，再將該次排程中所有訂單的交期滿足度取平均值即為所求。公式如下所示：

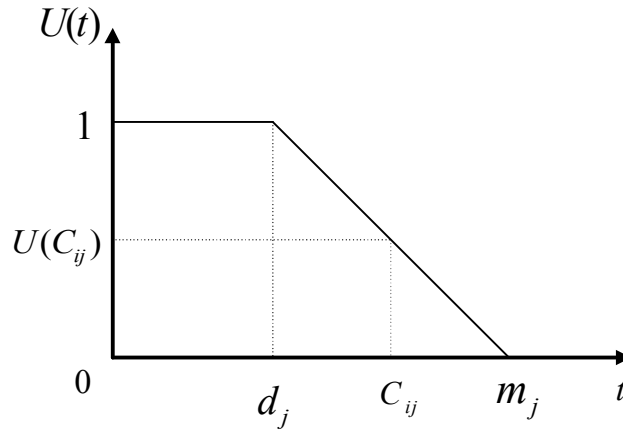


圖 3.8 交期之模糊隸屬函數

$$\left\{ \begin{array}{ll} U(C_{ij}) = 1 & , \quad C_{ij} \leq d_j \\ U(C_{ij}) = \frac{m_j - C_{ij}}{m_j - d_j} & , \quad d_j \leq C_{ij} \leq m_j \\ U(C_{ij}) = 0 & , \quad m_j \leq C_{ij} \end{array} \right. \quad \text{式 (3.9)}$$

$$D_i = \frac{\sum_{j=1}^n U(C_{ij})}{n} \quad \text{式 (3.10)}$$

其中，

$n$ ：該次排程之訂單數目

$U(C_{ij})$ ：個體解  $i$  的第  $j$  張訂單之交期滿足度

$C_{ij}$ ：個體解  $i$  的第  $j$  張訂單完成時間

$d_j$ ：訂單  $j$  之限定交期

$m_j$ ：訂單  $j$  之交期上限

$D_i$ ：個體解  $i$  之交期評估值

### 3.5.3 機器使用率評估

排程規劃時除了製距、交期的評估之外，機器使用率也是常見的績效衡量指標。關於機器使用率的評估方式是先個別加總每一機器上每項作業



的加工時間，再除以機器上最後一項作業的完成時間，可得到每一機器個別之使用率，再將所有機器使用率取平均值即為所求。

### 3.6 遺傳演算法之運作

本節將說明研究中關於遺傳演算法的設計及其運算過程。

#### 3.6.1 編碼表示法

本研究中遺傳演算法個體分為上半部與下半部基因，編碼方式如下：

1. 基因數目等於所有訂單作業數總和的兩倍。
2. 個體中上半部的基因，每一基因代表一作業的排程的優先順序，而基因的排列是先依照訂單再依照訂單本身的作業順序。也就是說第一個基因為零件訂單一的第一個作業，第二個基因為零件訂單一的第二個作業，直至零件訂單一的最後一個作業；而其下一個基因即為零件訂單二的第一個作業，依此類推。
3. 下半部基因的數目，與上半部相等，每一基因代表一百分比，也就是一作業在其加工途程所指定的工作中心中佔有的多廠機器數的百分比。而基因的排列方式與上半部基因相同。上下半部的基因兩兩相對，也就是一作業對應了兩個基因。編碼的示意圖如圖 3.9。

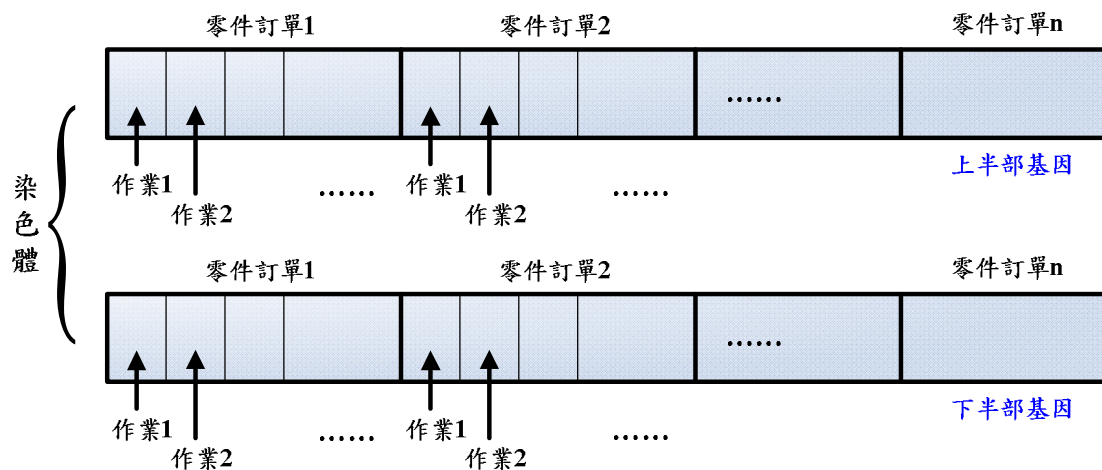


圖 3.9 編碼表示法示意圖

#### 3.6.2 初始族群的產生

為了能讓系統在起始搜尋時，對於每一狀態空間(state space)都有同等機會，本研究將採用隨機的方式產生初始族群。假設一共有  $N$  個作業，亦即每一個體有  $2N$  個基因，對於上半部基因來說，初始族群產生方式是隨機產生一個  $1 \sim N$  不重複的數字串列個體的起始基因，如圖 3.10。

	作業 1	作業 2	...	作業 m
訂單 1				
訂單 2				
...				
訂單 n				

將 1~(n × m) 的數字  
以隨機方式填入

圖 3.10 上半部基因初始族群示意圖

但是，以這種方式產生初始族群會有不理解的現象發生，例如，訂單 1 有 5 個作業，而其基因分別為 [5, 8, 6, 10, 12]，我們發現訂單 1 中的作業 3 因為加工途程的限制必須在作業 2 之後方可開始加工，但作業基因優先於作業 2，此個體為一不理解的，關於這一點，我們將藉由一啟發式修正的方式將其合理化。修正方式是將不理解的兩個基因做交換。在上例中，我們將 8 與 6 的交換之後的個體即為一理解。

下半部基因代表的是百分比數值，設計上給予 1~10 十個數字，1 代表佔有工作中心 10% 的機器數，2 代表佔有工作中心 20% 的機器數，依此類推。初始族群產生方式為在每一基因上隨機產生 1~10 中的一個數字，如圖 3.11。

	作業 1	作業 2	...	作業 m
訂單 1				
訂單 2				
...				
訂單 n				

隨機方式填入 1~10  
中的一個數字

圖 3.11 下半部基因初始族群示意圖

在下半部基因的設計上，產生一動態批量分割機制。佔有的機器數多寡便決定了訂單作業分割批量的大小，佔有的機器數愈多，批量愈小，相對的佔有的機器數愈少，批量愈大。因本研究考慮上機批量限制的關係，上述產生初始族群的方法也會有不理解的現象。例如，訂單 1 的總批量為 120，其作業 1 指定的工作中心有十台機器，上機批量限制為 35，而作業 1 的基因為 [4]，代表作業 1 需佔有工作中心 40% 的加工機器，也就是 4 台機器，所以每台機器上的加工批量為  $120/4=30$ ，沒有達到上機批量限制，此個體為一不理解的，修正的方式是將基因碼逐次減 1，直到達到上機批

量。上例中，將 4 減 1 得到基因碼為 3，加工批量變為 40，此個體即為一合理解。

### 3.6.3 適應性函數之設計

本研究所設計的適應性函數由兩個部分構成：多目標函數與懲罰函數。多目標函數是針對製造現場中的績效衡量指標，懲罰函數則是利用 AHP 法所求得質性訂單順序計算轉換而來。由於是利用遺傳演算法來搜尋出較佳的作業順序，其中每一個體皆是一組作業順序，將其轉換為訂單順序，算出其與質性訂單順序的差異作為懲罰函數。

#### 懲罰函數之制訂

在遺傳演算法運作的過程中，我們是將第 3.4 節中所計算出之定性因素下訂單順序作為限制，演算所得的個體基因再與此訂單順序比較，研究中以懲罰函數來處理違反定性因素訂單順序限制的情形。首先，我們必須求出個體解中訂單的順序，計算方式是由個體的基因順序算出每張訂單平均的基因順序，將其排序。懲罰函數的計算方式是加總排序後的訂單順序與定性因素訂單順序之間的總差異平方和，再將此懲罰值正規化，將懲罰值除以可能的最大差異平方和。正規化的計算如式 3.11：

(i) 當訂單數  $n$  為偶數時，其可能的最大差異平方和是 式 (3.11)

$$2[(n-1)^2 + (n-3)^2 + \dots + 3^2 + 1^2]$$

(ii) 當訂單數  $n$  為奇數時，其可能最大差異平方和是

$$2[(n-1)^2 + (n-3)^2 + \dots + 4^2 + 2^2]。$$

例如當訂單數  $n=7$  時，且其定性因素訂單順序為：訂單 1 5 3 4 7  
2 6

其有最大的差異平方和的順序是：訂單 6 2 7 4 3 5 1

可能最大差異平方和等於  $2[6^2 + 4^2 + 2^2] = 56$

以 5 張訂單舉例說明如下：假設這些訂單透過 AHP 得到在質性因素的訂單順序為：( 訂單 3 ) → ( 訂單 5 ) → ( 訂單 2 ) → ( 訂單 1 ) → ( 訂單 4 )，而由遺傳演算法所得的個體解中的基因與訂單順序的計算如表 3.6，懲罰函數之計算如表 3.7。

表 3.6 訂單平均基因順序

	作業 1	作業 2	作業 3	作業 4	作業 5	加總	平均基因順序	排序
訂單 1	1	6	11	16	21	55	11	1
訂單 2	2	7	12	17	22	60	12	2
訂單 3	3	8	13	18	23	65	13	3
訂單 4	4	9	14	19	24	70	14	4
訂單 5	5	10	15	20	25	75	15	5

表 3.7 懲罰函數之計算

位置	1	2	3	4	5	總差異量
個體解中的順序	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	
質性因素順序	訂單 3	訂單 5	訂單 2	訂單 1	訂單 4	
差異量	$(-3)^2$	$(-1)^2$	$2^2$	$(-1)^2$	$3^2$	24

然後將總差異量正規化

$$24 / 2(4^2+2^2)=0.6$$

而為了能正確的利用由 AHP 法所計算出的定性因素與定量因素權重，所以整個適應函數值計算如式 3.12

$$f(x) = W_1 \cdot (\sum \alpha_i \cdot f_i) + W_2 \cdot (1 - p(x)) \quad \text{式 (3.12)}$$

其中，

$f(x)$ ：個體解  $x$  的適應函數值。

$p(x)$ ：個體解  $x$  的懲罰函數值。

$W_1$ ：定量因素的權重。

$W_2$ ：定性因素的權重。

$\alpha_i$ ：定量因素  $i$  的權重。

$f_i$ ：定量因素  $i$  的評估值。

$W_1 + W_2 = 1$ ， $\sum \alpha_i = 1$ 。

### 3.6.4 交配運算子設計

由於染色體中包含了零件訂單之作業排程的優先順序與在其加工途程所指定的工作中心中佔有的多廠機器數的百分比兩項資訊，因此一般常用排序問題之交配運算子已無法使用，本研究引用 [62] 所提出的

MCUOX(Multi-Component Uniform Order-Based Crossover Operator)交配方法並加以改良，使遺傳演算法運用於多廠區排程模式的搜尋求解能更有效率，交配的步驟如下：

步驟一：在育種選擇策略選出的母體中，隨機選擇二條染色體作為父代。

步驟二：隨機選取此二父代染色體其中之一。

步驟三：由上半部基因的第一個基因開始搜尋，將被選取的父代中還未被指派至子代的作業排程優先順序數字，填入子代染色體空白的上半部基因中。

步驟四：從該優先順序數字在二條父代染色體中所對應的下半部基因百分比數字，隨機選擇一個作為此作業佔有的多廠機器數百分比，填入相同位置的下半部基因中。

步驟五：重複步驟二～四直到所有子代染色體中的基因被填滿數字，子代染色體便產生。

舉例來說，如圖 3.12。假設從育種選擇選出的母體中，隨機挑出了二條染色體，指定其為父代 1 與父代 2，隨機選擇一父代，假設選到父代 1，然後從父代 1 上半部基因的第一個基因開始，將未被指派的數字(數字 4)，填入子代空白的上半部基因(第一位置)中，然後隨機選擇該數字在二父代中對應的下半部基因百分比數字(數字 1 與 7)，得到此作業的下半部基因數字為 7；重複上述步驟，假設第二次選到父代 2，從父代 2 上半部基因中，將未被指派的數字(數字 2)，填入子代空白的上半部基因(第二位置)中，隨機選擇數字 2 在二父代中對應的下半部基因百分比數字(數字 2 與 6)，得到此作業的下半部基因數字為 6；重覆上述步驟直到所有基因填滿數字，子代染色體便產生。

### 3.6.5 突變運算子設定

在突變運算子的設計上是以考量每一基因為主，不同於以往大部分的研究著重於考慮每一染色體的突變機率，本研究對每一基因均給予突變機率來決定是否需要進行突變，希望藉此能拓展遺傳演算法的搜尋空間，使其搜尋能更有效率；另外由於多廠區排程問題的特殊型態，需考量作業排程的優先順序與其佔有的多廠機器數目，因此本研究採取以下突變方法。

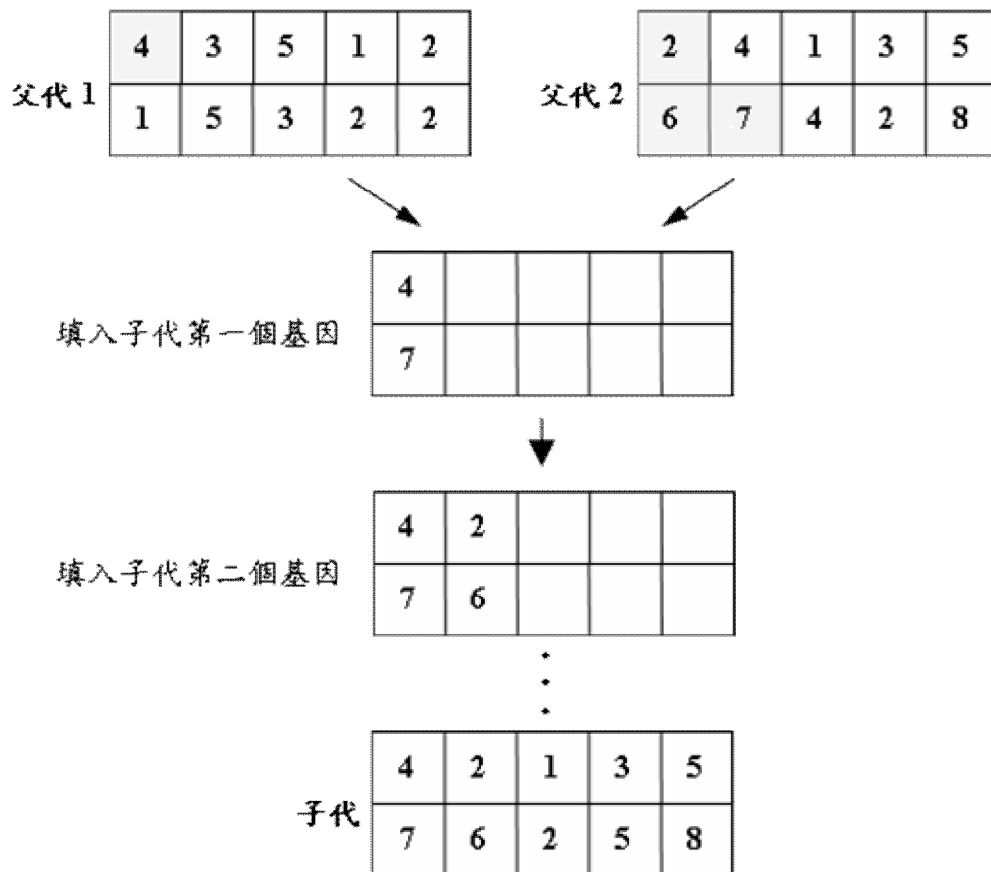


圖 3.12 交配示意圖

新產生的子代染色體中的上半部基因，若依突變機率決定要突變，則隨機選擇同一染色體中的另一上半部基因，二基因中的數字互換，而在此二基因對應的下半部基因中，則重新隨機選擇百分比數字填入，若選到與原先相同的數字，一樣將此數字填入下半部基因中。

下圖所示即為此染色體上半部基因中的第二位置依突變機率決定要突變，再隨機選擇第四位置，兩位置的上半部基因數字交換，代表第二位置的作業與第四位置的作業排程順序互換，而其對應的下半部基因位置的數字也重新隨機產生，由 5,2 變為 7,4，代表作業佔有的多廠機器數目也重新選擇。

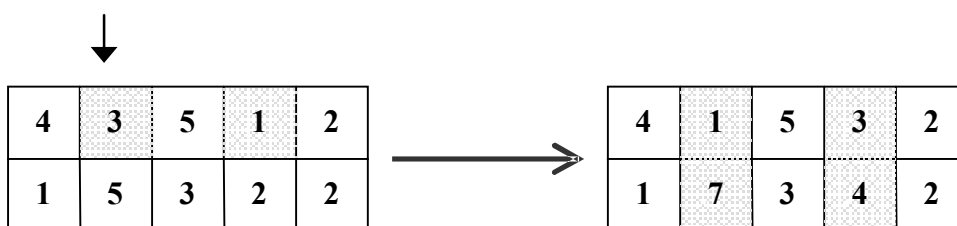


圖 3.13 突變示意圖

### 3.6.6 育種選擇

在遺傳演算法中，選取個體來產生下一子代，通常是藉由輪盤法 (roulette wheel) 來進行此一機制。而關於輪盤法中每個槽 (slot) 大小的設計方式一般常用的是直接以個體的適應函數值來設計，如式 3.13

$$P(x_i) = \frac{f(x_i)}{\sum_i f(x_i)} \quad \text{式 (3.13)}$$

其中，

$P(x_i)$ ：個體  $x_i$  被選中的機率。

$f(x_i)$ ：個體  $x_i$  的適應函數值。

但是，以這種方式來執行選擇機制，隨著族群演化，族群中個體的適應函數值逐漸提高，而且彼此之間的差距愈趨減小，如此一來，將會造成最佳解與最差解被選中的機率接近 1：1，使得整個演算過程中族群的適應函數值發生停滯。

基於上述，本研究在育種選擇機制上採用另一種方式，以競賽法 (tournament selection) 當作育種的機制，在每一代的演化過程中，首先隨機選取兩個或更多個物種，具有最大適應函數值的物種即被選中送至交配池中。由於競賽法所需的計算量較少，且可以藉由一次選取物種個數的多寡來控制競爭的速度，因此採用競爭式選擇。

一般利用遺傳演算法求解多目標問題，在育種選擇的機制中通常會採用保留精華的策略 (elite preserve strategy)。所謂精華保留策略是指在產生下一族群時，除了保留適應函數值較高者 (總體目標)，另外再保留各單一目標評估值較佳的個體解至下一代，藉此讓族群中保留有各種目標的優良個體，再透過交配運算子來產生有更好適應函數值的個體解。基於此一點，本研究中也將採用精華保留策略，讓系統的搜尋過程更有效率。

### 3.7 混合式遺傳演算法(Hybrid Genetic Algorithm)之設計

傳統遺傳演算法是透過編碼將問題的解答以基因形式來表達，再依據生物演化「適者生存」的原理進行交配與突變等運算；而禁忌搜尋法則主要利用彈性記憶（禁忌串列）讓搜尋過程有跳離區域最佳解的機會。在遺傳演算法與禁忌搜尋法各有其求解的獨特優點之下，本研究將以混合式遺傳演算法（傳統遺傳演算法結合禁忌搜尋法），來求解預排階段的排程解，其中是以傳統 GA 先求出一個解，再由禁忌搜尋法來進行鄰域搜尋，然後將「最佳解」、「次佳解」以及「最差解」進行禁忌搜尋，除了讓最佳解有搜尋到更佳解的機會，最差解也有跳脫區域的機會。

本研究將採用短期記憶結構的禁忌搜尋法對之前遺傳演算法所得的排程解進行鄰域搜尋，以提升系統搜尋的效率，執行禁忌搜尋時所使用的目標函數與遺傳演算法中的適應性函數相同，關於禁忌搜尋的步驟如下。

步驟一：由個體 I 中每一訂單以隨機的方式抽選一作業來進行禁忌搜尋，被抽選到的作業順序元素為 S。

步驟二：設定禁忌串列的大小為 M 及搜尋次數 Z。

設定最佳排程順序  $S^*$  為目前個體的作業順序。

設定最佳目標函數  $G(S^*)=G(I)$ 。

設定  $S_1=S$ 。

設定  $z=1$ 。

步驟三：搜尋  $S_z$  的鄰域， $S^1, S^2, \dots, S^{N-1}$ ，將  $S^1, S^2, \dots, S^{N-1}$ ，回填至個體 I 中，檢驗其是否有違反順序限制，若有則進行調整，否則並計算其目標函數值  $G(S^1), G(S^2), \dots, G(S^{N-1})$ ；N 為排列順序中的元素個數。

步驟四：從  $G(S^1), G(S^2), \dots, G(S^{N-1})$  選取不屬於禁忌串列中之元素對的最佳目標函數值  $G(S_z^*)$ 。如果  $G(S^*) < G(S_z^*)$ ，則令  $S^* = S_z^*$ 。

步驟五：以 FIFO 的方式更新禁忌串列； $z = z+1$ ，如果  $z = Z$  則停止。

步驟六：將最初的個體 S 以  $S^*$  替代，進行下一族群的遺傳演算法運作。

舉例說明：假設問題為  $5 \times 5$  的 Job shop 排程問題，而目前染色體基因 I，如表 3.8，其適應性函數  $G(I)=10$ 。



表 3.8 各訂單作業之基因

	作業 1	作業 2	作業 3	作業 4	作業 5
訂單 1	1	6	11	16	21
訂單 2	2	7	12	17	22
訂單 3	3	8	13	18	23
訂單 4	4	9	14	19	24
訂單 5	5	10	15	20	25

\*灰色表示被選中者

1. 訂單 1~5 被選中的作業之順序元素

$$S = [6, 12, 3, 14, 20] = S_1$$

2. 搜尋  $S_1$  的鄰域，並將其回填至原個體 I 中，計算其目標函數值。

設定搜尋次數  $Z = 7$ ，禁忌串列  $M = 3$ ， $G(S^*) = 10$

當  $z = 1$

表 3.9  $S_1$  的鄰域

鄰域解	12, 6, 3, 14, 20	6, 3, 12, 14, 20	6, 12, 14, 3, 20	6, 12, 3, 20, 14
目標函數	9.5*	8	8	6

此時，禁忌串列 =  $\{(6, 12)\}$

當將鄰域解(12, 6, 3, 14, 20)回傳至原個體時，訂單 1、2 的作業中的順序分別為： $(1, 12, 11, 16, 21)$ 與  $(2, 7, 6, 17, 22)$ 皆違反訂單加工途程的限制必需將其調整為： $(1, 11, 12, 16, 21)$ 與  $(2, 6, 7, 17, 22)$

1. 當  $z = 2$

表 3.10  $S_2$  的鄰域

鄰域解	6, 12, 3, 14, 20	12, 3, 6, 14, 20	12, 6, 14, 3, 20	12, 6, 3, 20, 14
目標函數	10	9.8*	9	8

雖然鄰域解 1 的目標函數最高，但其是由(6, 12)互換所得，而(6, 12)是目前禁忌串列中的元素對，不能選取，所以選取第 2 個鄰域解，此時，禁忌串列為  $\{(6, 12), (3, 6)\}$ 。

2. 當  $z = 3$

表 3. 11  $S_3$  的鄰域

鄰域解	3, 12, 6, 14, 20	12, 6, 3, 14, 20	12, 3, 14, 6, 20	12, 3, 6, 20, 14
目標函數	9.8*	9.5	7	9

此時禁忌串列為  $\{(6, 12), (3, 6), (12, 3)\}$ ，而因為此鄰域解的目標函數  $12 > 10$  (原個體的適應性函數)，所以原個體將會被替換。

3. 當  $z=4$

表 3. 12  $S_4$  的鄰域

鄰域解	12, 3, 6, 14, 20	3, 6, 12, 14, 20	3, 12, 14, 6, 20	3, 12, 6, 20, 14
目標函數	9.8	9	8	11*

此時禁忌串列為  $\{(3, 6), (12, 3), (14, 20)\}$ ，持續進行至  $z=7$ 。

結合禁忌搜尋後的混合式遺傳演算法運作流程如圖 3.14。本節中混合式遺傳演算法的設計理念，主要是藉由傳統 GA 交配、突變及育種選擇等運算子，使得求解過程可以搜尋出與原族群差異較大的個體，讓系統保有開發(explore)的功能。另一方面，當每獲得一排程解時，利用 TS 搜尋在此基因型態下的鄰域解，提供系統有精鍊(refine)的功能。

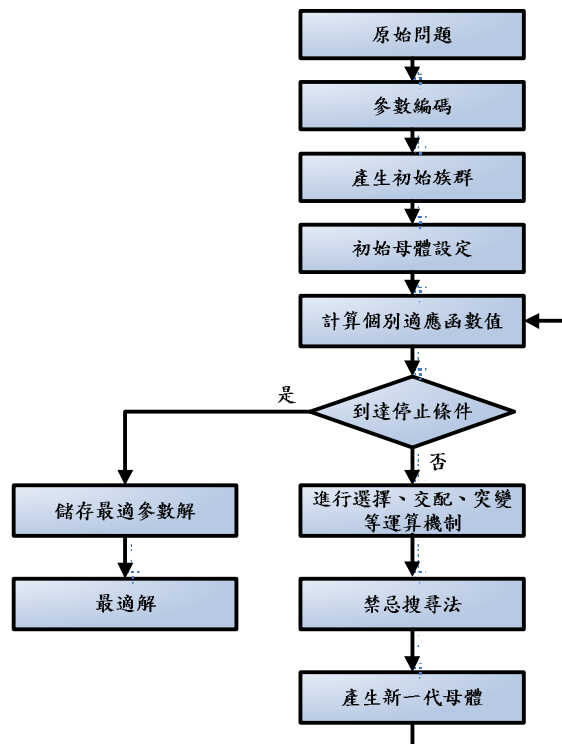


圖 3. 14 混合式 GA 運作流程圖

## 第四章 系統實證

在實證當中，本章在廠內的生產方式將以 Job Shop 排程問題為例，依據第三章所述的系統架構來進行驗證。首先於 4.1 小節中說明產品訂單模組之建立；4.2 小節則以實際訂單為例，對第三章中所探討的各項機制之設計進行探討。

### 4.1 相關基本資料之設立

在多廠規劃當中，本研究假設包含總廠及三家衛星廠製造廠，總廠並不具有生產能力，負責接收外部訂單、訂單指派與訂單修正等功能，並且擁有制訂最後生產決策的能力。

對於製造廠而言，假設有三間製造廠。在產能方面，因為三家製造廠設施規劃的差異性，每間製造廠各有一條生產線及 8 台不同種類之加工機器。本次排程規劃週期中有 10 張訂單，為了能更接近現實環境的情況，針對每一種類產品的各加工途程，制定上機批量數的下限，如果切割出的批量數沒有大於此限制，代表加工此批量不符合經濟效益，便不予以開機加工；而訂單之交期為一線性模糊的三角模糊數，與傳統 Job shop 不同的是每張訂單的作業數並不相同。訂單資料如表 4.1 所示。

### 4.2 定性資料評估

在定性因素考量方面，本章實證中將考量訂單本身的利潤、該訂單顧客以往的歷史交易、該訂單產品在市場上的因素以及該訂單顧客未來的潛在訂單四項因素。定量因素上，以排程的製距、訂單交期滿足度以及機器使用率作為評估指標。而適應性函數中定性與定量以及各定性因素、定量因素之權重必須由規劃人員評判各因素之重要性做配對比較(如表 4.2、表 4.3、表 4.4)，關於定性因素也是由規劃人員依據各因素來對訂單作兩兩配對比較(如表 4.7、表 4.8、表 4.9、表 4.10)。最後計算出各訂單在定性因素下的加權值，求出定性因素之訂單順序(表 4.11)。所以，在本章實證中定性因素之訂單順序為：

訂單 4=> 訂單 1=> 訂單 3=> 訂單 9=> 訂單 8=> 訂單 7=> 訂單 6=>

訂單 2=> 訂單 5=> 訂單 10

表 4.1 訂單資料

零件 訂單	作業	加工 途程	加工 時間	交 期				
1	1	3	22	55	7	3	2	17
	2	4	8			4	8	24
	3	7	21			5	5	19
	4	1	15			6	6	12
	5	6	13			1	2	15
	6	8	20			2	7	20
2	1	8	23	100	8	3	1	22
	2	3	17			4	8	21
	3	1	14			5	5	23
	4	2	16			1	4	14
	5	5	17			2	6	15
	6	7	19			3	7	17
3	1	3	21	104	9	4	2	17
	2	2	18			5	3	14
	3	4	7			6	8	22
	4	5	16			1	2	9
	5	8	24			2	1	19
4	1	6	11	121	10	3	7	20
	2	5	17			4	5	7
	3	4	10			5	4	15
	4	2	19			6	3	22
	5	7	18			1	8	13
	6	1	16			2	5	11
5	1	1	15	60	10	3	3	15
	2	3	19			4	2	8
	3	7	19			5	6	20
	4	5	18					
6	1	1	17	72				
	2	4	9					

表 4.2 定性與定量之權重值

	定性因素	定量因素	$\lambda=1.995$	評估值
定性因素	1	1/3	0.3148	0.25
定量因素	3	1	0.9492	0.75

表 4.3 定性因素之權重值

定性因素	訂單利潤	歷史交易	市場因素	潛在訂單	$\lambda=4.1224$	評估值
訂單利潤	1	3	2	5	0.8028	0.47
歷史交易	1/3	1	1/3	3	0.2683	0.16
市場因素	1/2	3	1	3	0.5153	0.30
潛在訂單	1/5	1/3	1/3	1	0.1342	0.07

表 4.4 定量因素之權重值

定量因素	製距	機器使用率	交期滿足	$\lambda=3.0537$	評估值
製距	1	5	1/3	0.3916	0.28
機器使用率	1/5	1	1/7	0.1005	0.07
交期滿足	3	7	1	0.9146	0.65

由表 4.2 與 4.4 可得，傳統遺傳演算法與混合式遺傳演算法進行時的適應性函數如下：

$$\begin{aligned} \text{最適適應性函數值} &= 0.75*(0.28*\text{製距}+0.65*\text{交期滿足}+0.07*\text{機器使用率}) \\ &+ 0.25*(1-\text{懲罰函數值}) \end{aligned}$$

### 4.3 基本參數設定

在進行演算法運作之前，我們先就本機制中的相關參數進行介紹。在遺傳演算法的基本參數設定方面，母體大小為 20 條個體，個體中的基因編碼在 3.6.1 節中已提及；而在突變機率的制定上，因本研究考慮每一基因的突變，因此給予較小的突變機率，以符合突變的原則。其他相關設定如表 4.5 及表 4.6 所示。

表 4.5 遺傳演算法中參數之設定值

參數	參數值
群體大小	20
演算代數	100
交配方式	MCUOX 之改良
突變機率	0.2
精華保留個體數	1

表 4.6 混合式遺傳演算法中參數之設定值

參數	參數值
群體大小	20
演算代數	100
交配方式	MCUOX 之改良
突變機率	0.2
精華保留個體數	1
禁忌搜尋次數	4

表 4.7 定性因素-各訂單利潤之評估值

利潤	Order_1	Order_2	Order_3	Order_4	Order_5	Order_6	Order_7	Order_8	Order_9	Order_10	$\lambda=16.2806$	評估值
Order_1	1	3	1/2	1	4	5	4	5	3	5	0.4572	0.1554
Order_2	1/3	1	1/3	1/2	2	4	1	3	1/2	4	0.233	0.0792
Order_3	2	3	1	2	4	5	4	4	3	5	0.5148	0.175
Order_4	1	2	1/2	1	4	5	3	4	2	5	0.4009	0.1363
Order_5	4	1/2	1/4	1/4	1	3	1/2	2	1/3	1/2	0.2241	0.0762
Order_6	5	1/4	1/5	1/5	1/3	1	1/3	1/2	1/4	1/2	0.193	0.0656
Order_7	1/4	1	1/4	1/3	2	3	1	2	1/2	3	0.1868	0.0635
Order_8	1/5	1/3	1/4	4	1/2	2	1/2	1	1/3	1/2	0.1777	0.0605
Order_9	1/3	2	1/3	2	3	4	2	3	1	4	0.3193	0.1085
Order_10	1/5	1/4	1/5	5	2	2	1/3	2	1/4	1	0.2348	0.0798

表 4.8 定性因素-各訂單顧客歷史交易之評估值

歷史交易	Order_1	Order_2	Order_3	Order_4	Order_5	Order_6	Order_7	Order_8	Order_9	Order_10	$\lambda=10.3237$	評估值
Order_1	1	2	1/3	1/4	2	3	1/3	3	1/2	4	0.2215	0.0852
Order_2	1/2	1	1/3	1/4	1	2	1/3	2	1/2	3	0.1554	0.0598
Order_3	3	3	1	1/2	3	4	1	5	2	5	0.4423	0.17
Order_4	4	4	2	1	4	4	2	5	3	5	0.638	0.2454
Order_5	1/2	1	1/3	1/4	1	2	1/3	2	1/2	3	0.1554	0.0598
Order_6	1/3	1/2	1/4	1/4	1/2	1	1/4	2	1/3	2	0.1082	0.0416
Order_7	3	3	1	1/2	3	4	1	5	2	5	0.4423	0.17
Order_8	1/3	1/2	1/5	1/5	1/2	1/2	1/5	1	1/4	2	0.0851	0.0328
Order_9	2	2	1/2	1/3	2	3	1/2	4	1	4	0.2845	0.1094

Order_10	1/4	1/3	1/5	1/5	1/3	1/2	1/5	1/2	1/4	1	0.0676	0.026
----------	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	--------	-------

表 4.9 定性因素-各訂單市場考量之評估值

市場考量	Order_1	Order_2	Order_3	Order_4	Order_5	Order_6	Order_7	Order_8	Order_9	Order_10	$\lambda=10.5937$	評估值
Order_1	1	4	3	2	5	3	4	1/2	2	5	0.4753	0.1839
Order_2	1/4	1	1/2	1/3	3	1/2	2	1/5	1/3	2	0.1284	0.0497
Order_3	1/3	2	1	1/3	4	1/2	3	1/4	1/2	3	0.1841	0.0712
Order_4	1/2	3	3	1	5	2	4	1/2	2	4	0.3751	0.1452
Order_5	1/5	1/3	1/4	1/5	1	1/4	1/2	1/7	1/4	1/2	0.0599	0.0232
Order_6	1/3	2	2	1/2	4	1	3	1/4	1/2	4	0.2269	0.0878
Order_7	1/4	1/2	1/3	1/4	2	1/3	1	1/5	1/4	2	0.096	0.0372
Order_8	2	5	4	2	7	4	5	1	3	5	0.65	0.2515
Order_9	1/2	3	2	1/2	4	2	4	1/3	1	4	0.296	0.1145
Order_10	1/5	1/2	1/3	1/4	2	1/4	2	1/5	1/4	1	0.0924	0.0358

表 4.10 定性因素-各訂單顧客潛在訂單之評估值

潛在訂單	Order_1	Order_2	Order_3	Order_4	Order_5	Order_6	Order_7	Order_8	Order_9	Order_10	$\lambda=10.5526$	評估值
Order_1	1	0.5	0.25	0.2	0.5	0.33	0.25	2	0.25	3	0.1076	0.0414
Order_2	2	1	0.33	0.2	1	0.2	0.33	2	0.5	3	0.1456	0.056
Order_3	4	3	1	0.5	3	2	1	4	0.5	5	0.3508	0.1349
Order_4	5	5	2	1	5	3	2	5	2	7	0.625	0.2403
Order_5	2	1	0.33	0.2	1	0.5	0.33	2	0.25	3	0.1403	0.054
Order_6	3	2	0.5	0.33	2	1	0.5	3	0.33	4	0.2259	0.0869
Order_7	4	3	1	1/2	3	2	1	4	1/2	5	0.3508	0.1349

Order_8	1/2	1/2	1/4	1/5	1/2	1/3	1/4	1	1/5	2	0.0853	0.0328
Order_9	4	4	2	1/2	4	3	2	5	1	5	0.4974	0.1913
Order_10	1/3	1/3	1/5	1/7	1/3	1/4	1/5	2	1/5	1	0.0716	0.0275

表 4.11 各訂單定性因素之評估值

	利潤	歷史交易	市場考量	潛在訂單	加權值	排序	訂單順序
權重	0.47	0.16	0.30	0.07			
Order_1	0.1554	0.0852	0.1839	0.0414	0.1447	2	4
Order_2	0.0792	0.0598	0.0497	0.056	0.0656	8	1
Order_3	0.175	0.17	0.0712	0.1349	0.1403	3	3
Order_4	0.1363	0.2454	0.1452	0.2403	0.1637	1	9
Order_5	0.0762	0.0598	0.0232	0.054	0.0562	9	8
Order_6	0.0656	0.0416	0.0878	0.0869	0.0699	7	7
Order_7	0.0635	0.17	0.0372	0.1349	0.0776	6	6
Order_8	0.0605	0.0328	0.2515	0.0328	0.1114	5	2
Order_9	0.1085	0.1094	0.1145	0.1913	0.1162	4	5
Order_10	0.0798	0.026	0.0358	0.0275	0.0544	10	10



## 4.4 傳統遺傳演算法(GA)、混合式遺傳演算法(HGA)比較

將上述例子運算結果，彙製成下列整體之最適適應函數值趨勢圖，如圖 4.1，可知混合式遺傳演算法(HGA)應用於多廠區排程時，所得到的結果差異明顯要比傳統遺傳演算法(GA)較佳。

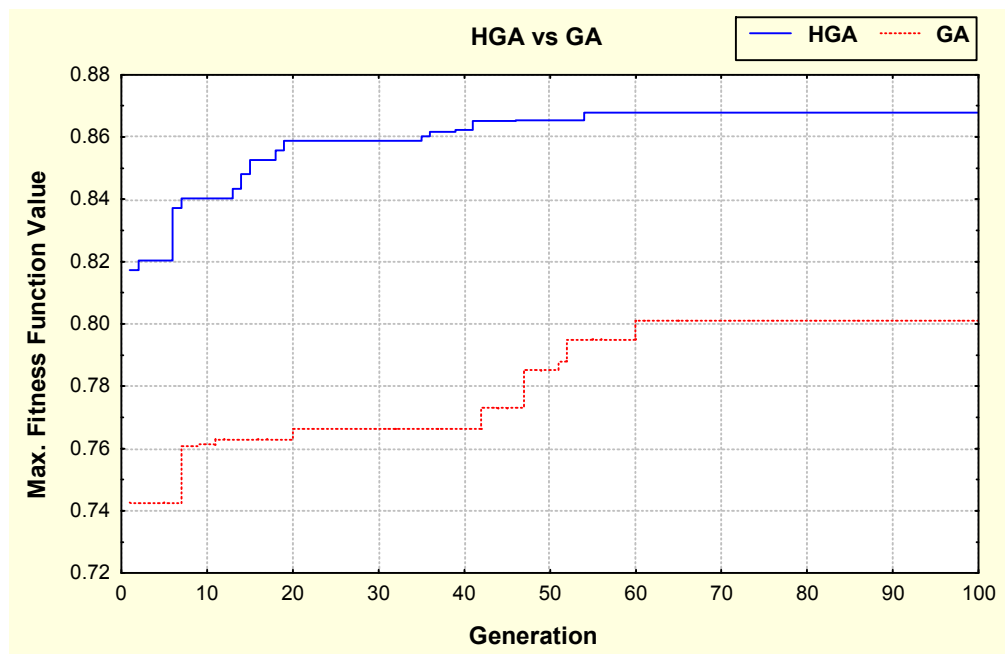


圖 4.1 廠一，二，三整體最適適應函數值趨勢圖

表 4.12 機台使用率(GA)

	M1	M2	M3	M4	M5	M6	M7	M8
廠一	0.6753	0.7716	0.4991	0.4776	0.7609	0.6574	0.6549	0.6470
廠二	0.6489	0.5806	0.5294	0.3461	0.5635	0.3910	0.5672	0.6898
廠三	0.5924	0.5161	0.4293	0.3051	0.4171	0	0.4425	0.5882

表 4.13 機台使用率(HGA)

	M1	M2	M3	M4	M5	M6	M7	M8
廠一	0.6148	0.8117	0.9050	0.4811	0.6272	0.5948	0.5617	0.8467
廠二	0.7963	0.5806	0.5865	0.8267	0.5665	0.4607	0.5332	0.5400
廠三	0.3889	0.5161	0.7747	0.3051	0.4496	0.6632	0.4307	0.5882

表 4.14 零件訂單交期滿足度(GA)

零件訂單編號	1	2	3	4	5	6	7	8	9	10	平均
交期	55	100	104	121	60	72	80	110	77	50	
完工時間	86.1	85.5	81.5	77	48	70.3	81.3	93.5	93.1	52.5	
交期滿足度	0	1	1	1	1	1	0.87	1	0	0.75	<b>0.762</b>

表 4.15 零件訂單交期滿足度(HGA)

零件訂單編號	1	2	3	4	5	6	7	8	9	10	平均
交期	55	100	104	121	60	72	80	110	77	50	
完工時間	106.5	85.5	80.8	78.8	61.8	64.8	80.3	90.6	61.6	54.5	
交期滿足度	0	1	1	1	0.82	1	0.97	1	1	0.55	<b>0.834</b>

表 4.16 最適排程結果(GA)

	作業		作業		作業		作業		作業		作業		
	開始	結束	開始	結束	開始	結束	開始	結束	開始	結束	開始	結束	
廠一	M1	6-1	2-3	1-4	4-6								
		0	17	29	43	48.1	53.1	61	77				
	M2	7-1	9-1	3-2	6-3	10-4	2-4	8-4					
		0	7.5	7.5	12	21	30	30	38.5	38.5	42.5	43	59
	M3	3-1	10-3	8-5	9-6								
		0	21	29	36.5	75.1	82.1	82.1	93.1				
	M4	4-3	3-3	9-5									
		14	24	30	37	52	67						
廠二	M5	4-2	10-2	3-4	9-4	6-5	2-5						
		5.5	14	14.1	25.1	37	45	45	52	52	58.3	59	76
	M6	4-1	8-2	10-5	1-5								
		0	5.5	14	29	42.5	52.5	53.1	66.1				
	M7	7-2	5-3	9-3	4-5	8-3	2-6						
		7.5	14.1	20.5	30	31	37.6	43	61	61	66.6	76	85.5
	M8	2-1	10-1	6-4	7-4	1-6	8-6						
		0	7.6	7.6	14.1	38.5	50.5	50.5	57.5	66.1	86.1	86.1	93.5
廠三	M1	5-1	7-3	1-4									
		0	7.5	14.1	36.1	48.1	53.1						
	M2	7-1	3-2	6-3	10-4	8-4							
		0	7.5	21	30	30	38.5	38.5	42.5	66.6	75.1		
	M3	1-1	5-2	2-2	10-3	8-5							
		0	11	11	20.5	20.5	29	29	36.5	75.1	82.1		
	M4	6-2											
		17	26										
廠四	M5	4-2	3-4	6-5	7-5								
		5.5	14	37	45	52	58.3	58.3	81.3				
	M6	4-1	10-5	6-6									
		0	2.25	42.5	52.5	58.3	70.3						
	M7	7-2	5-3	9-3	1-3	8-3	2-6						
		7.5	14.1	20.5	30	31	37.6	37.6	48.1	61	66.6	76	85.5
	M8	2-1	10-1	6-4	7-4	3-5	8-6						
		0	7.6	7.6	14.1	38.5	50.5	50.5	57.5	57.5	81.5	86.1	93.5
廠五	M1	5-1	9-2	1-4									
		0	7.5	12	31	48.1	53.1						
	M2	9-1	4-4										
		7.5	12	24	43								
	M3	1-1	5-2	2-2	9-6								
		0	11	11	20.5	20.5	29	82.1	93.1				
	M4	8-1	1-2										
		0	14	14	22								
廠六	M5	5-4	6-5										
		30	48	52	58.3								
	M6	4-1											
		0	2.25										
	M7	7-2	9-3	1-3	8-3								
		7.5	14.1	31	37.6	37.6	48.1	61	66.6				
	M8	2-1	7-4	8-6									
		0	7.6	50.5	57.5	86.1	93.5						

表 4.17 最適排程結果(HGA)

	作業		作業		作業		作業		作業		作業		
	開始	結束	開始	結束	開始	結束	開始	結束	開始	結束	開始	結束	
廠一	M1	9-2	2-3	7-3	4-6								
		3	22	22	26.6	48.6	59.6	73.5	78.8				
	M2	9-1	7-1	6-3	10-4	8-4	2-4						
		0	3	3	8	26	34.5	36.5	44.5	45.6	54.1	54.1	62.1
	M3	3-1	10-3	9-6	8-5								
		0	21	29	36.5	50.6	61.6	61.6	68.6				
	M4	8-1	6-2	4-3	3-3								
		0	7	17	26	28	38	39	46				
M5	4-2	9-4	6-5	3-4	7-5								
	11	28	28.6	35.6	46.5	52.8	52.8	68.8	68.8	80.3			
M6	4-1	8-2	6-6	1-5									
	0	11	14	19	52.8	64.8	73.5	86.5					
M7	9-3	8-3	1-3	4-5	2-6								
	22	28.6	28.6	45.6	48	58.5	58.5	67.5	79.1	85.5			
M8	2-1	10-1	6-4	7-4	3-5	1-6							
	0	7.6	7.6	12	34.5	46.5	59.6	66.6	68.8	80.8	86.5	106.5	
廠二	M1	6-1	2-3	7-3	4-6								
		0	17	22	26.6	48.6	59.6	73.5	78.8				
	M2	9-1	7-1	6-3	8-4	2-4							
		0	3	3	8	26	34.5	45.6	54.1	54.1	62.1		
	M3	1-1	2-2	5-2	10-3	9-6							
		0	11	11	19.5	19.5	29	29	36.5	50.6	61.6		
	M4	1-2	9-5										
		11	19	35.6	50.6								
M5	10-2	6-5	5-4	2-5									
	12	23	46.5	52.8	52.8	61.8	62.1	79.1					
M6	8-2	10-5											
	14	19	44.5	54.5									
M7	9-3	7-2	4-5	2-6									
	22	28.6	28.6	48.6	58.5	67.5	79.1	85.5					
M8	2-1	10-1	6-4	7-4	3-5								
	0	7.6	7.6	12	34.5	46.5	59.6	66.6	68.8	80.8			
廠三	M1	5-1	2-3	1-4	4-6								
		0	15	22	26.6	58.5	73.5	73.5	78.8				
	M2	9-1	7-1	3-2	4-4								
		0	3	3	8	21	39	39	58				
	M3	1-1	2-2	5-2	8-5								
		0	11	11	19.5	19.5	29	61.6	68.6				
	M4	8-1											
		0	7										
M5	6-5	5-4	7-5										
	46.5	52.8	52.8	61.8	68.8	80.3							
M6	8-2	10-5											
	14	19	44.5	54.5									
M7	9-3	5-3	1-3	2-6									
	22	28.6	29	48	48	58.5	79.1	85.5					
M8	2-1	10-1	7-4	8-6									
	0	7.6	7.6	12	59.6	66.6	68.6	90.6					

表 4. 18 各零件訂單指派順序(GA)

零件訂單 編號	作業	分割批量數	指派廠區 編號	開始時間	結束時間
1	1	2	2, 3	0	11
	2	1	3	14	22
	3	2	2, 3	37.6	48.1
	4	3	1, 2, 3	48.1	53.1
	5	1	1	53.1	66.1
	6	1	1	66.1	86.1
2	1	3	1, 2, 3	0	7.6
	2	2	2, 3	20.5	29
	3	1	1	29	43
	4	1	1	43	59
	5	1	1	59	76
	6	2	1, 2	76	85.5
3	1	1	1	0	21
	2	2	1, 2	21	30
	3	1	1	30	37
	4	2	1, 2	37	45
	5	1	2	57.5	81.5
4	1	2	1, 2	0	5.5
	2	2	1, 2	5.5	14
	3	1	1	14	24
	4	1	3	24	43
	5	1	1	43	61
	6	1	1	61	77
5	1	2	2, 3	0	7.5
	2	2	2, 3	11	20.5
	3	2	1, 2	20.5	30
	4	1	3	30	48
6	1	1	1	0	17
	2	1	2	17	26
	3	2	1, 2	30	38.5
	4	2	1, 2	38.5	50.5
	5	3	1, 2, 3	52	58.3
	6	1	2	58.3	70.3
7	1	2	1, 2	0	7.5
	2	3	1, 2, 3	7.5	14.1
	3	1	2	14.1	36.1
	4	3	1, 2, 3	50.5	57.5
	5	1	2	58.3	81.3
8	1	1	3	0	14
	2	1	1	14	29
	3	3	1, 2, 3	61	66.6
	4	2	1, 2	66.6	75.1
	5	2	1, 2	75.1	82.1
	6	3	1, 2, 3	86.1	93.5
9	1	2	1, 3	7.5	12
	2	1	3	12	31
	3	3	1, 2, 3	31	37.6
	4	1	1	45	52
	5	1	1	52	67
	6	2	1, 3	82.1	93.1
10	1	2	1, 2	7.6	14.1
	2	1	1	14.1	25.1
	3	2	1, 2	29	36.5
	4	2	1, 2	38.5	42.5
	5	2	1, 2	42.5	52.5

表 4.19 各零件訂單指派順序(HGA)

零件訂單 編號	作業	分割批量數	指派廠區 編號	開始時間	結束時間
1	1	2	2, 3	0	11
	2	1	2	11	19
	3	2	1, 3	48	58.5
	4	1	3	58.5	73.5
	5	1	1	73.5	86.5
	6	1	1	86.5	106.5
2	1	3	1, 2, 3	0	7.6
	2	2	2, 3	11	19.5
	3	3	1, 2, 3	22	26.6
	4	2	1, 2	54.1	62.1
	5	1	2	62.1	79.1
	6	3	1, 2, 3	79.1	85.5
3	1	1	1	0	21
	2	1	3	21	39
	3	1	1	39	46
	4	1	1	52.8	68.8
	5	2	1, 2	68.8	80.8
4	1	1	1	0	11
	2	1	1	11	28
	3	1	1	28	38
	4	1	3	39	58
	5	2	1, 2	58.5	73.5
	6	3	1, 2, 3	73.5	78.8
5	1	1	3	0	15
	2	2	2, 3	19.5	29
	3	1	3	29	48
	4	2	2, 3	52.8	61.8
6	1	1	2	0	17
	2	1	1	17	26
	3	2	1, 2	26	34.5
	4	2	1, 2	34.5	46.5
	5	3	1, 2, 3	46.5	52.8
	6	1	1	52.8	64.8
7	1	3	1, 2, 3	3	8
	2	1	2	28.6	48.6
	3	2	1, 2	48.6	59.6
	4	3	1, 2, 3	59.6	66.6
	5	2	1, 3	68.8	80.3
8	1	2	1, 3	0	7
	2	3	1, 2, 3	14	19
	3	1	1	28.6	45.6
	4	2	1, 2	45.6	54.1
	5	2	1, 3	61.6	68.6
	6	1	3	68.6	90.6
9	1	3	1, 2, 3	0	3
	2	1	1	3	22
	3	3	1, 2, 3	22	28.6
	4	1	1	28.6	35.6
	5	1	2	35.6	50.6
	6	2	1, 2	50.6	61.6
10	1	1	1, 2, 3	7.6	12
	2	1	2	12	23
	3	2	1, 2	29	36.5
	4	1	1	36.5	44.5
	5	2	2, 3	44.5	54.5

## 4.5 多廠區最適排程解

依據多廠區生產指派與排程系統運算，可得知各廠區機台上分佈之零件訂單分割批量以及最終之完工時間，並將最適排程結果，如表 4.12 及 4.13 作為多廠間進行訂單指派及生產排程的參考依據。各廠區之最適排程結果如下：

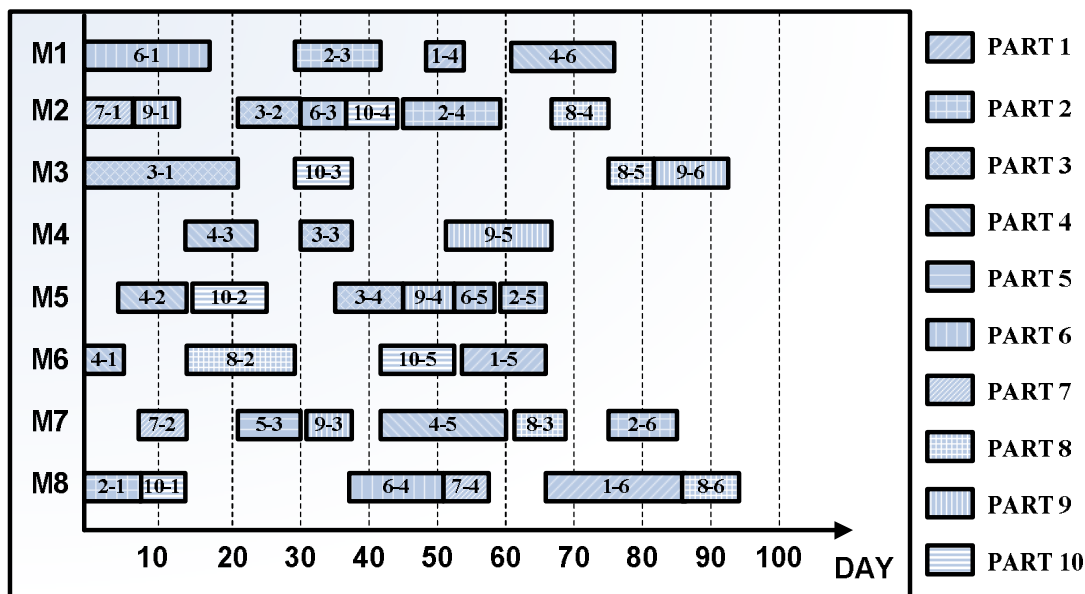


圖 4.2 廠一最適排程(GA)

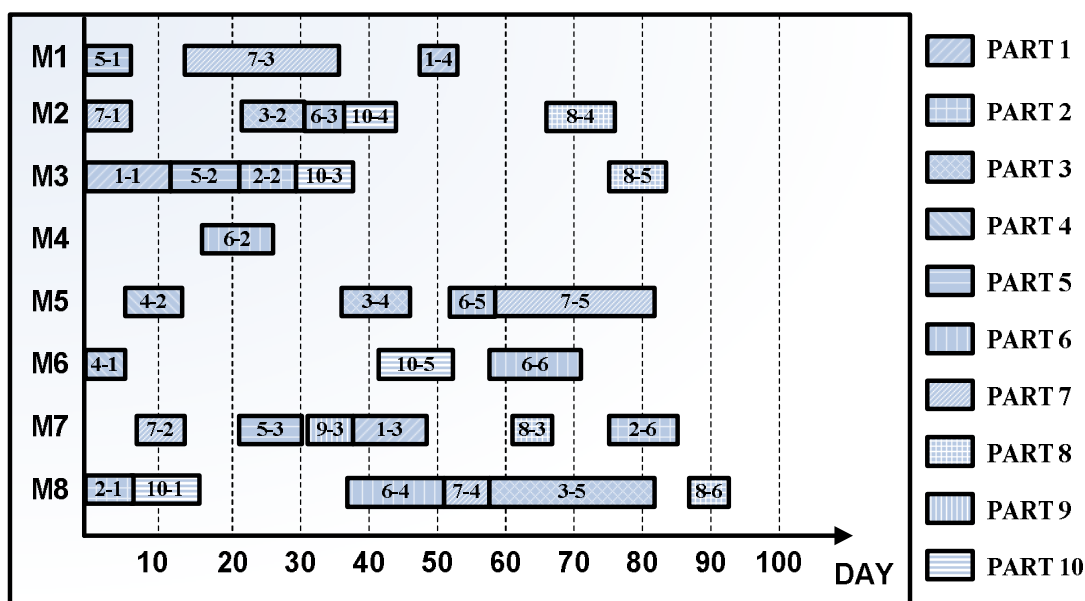


圖 4.3 廠二最適排程(GA)

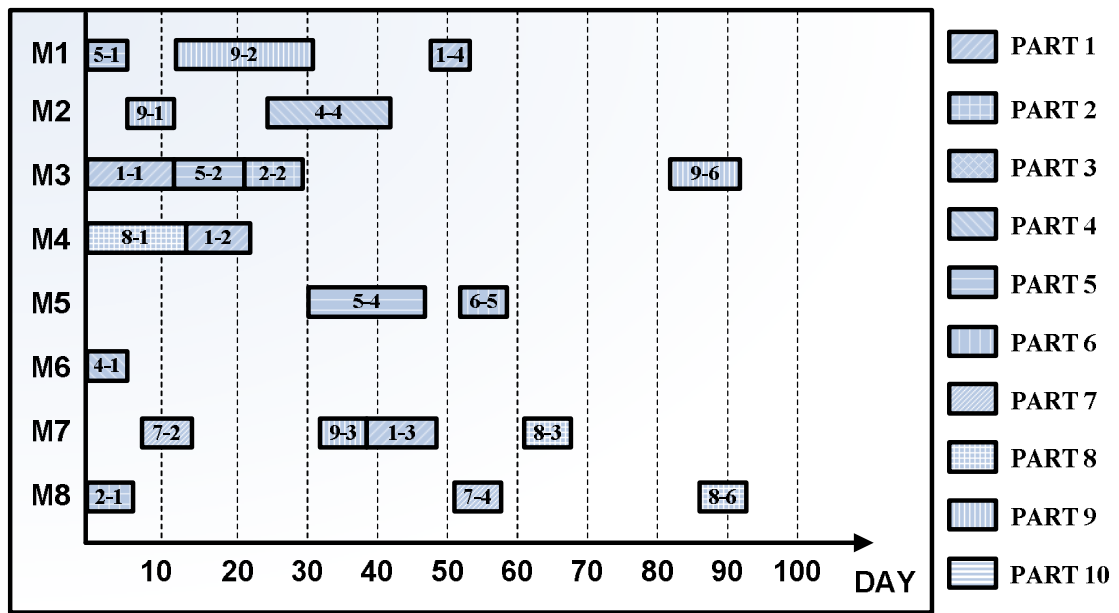


圖 4.4 廠三最適排程(GA)

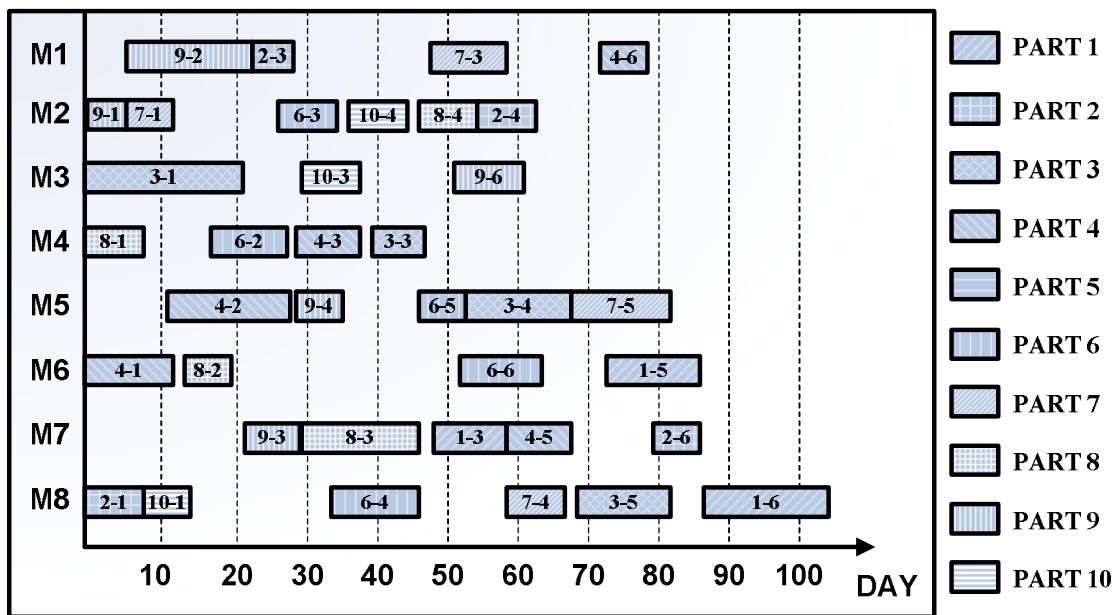


圖 4.5 廠一最適排程(HGA)

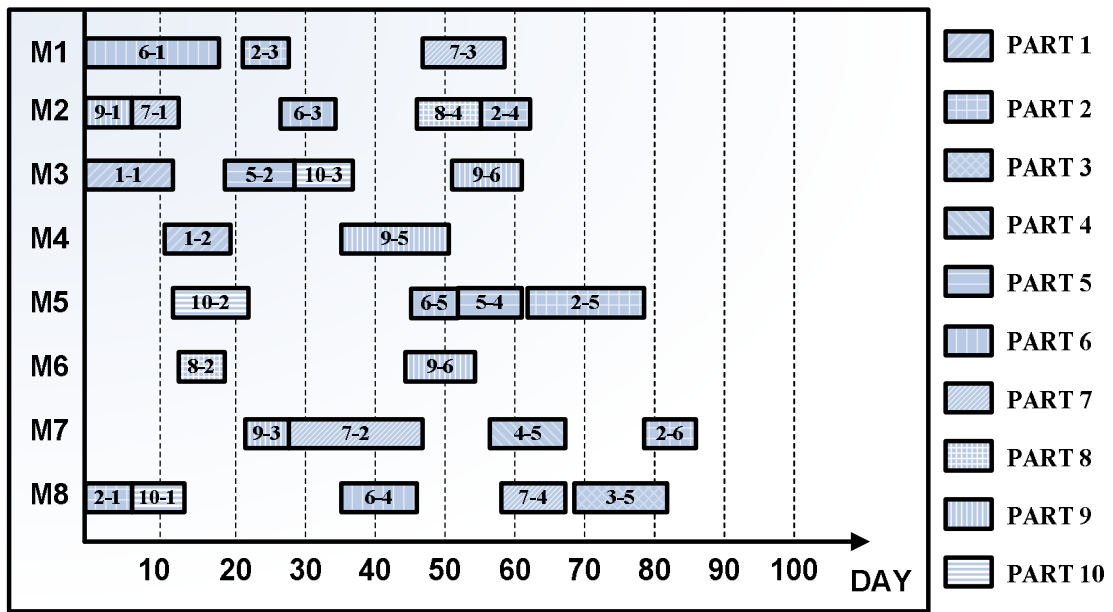


圖 4.6 廠二最適排程(HGA)

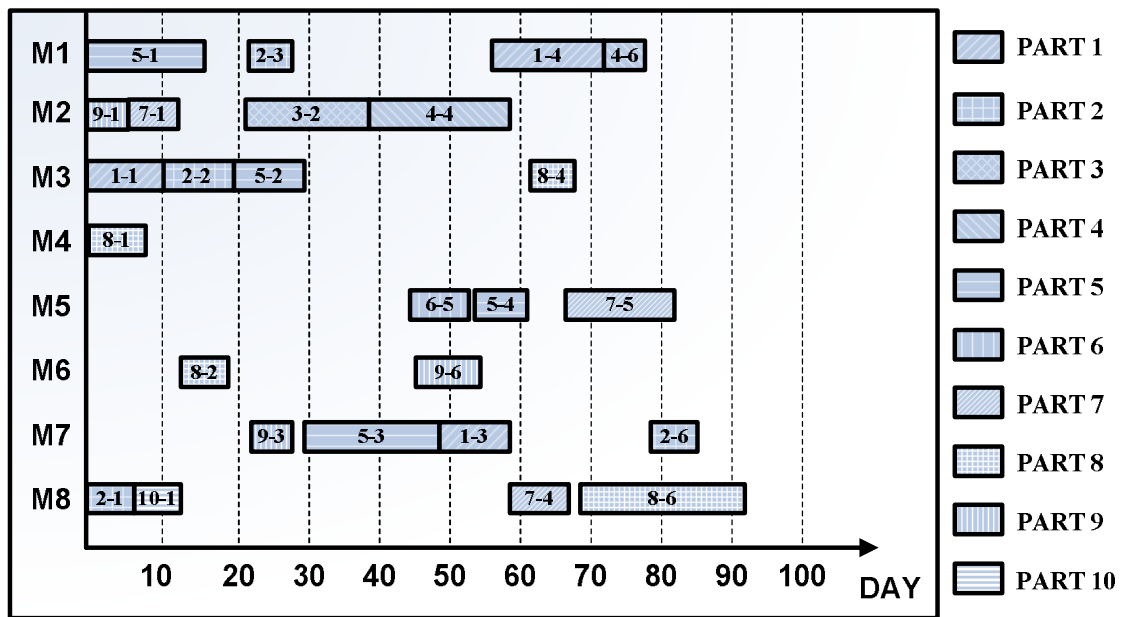


圖 4.7 廠三最適排程(HGA)



## 4.6 實驗設計

多廠區生產指派與排程問題求解，分別利用不同實驗設計，以傳統遺傳演算法(GA)與混合式遺傳演算法(HGA)比較其效能上差異。演算代數皆為 100 代，母體為 20，禁忌搜尋次數為 4 次。實驗運算結果如下：

表 4. 20 10 張零件訂單(以相同運算代數比較)

總零件訂單數	總作業數	總廠區數	總機台數	代數	實驗回數	最佳解		搜尋時間(sec)	
						GA	HGA	GA	HGA
10	55	3	24	100	1	0.8114	0.8915	5544.8	13694.6
					2	0.7925	0.8815	5487.2	14191.6
					3	0.8224	0.8857	5528.7	14875.3
					4	0.8025	0.8745	5641.3	13589.5
					5	0.7856	0.8624	5513.5	14367.8
					平均	<b>0.8029</b>	<b>0.8791</b>	<b>5543.1</b>	<b>14143.8</b>
10	55	4	32	100	1	0.8752	0.9417	6854.1	17135.2
					2	0.8657	0.9381	6864.3	17121.6
					3	0.8643	0.9452	6821.5	17221.3
					4	0.8751	0.9428	6881.4	17243.7
					5	0.8681	0.9327	6892.5	17112.5
					平均	<b>0.8697</b>	<b>0.9401</b>	<b>6862.8</b>	<b>17166.9</b>

表 4. 21 20 張零件訂單(以相同運算代數比較)

總零件訂單數	總作業數	總廠區數	總機台數	代數	實驗回數	最佳解		搜尋時間(sec)	
						GA	HGA	GA	HGA
20	110	3	24	100	1	0.5690	0.6910	13086.1	55158.4
					2	0.5782	0.7221	13136.4	55231.5
					3	0.5721	0.7114	13034.2	55325.1
					4	0.5832	0.6865	13241.5	55178.3
					5	0.5731	0.7243	13301.1	55258.5
					平均	<b>0.5751</b>	<b>0.7071</b>	<b>13159.9</b>	<b>55230.4</b>
20	110	4	32	100	1	0.7086	0.8073	15472.7	64315.2
					2	0.7214	0.8128	15446.6	64378.8
					3	0.7152	0.7965	15443.1	63985.6
					4	0.7043	0.8087	15415.8	64298.2
					5	0.7078	0.8045	15463.6	64487.5
					平均	<b>0.7115</b>	<b>0.8060</b>	<b>15448.4</b>	<b>64293.1</b>
20	110	5	40	100	1	0.7862	0.8748	20245.1	80572.5
					2	0.7956	0.8856	20241.6	80486.5
					3	0.7885	0.8812	20356.8	80754.5
					4	0.7984	0.8756	20568.7	80658.4
					5	0.7996	0.8789	20485.5	80568.9

					平均	<b>0.7937</b>	<b>0.8792</b>	<b>20379.5</b>	<b>80608.2</b>
					1	0.8546	0.9226	23244.1	92154.5
					2	0.8532	0.9257	23181.5	92584.2
20	110	6	48	100	3	0.8478	0.9121	23224.2	92231.1
					4	0.8556	0.9113	23258.8	91987.5
					5	0.8547	0.9251	23458.6	92348.3
					平均	<b>0.8532</b>	<b>0.9194</b>	<b>23273.4</b>	<b>92261.1</b>

表 4. 22 30 張零件訂單(以相同運算代數比較)

總零件訂單數	總作業數	總廠區數	總機台數	代數	實驗回數	最佳解		搜尋時間(sec)	
						GA	HGA	GA	HGA
30	165	3	24	100	1	0.5271	0.5883	19990.2	133908.3
					2	0.5381	0.6021	19925.5	133951.2
					3	0.5287	0.5898	19987.1	133895.5
					4	0.5296	0.5951	19947.5	133945.2
					5	0.5397	0.6151	19996.7	133957.6
					平均	<b>0.5327</b>	<b>0.5981</b>	<b>19969.4</b>	<b>133931.6</b>
30	165	4	32	100	1	0.5571	0.6754	24808.9	165600.5
					2	0.5728	0.7024	24878.2	165524.2
					3	0.5542	0.6958	24752.4	165682.2
					4	0.5683	0.6765	24764.1	166152.5
					5	0.5423	0.6825	24892.1	166252.8
					平均	<b>0.5589</b>	<b>0.6865</b>	<b>24819.1</b>	<b>165842.4</b>

由以上實驗結果可知，在相同運算代數(100 代)情況下，以傳統遺傳演算法(GA)與混合式遺傳演算法(HGA)比較其效能上差異，混合式 GA 明顯較優於傳統 GA，但實際上對於傳統 GA 並不公平，因在求解代數上，若傳統 GA 運算時間與混合式 GA 相同，傳統 GA 在效能上未必會比混合式 GA 來的差，因此，下面將以相同運算搜尋時間為基準來驗證此項假設，並比較效能上之差異。

表 4. 23 10 張零件訂單(以相同運算搜尋時間比較)

總零件訂單數	總作業數	總廠區數	總機台數	運算搜尋時間(sec)	實驗回數	最佳解/代數	
						GA	HGA
10	55	3	24	14143.8	1	0.8396/247	0.8915/100
					2	0.8401/250	0.8815/100
					3	0.8321/243	0.8857/100
					4	0.8386/248	0.8745/100
					5	0.8412/251	0.8624/100
					平均	<b>0.8383</b>	<b>0.8791</b>
10	55	4	32	17166.9	1	0.8971/265	0.9417/100
					2	0.8832/269	0.9381/100
					3	0.8962/261	0.9452/100
					4	0.8721/259	0.9428/100
					5	0.8845/261	0.9327/100
					平均	<b>0.8867</b>	<b>0.9401</b>

表 4. 24 20 張零件訂單(以相同運算搜尋時間比較)

總零件訂單數	總作業數	總廠區數	總機台數	運算搜尋時間(sec)	實驗回數	最佳解/代數	
						GA	HGA
20	110	3	24	55230.4	1	0.6221/415	0.6910/100
					2	0.6339/412	0.7221/100
					3	0.6298/423	0.7114/100
					4	0.6225/418	0.6865/100
					5	0.6254/415	0.7243/100
					平均	<b>0.6267</b>	<b>0.7071</b>
20	110	4	32	64293.1	1	0.7532/420	0.8073/100
					2	0.7545/418	0.8128/100
					3	0.7485/423	0.7965/100
					4	0.7448/421	0.8087/100
					5	0.7559/425	0.8045/100
					平均	<b>0.7514</b>	<b>0.8060</b>
20	110	5	40	80608.2	1	0.8225/397	0.8748/100
					2	0.8335/392	0.8856/100
					3	0.8345/401	0.8812/100
					4	0.8287/396	0.8756/100
					5	0.8285/395	0.8789/100
					平均	<b>0.8295</b>	<b>0.8792</b>
20	110	6	48	92261.1	1	0.8428/396	0.9226/100
					2	0.8565/398	0.9257/100
					3	0.8587/402	0.9121/100
					4	0.8545/396	0.9113/100
					5	0.8456/392	0.9251/100
					平均	<b>0.8516</b>	<b>0.9194</b>

表 4.25 30 張零件訂單(以相同運算搜尋時間比較)

總零件訂單數	總作業數	總廠區數	總機台數	運算搜尋時間(sec)	實驗回數	最佳解(代數)	
						GA	HGA
30	165	3	24	133931.6	1	0.5521/669	0.5883/100
					2	0.5487/672	0.6021/100
					3	0.5497/671	0.5898/100
					4	0.5565/668	0.5951/100
					5	0.5482/678	0.6151/100
					平均	<b>0.5510</b>	<b>0.5981</b>
30	165	4	32	165842.4	1	0.6015/667	0.6754/100
					2	0.6042/669	0.7024/100
					3	0.5998/674	0.6958/100
					4	0.6124/667	0.6765/100
					5	0.6127/678	0.6825/100
					平均	<b>0.6061</b>	<b>0.6865</b>

由以上實驗，可以證實無論是傳統 GA 或混合式 GA，在相同代數及相同運算搜尋時間條件下，混合式 GA 應用在多廠區指派與排程系統上，效能比傳統 GA 好。

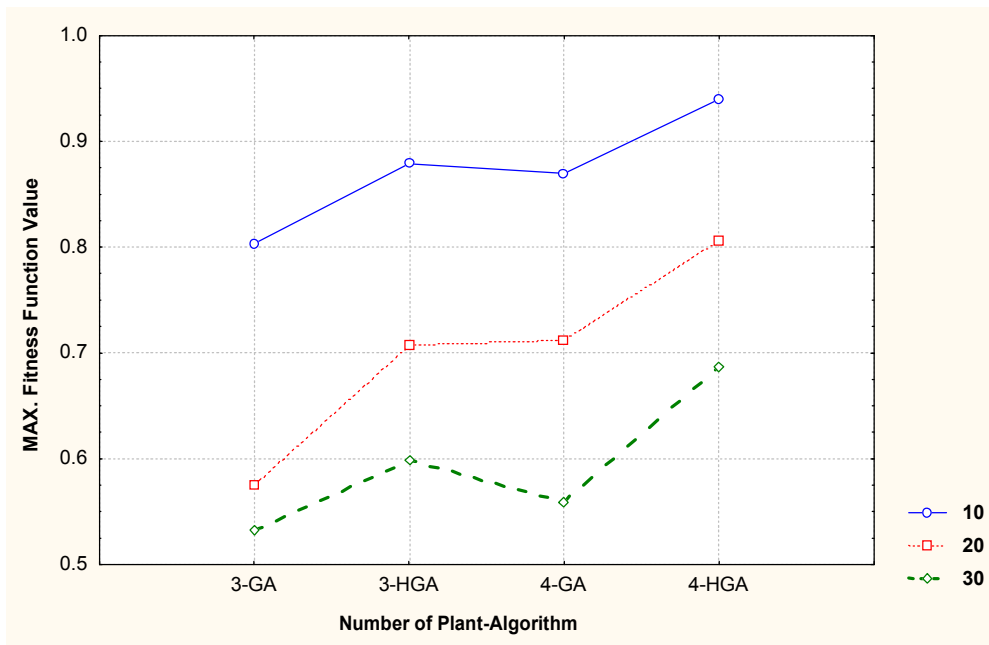


圖 4.8 不同零件訂單數目效能比較(以相同運算代數比較)

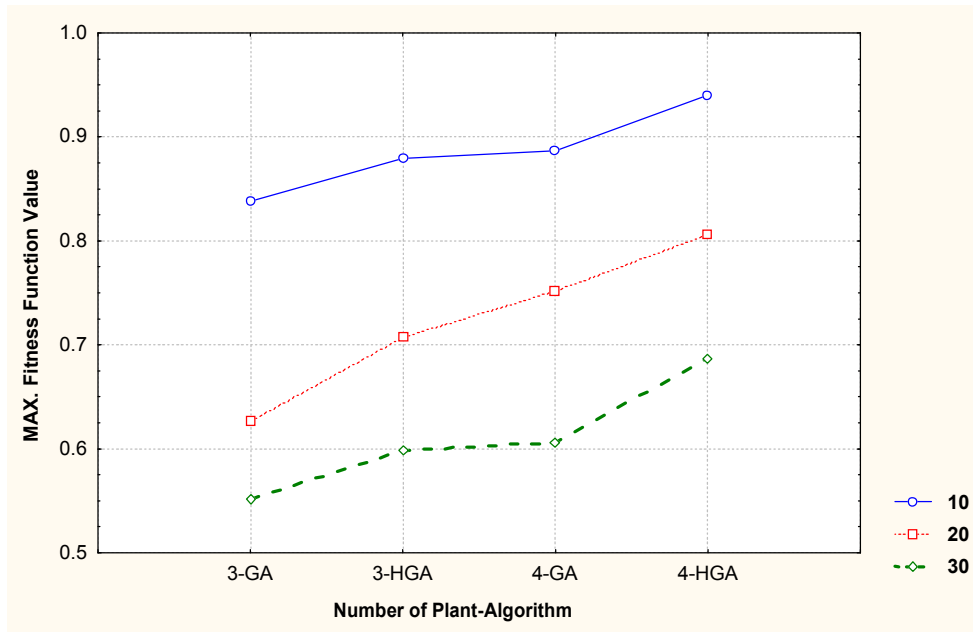


圖 4.9 不同零件訂單數目效能比較(以相同運算搜尋時間比較)

由圖 4.8 可得知在不同零件訂單數下(以相同運算代數比較)以及圖 4.9 不同零件訂單數下(以相同運算搜尋時間比較)，混合式 GA 表現比傳統 GA 好。

## 第五章 結論與未來研究方向

### 5.1 研究總結

由上述實驗分析可知，當我們將混合式遺傳演算法(HGA)應用於多廠區排程時，所得到的結果差異明顯要比傳統遺傳演算法(GA)較佳，因此我們可推測出以下幾點結論：

1. 以混合式遺傳演算法求解多廠區生產指派與排程問題時，效能表現上遠比傳統遺傳演算法來的較好。
2. 在一般的多廠區訂單指派研究當中，並無將廠內的生產排程內入考慮，本研究結合廠內生產排程與訂單指派使的在訂單指派時，更能以整體性的考量來做指派的動作，並以整體最佳化為理論基礎，與兩階段考量方式不同，整體化考量的效果明顯較好。
3. 在一般多目標排程的研究，其皆是以現場製造的績效，例如製距、使用率等做為評估因子，本研究除了考量現場績效，另外也將定性因素納入考量。
4. 排程演算法設計中考慮零件訂單作業的動態批量分割，使一作業同時在多廠機器上處理，能有效地縮短總生產時間，發揮多廠機器的功用。
5. 將最適結果可作為多廠間進行訂單指派及生產排程的參考依據。

### 5.2 未來研究方向

雖然本研究經由實證驗證，確實有達到預期的效果，但在發展本研究架構與問題實作的過程中，發現一些值得進一步探討的問題，將可使本研究更為完善，以下歸納出值得探討的未來研究方向：

1. 以方法論而言，本研究著重於發展系統架構，對於其中運算子並無做深入的探討與設計，所以後續研究可以針對演算法中各運算子，提出更有效率的運算子設計。
2. 本研究在假設條件上對於機台的整備成本暫不予以考慮。而後續研究可以就整備成本及其他更貼近現實的環境條件加入系統，使得系統在執行能力上能更貼近現實環境。

## 參考文獻

- [1] 吳信儀，「以改良之進化策略演算法解決排序問題之研究—SRS演算法與多重工作者系統之發展」，東海大學工業工程研究所碩士論文，1996。
- [2] 林我聰，現場排程專家系統-應用個體導向技術建立之研究，資訊與電腦公司出版，1994。
- [3] Adenso, D., “An SA/TA mixture algorithm for the scheduling tardiness problem”, *European Journal of Operation Research*, Vol. 88, pp516-524, 1996.
- [4] Allahverdi, A., “The two- and m-machine flowshop scheduling problems with bicriteria of makespan and mean flowtime”, *European Journal of Operational Research*, 147, pp.373-396, 2003.
- [5] Allahverdi, A., “A new heuristic for m-machine flowshop scheduling problem with bicriteria of makespan and maximum tardiness”, *Computers and Operations Research*, 31, pp.157-180, 2004.
- [6] Baker, K.R., *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York, 1974.
- [7] Baker, R.K., “Sequence rules and due-date assignments in a job shop”, *Management Science*, 30, pp.1093-1104, 1984.
- [8] Birman, M. and Mosheiov, G., “A note on a due-date assignment on a two-machine flow-shop”, *Computers and Operations Research*, 31, pp.473-480, 2004.
- [9] Brown, J.R. and Ozgur, C.O., “Priority class scheduling: product scheduling for multi-objective environment”, *Production Planning and Control*, 8, pp.762-770, 1997.
- [10] Bullinger, H.-J., Faehnrich, K.-P., Laubscher, H.-P., “Planning and multi-site production – an object-oriented model”, *Int. J. Production Economics*, 51, (1997), 19-35.
- [11] Chang, P.T., Lo, Y. T., “Modelling of job-shop scheduling with multiple quantitative and qualitative objectives and a GA/TS mixture approach”, *International Journal of Computer Integrated Manufacturing*, v 14, n 4, July/August, 2001, p 367-384.
- [12] Chang, P.-C., Hsieh, J.-C. and Lin, S.-G., “The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem”, *International Journal of Production Economics*, 79, pp.171-183, 2002.
- [13] Cheng, R. and Gen, M., “Parallel machine scheduling problems using mimetic algorithms”, *Computer and Industrial Engineering*, Vol. 33, No. 3-4, pp761-764, 1997.
- [14] Chou, F.D. and Lee, C.E., “Two-machine flowshop scheduling with bicriteria problem”, *Computers & Industrial Engineering*, 36, pp.549-564, 1999.
- [15] Croce, F.D., Tadei, R. and Volta, G., “A genetic algorithm for the job shop problem”, *Computers Ops.res.* 22:1 (1995) 15-24.
- [16] Daniels, R.L., “Incorporating performance information into multi-objective scheduling”, *European Journal of Operational Research*, 77, pp.272-286, 1994.
- [17] Esquivel, S., Ferrero, S., Gallard, R., Salto, C., Alfonso, H. and Schutz, M., “Enhanced

- evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem”, *Knowledge-Based Systems*, 15, pp.13-25, 2002.
- [18] Fang, H. L., “Genetic Algorithm in timetabling and scheduling”, Ph. D. dissertation, Department of Artificial Intelligent, University of Edinburgh, 1994.
- [19] Forgy, T. C., “Varying the probability of mutation in the genetic algorithm”, *Proceedings of the 3rd International Conference on Genetic algorithms*, pp104-109, 1989.
- [20] Framinan, J.M., Leisten, R. and Ruiz-Usano, R., “Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimization”, *European Journal of Operational Research*, 141, pp.559-569, 2002.
- [21] Ghrayeb, O.A., “A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems”, *Applied Soft Computing Journal*, 2, pp.197-210, 2003.
- [22] Glover, F., “Tabu search-part I”, *ORSA Journal on Computing*, Vol. 1, No. 3, pp190-206, 1989.
- [23] Glover, F., “Tabu search-part II”, *ORSA Journal on Computing*, Vol. 2, No. 1, pp4-32, 1990.
- [24] Glover, F., “Tabu search : a tutorial ”, *Interface*, Vol. 20, No. 4, pp74-94, 1990.
- [25] Gnoni, M.G., Iavagnilio, R., Mossa, G., Mummolo, G., Di Leva, A., “Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry”, *Int. J. Production Economics* 85 (2003) 251-262.
- [26] Goldberg, D.E., “*Genetic Algorithm in Search, Optimization and Machine Learning*”, Addison-Wesley Publishing Co., 1989.
- [27] Guinet, A., “Multi-site planning: A transshipment problem”, *Int. J. Production Economics* 74 (2001) 21-32.
- [28] Ishibuchi, H. and Murata, T., “A multi-objective genetic local search algorithm and its application to flowshop scheduling”, *IEEE Transactions on System, Man, and Cybernetics-Part C: Application and Review*, 28, pp.392-403, 1998.
- [29] Ishibuchi, H., Yoshida, T. and Murata, T., “Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling”, *IEEE Transactions on Evolutionary Computation*, 7, pp.204-223, 2003.
- [30] Itoh, K, Huang, D. and Enkawa, T., “Twofold look-ahead search for multi-criterion job shop scheduling”, *International Journal of Production Research*, 31, pp.2215-2234, 1993.
- [31] Jolayemi, J.K., Olorunniwo, F.O., “A deterministic model for planning production quantities in a multi-plant, multi-warehouse environment with extensible capacities”, *Int. J. Production Economics*, 87, (2004), 99–113.
- [32] Kacem, I., Hammadi, S. and Borne, P., “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems”, *IEEE Transactions on Systems, Man and Cybernetics*, 32, pp.1-13, 2002.
- [33] Kim, G. H. and C. S. G Lee, “An evolutionary approach to the job-shop scheduling



- problem”, Proceedings IEEE International Conference on Robotics and Automation, Vol. 1, pp501-506, 1994.
- [34] Kim, C.-O., Min, H.-S. and Yih, Y., “Integration of inductive learning and neural networks for multi-objective FMS scheduling”, *International Journal of Production Research*, 36, pp.2497-2509, 1998.
- [35] Kim, D.W., Na, D.G. and Chen, F., “Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective”, *Robotics and Computer-Integrated Manufacturing*, 19, pp.173-181, 2003.
- [36] Kumar, N.S.H. and Srinivasan, G., “A genetic algorithm for job shop scheduling- A case study”, *Computers in Industry* 31 (1996) 155-160.
- [37] Lee, I., “Artificial intelligence search methods for multi-machine two-stage scheduling with due date penalty, inventory, and machining costs”, *Computers and Operations Research*, 28, pp.835-852, 2001.
- [38] Levis, A.A., “Papageorgiou, L.G., A hierarchical solution approach for multi-site capacity planning under uncertainty in the pharmaceutical industry”, *Computers and Chemical Engineering* 28 (2004) 707–725.
- [39] Leung, S.C.H., Wu, Y., Lai, K.K., “Multi-site aggregate production planning with multiple objectives: a goal programming approach”, *Production Planning & Control* 14: 5(2003) 425–436.
- [40] Liaw, C. F., “A hybrid genetic algorithm for the open shop scheduling problem”, Proceedings of the CIIE National Conference, pp36-41, 1998.
- [41] Liaw, C.-F., “A hybrid genetic algorithm for the open shop scheduling problem”, *European Journal of Operational Research*, 124, (2000), 28-42.
- [42] Lin, H.-T and Liao, C.-J., “A case study in a two-stage hybrid flow shop with setup time and dedicated machines”, *International Journal of Production Economics*, 86, pp.133-143, 2003.
- [43] Mellor, P., “A review of job shop scheduling”, *Operational Research Quarterly*, Vol. 17, No. 2, pp161-170, 1966.
- [44] Min, H. S., Yih, Y. and Kim, C.-O., “A competitive neural network approach to multi-objective FMS scheduling”, *International Journal of Production Research*, 36, pp.1749-1765, 1998.
- [45] Michalewicz, Z., *Genetic Algorithm + Data Structures = Evolution Programs*, Springer-Verg Berlin Heidelberg, 1994.
- [46] Moon, C., Kim, J., Hur, S., “Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain”, *Computers & Industrial Engineering*, 43, 2002, 331-349.
- [47] Murata, T. and Ishibuchi, H., “Performance evaluation of genetic algorithms for flowshop scheduling problems,” Proceedings the First IEEE International Conference on Evolutionary Computation, 2, pp.812-817, 1994.

- [48] Murata, T., Ishibuchi, H. and Tanaka, H., “Multi-objective genetic algorithm and its applications to flowshop scheduling”, *Computers and Industrial Engineering*, 30, pp.957-968, 1996.
- [49] Nagar, A., Haddock, J. and Heragu, S., “Multiple and bicriteria scheduling: a literature survey”, *European Journal of Operational Research*, 81, pp.88-104, 1995.
- [50] Neppalli, V.R., Chen, C.L. and Gupta, J., “Genetic algorithms for two-stage bicriteria flowshop problem”, *European Journal of Operational Research*, 95, pp.356-373, 1996.
- [51] Norman, B.A., Bean, J.C., “A genetic algorithm methodology for complex scheduling problems”, *Naval Research Logistics* 46 (1999) 199-211.
- [52] Petty, C. B., M. R. Leuze and J. J. Grefenstette, “A parallel genetic algorithm”, *Proceedings of the Second International Conference on Genetic Algorithm*, pp155-161, 1987.
- [53] Park, B.J., Choi, H.R. and Kim, H.S., “A hybrid genetic algorithm for the job shop scheduling problems”, *Computers & Industrial Engineering* 45 (2003) 597–613.
- [54] Pinedo, M., *Scheduling : theory, algorithms, and systems*, pp118-141, Prentice Hall, Inc., 1995.
- [55] Pirkul, H., Jayaraman, V., “A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution”, *Computers Ops Res.* 25:10(1998) 869-878.
- [56] Rajendran, C. and Ziegler, H., “Heuristics for scheduling in flowshops and flowline-based manufacturing cells to minimize the sum of weighted flowtime and weighted tardiness of jobs”, *Computers and Industrial Engineering*, 37, pp.671-690, 1999.
- [57] Rajendran, C. and Ziegler, H., “Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times”, *European Journal of Operational Research*, 149, pp.513-522, 2003.
- [58] Rous, W., Dauzere-Pereas, S., Lasserre, J.B., “Planning and scheduling in a multi-site environment”, *Production Planning & Control* 10:1 (1999)19- 28.
- [59] Sauer, J., Suelmann, G., Appelrath, H.-J., “Multi-site scheduling with fuzzy concepts”, *International Journal of Approximate Reasoning* 19 (1998) 145-160.
- [60] Sambasivan, M., Schmidt, C.P., “A heuristic procedure for solving multi-plant, multi-item, multi-period capacitated lot-sizing problems”, *Asia-pacific Journal of Operational Research* (2002) 87-105.
- [61] Satty, T. L., *The Analytic Hierarchical Process*, McGraw-Hill, Inc., 1977.
- [62] Sivrikaya-Serifođlu, F. and Ulusoy, G., “Parallel machine scheduling with earliness and tardiness penalties”, *Computers and Operations Research*, Vol. 26, No. 8, pp 773-787, 1999.
- [63] Thierry, C., Besnard, P., Ghattas, D., Bel G., “Multi-site planning: non flexible production units and set-up time treatment, Emerging Technologies and Factory Automation”, 1995. ETFA '95, Proceedings., 1995 INRIA/IEEE Symposium on , 3 (1995) 261–269.
- [64] Timpe, C.H., Kallrath, J., “Optimal planning in large multi-site production networks”, *European Journal of Operational Research* 126 (2000) 422-435.

- [65] Vercellis, C., “Multi-plant production planning in capacitated self-configuring two-stage serial systems”, *European Journal of Operational Research* 119 (1999) 451-460.
- [66] Wesley, B. J., and J. B. Chambers, “Solving the job shop scheduling problem with tabu search”, *IIE Transactions*, Vol.27, No. 2, pp257-263, 1995.
- [67] Wellman, M. A. and D. D. Gemmill, “A genetic algorithm approach to optimization of asynchronous automatic assembly systems”, *International Journal of Flexible Manufacturing Systems*, Vol. 7, pp27-46, 1995.
- [68] Yum, Y.S., “Genetic algorithms with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems”, *Computers & Industrial Engineering* 43 (2002) 623–644.