

東 海 大 學  
工業工程與經營資訊研究所

碩士論文

多目標平行機台零工式工廠重排程之探討

A Study of Multi-Objective Parallel-Machine

Job Shop Rescheduling

研 究 生：潘國丞

指 導 教 授：張炳騰 博士

曾宗瑤 博士

中 華 民 國 九 十 五 年 六 月

# 多目標平行機台零工式工廠重排程之探討

研究生：潘國丞

指導教授：張炳騰 博士

曾宗瑤 博士

東海大學工業工程與經營資訊研究所

## 摘要

在現實動態的製造環境中，通常會存在許多無法預期的突發狀況或變數，使得原先規劃的生產排程計劃無法持續有效的實施，必須修正或更新既有的排程計畫。由於現實環境快速的生產步調，排程規劃人員面對這些突發的中斷事件，除了需要重新規劃出一個具高生產績效的排程之外，更需要在時間成本效益的考量下，快速的規劃出重排程結果，以便立即對此突發的干擾狀況做出回應，使生產現場馬上恢復正常的作業狀態。然而傳統上大多數的排程研究，主要在於發展求解一個最佳解或是近似最佳解的演算法，在動態的生產環境中，排程結果時常需要修正或重排規劃，一個最佳化的排程可能因為實際生產環境的變動而難以有效實施，相反的，一個能夠容忍或降低中斷事件負面影響的穩健排程，也許較一般最佳解之排程來得有價值。

本研究主要於零工式平行機台的架構下，建構重排程的方法與穩健排程的求解，系統以多目標適應函數同時考量質性與量性目標。在重排程中，系統以機台故障、緊急插單、物料短缺三種情境作為排程中斷事件，受中斷後的排程以部份重排法與完全重排法進行重排。研究中以部份重排法修正排程規劃，並盡可能保持初始排程的結構，以維持生產作業的穩定性；在完全重排法中，以混合式遺傳演算法重新演化求解，以求得一個較高品質的排程規劃結果。此外，本研究亦以混合式遺傳演算法求解穩健排程，期望增加排程的彈性與穩健性，提升排程對於中斷事件的容忍程度。穩健解於排程上的涵義是指，當生產活動發生若干干擾或變異時，目標函數仍能維持高的數值。在實際生產的動態環境中，干擾情況時常發生，導致排程變更頻繁，若排程經過中斷干擾並重排後，仍然可維持不錯的績效指標，則此排程對於生產績效的維持穩定可有很大的幫助。

**關鍵字：**多目標、零工式、平行機台、重排程、穩健排程

# **A Study of Multi-Objective Parallel-Machine Job Shop Rescheduling**

Student : Guo-Cheng Pan

Advisor : Dr. Ping-Teng Chang

Dr. Tsueng-Yao Tseng

Department of Industrial Engineering and Enterprise Information  
Tunghai University

## **ABSTRACT**

In the dynamic production environment, unexpected events or disruptions occur frequently, make the production schedule infeasible and then need to be updated. This research addresses a job-shop parallel-machine rescheduling problem with multiple-objectives concerned. Rescheduling factors included in this research are machine breakdown, shortage of materials and rush order. In this research, a rescheduling method that uses partial rescheduling and complete rescheduling is proposed. We use partial rescheduling to update the initial schedule and preserve it as much as possible. In the complete rescheduling, we use Hybrid Genetic Algorithm to regenerate a schedule with better quality.

In addition to the study of rescheduling method, we present a Hybrid Genetic Algorithm to find a robust schedule (solution). A robust solution means that the value of the objective function remains high when some variations occur. Because a manufacturing system is dynamic, a traditionally optimal schedule may be hard to implement due to disruptions or changes. Thus, a robust or flexible schedule may be more valuable than an optimal schedule.

**Keywords: Multiple objectives; Parallel machines; Job shop; Rescheduling;  
Robust schedule**

## 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，承蒙曾宗瑤博士、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，使的論文內容更為完備，也體會到學海無崖的道理，在此致上最深的謝意。

在這兩年的研究過程中，感謝國禎、國平、志昇、龍廷學長在課業及生活上的幫忙，同學禎祥、香君、鼎翰的相互扶持，一起走過最艱難的時光，此外亦感謝研究室學弟妹，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。

潘國丞 謹誌於

東海大學工業工程與經營資訊研究所

民國九十五年六月

# 目錄

目錄.....	IV
表目錄.....	VI
圖目錄.....	IX
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究範圍.....	3
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 排程問題之描述.....	5
2.1.1 排程問題之分類.....	5
2.1.2 多目標排程.....	7
2.2 重排程問題之描述.....	10
2.2.1 重排程因素與問題架構.....	10
2.2.2 重排程方法.....	12
2.3 平行機器排程.....	14
2.3.1 平行機器的類型.....	15
2.3.2 單階平行機器排程.....	15
2.3.3 多階平行機器排程.....	16
2.4 遺傳演算法.....	17
2.4.1 常見 Job shop 排程問題的編碼表示法.....	19
2.4.2 常見排序問題之運算子.....	23
2.4.3 排程中遺傳演算法之應用.....	26
2.5 禁忌搜尋法.....	27
2.5.1 禁忌搜尋簡介.....	27
2.5.2 禁忌搜尋在排程上之應用.....	28
第三章 多目標平行機台零工式工廠重排程架構.....	30
3.1 多目標模式與零工式平行機台生產環境.....	31
3.1.1 定性因素模式架構.....	32
3.1.2 定量因素模式架構.....	34
3.1.3 零工式工廠平行機台生產環境.....	37
3.2 排程演算法.....	39
3.2.1 遺傳演算法的運作.....	39
3.2.2 排程法則.....	47
3.2.3 混合式遺傳演算法(Hybrid Genetic Algorithm)之設計.....	52
3.3 重排程方法.....	57
3.3.1 機器故障的重排程.....	59
3.3.2 物料短缺的重排程.....	61
3.3.3 緊急插單的重排程.....	63
3.4 穩健排程(Robust Schedule).....	64

3.4.1 穩健性指標(Robustness).....	64
3.4.2 穩定性指標(Stability).....	65
3.4.3 穩健排程方法.....	67
第四章 系統實證.....	70
4.1 實證問題說明.....	70
4.2 定性因素評估.....	72
4.3 一般排程結果.....	76
4.4 重排程結果.....	89
4.4.1 機台故障重排程.....	89
4.4.2 物料短缺重排程.....	93
4.4.3 緊急插單重排程.....	102
4.5 穩健排程結果.....	111
4.5.1 機台故障情境的穩健排程.....	111
4.5.2 物料短缺情境的穩健排程.....	115
4.5.3 緊急插單情境的穩健排程.....	118
第五章 結論與未來研究方向.....	122
5.1 結論.....	122
5.2 未來研究方向.....	123
參考文獻.....	124

## 表目錄

表 2.1 排程績效目標 .....	6
表 2.2 多目標排程文獻整理 .....	8
表 2.3 常見的重排程因素 .....	11
表 2.4 常見的重排程方案 .....	11
表 2.5 排程修正方法文獻整理 .....	13
表 2.6 單階平行機器排程文獻整理 .....	17
表 2.7 遺傳演算法之優缺點歸納表 .....	27
表 3.1 AHP 評估尺度 .....	33
表 3.2 各因素之權重制訂 .....	33
表 3.3 各方案之評估值 .....	34
表 3.4 各方案之加權排序 .....	34
表 3.5 各訂單之加權值 .....	43
表 3.6 訂單平均基因順序 .....	43
表 3.7 加權後差異平方總和之計算 .....	44
表 3.8 加權後最大差異平方總和之計算 .....	44
表 3.9 各訂單作業之基因 .....	54
表 3.10 S 1 的鄰域 .....	54
表 3.11 S 2 的鄰域 .....	54
表 3.12 S 3 的鄰域 .....	55
表 3.13 S 4 的鄰域 .....	55
表 3.14 前排程作業上半部基因 .....	60
表 3.15 需重排作業上半部基因 .....	60
表 3.16 需重排作業修正後的上半部基因 .....	60
表 4.1 各工作中心之平行機台數目 .....	70
表 4.2 訂單資料 .....	70
表 4.3 定性與定量之權重值 .....	72
表 4.4 定性因素之權重值 .....	72
表 4.5 定量因素之權重值 .....	72
表 4.6 定性因素-各訂單利潤之評估值 .....	73
表 4.7 定性因素-各訂單顧客歷史交易之評估值 .....	73
表 4.8 定性因素-各訂單市場考量之評估值 .....	74
表 4.9 定性因素-各訂單顧客潛在訂單之評估值 .....	74
表 4.10 各訂單定性因素之評估值 .....	75
表 4.11 定性評估之訂單順序結果 .....	75
表 4.12 遺傳演算法(GA)中參數之設定值 .....	76
表 4.13 混合式遺傳演算法(HGA)中參數之設定值 .....	76
表 4.14 整體適應函數值與搜尋時間 .....	77
表 4.15 製距值比較 .....	77
表 4.16 訂單交期滿足(GA) .....	77
表 4.17 訂單交期滿足(HGA) .....	78
表 4.18 機器使用率(GA) .....	78
表 4.19 機器使用率(HGA) .....	78
表 4.20 訂單的平均基因順序(GA) .....	78
表 4.21 訂單的平均基因順序(HGA) .....	79

表 4.22 搜尋結果所得之訂單順序與定性因素下訂單順序差異.....	79
表 4.23 訂單排程結果(GA).....	81
表 4.24 機器之作業排程(GA).....	83
表 4.25 訂單排程結果(HGA).....	85
表 4.26 機器之作業排程(HGA).....	87
表 4.27 機台故障情境假設.....	89
表 4.28 前排程各作業的排程優先順序.....	90
表 4.29 需重排作業原先的排程順序.....	90
表 4.30 需重排作業的排程順序.....	91
表 4.31 機器之作業排程(機台故障重排程結果).....	91
表 4.32 物料短缺情境假設.....	93
表 4.33 訂單 2 作業 3 於前排程的「作業開始整備時間」.....	94
表 4.34 階段一需重排的作業.....	94
表 4.35 階段一需重排作業的排程優先順序.....	95
表 4.36 機器之作業排程(物料短缺階段一重排結果).....	95
表 4.37 階段二需重排的作業.....	97
表 4.38 階段二需重排作業的排程優先順序.....	97
表 4.39 缺料階段二重排-各方式結果比較.....	99
表 4.40 機器之作業排程(物料短缺階段二重排結果).....	99
表 4.41 緊急插單資料.....	102
表 4.42 需重排的作業.....	102
表 4.43 定性因素-各訂單利潤之評估值.....	103
表 4.44 定性因素-各訂單顧客歷史交易之評估值.....	103
表 4.45 定性因素-各訂單市場考量之評估值.....	104
表 4.46 定性因素-各訂單顧客潛在訂單之評估值.....	104
表 4.47 各訂單定性因素之評估值.....	105
表 4.48 定性評估之訂單順序結果.....	105
表 4.49 原前排程的定性評結果.....	106
表 4.50 加入緊急插單後的定性評結果.....	106
表 4.51 插單重排作業順序.....	107
表 4.52 插單重排訂單平均順序.....	107
表 4.53 重排後訂單交期滿足度.....	108
表 4.54 機器之作業排程(緊急插單重排結果).....	108
表 4.55 工作中心 1-編號 3 機台 歷史故障狀況.....	112
表 4.56 穩健排程搜尋參數(機台故障情境).....	112
表 4.57 干擾對象各項指標值(機台故障情境).....	113
表 4.58 穩健排程結果(機台故障情境).....	113
表 4.59 最適一般排程解結果.....	113
表 4.60 一般排程解 vs 穩健排程解(機台故障情境).....	114
表 4.61 物料短缺可能狀況.....	115
表 4.62 穩健排程搜尋參數(物料短缺情境).....	115
表 4.63 干擾對象各項指標值(物料短缺情境).....	116
表 4.64 穩健排程結果(物料短缺情境).....	116
表 4.65 最適一般排程解結果.....	116
表 4.66 一般排程解 vs 穩健排程解(物料短缺情境).....	117
表 4.67 緊急插單可承接範圍.....	118

表 4.68 穩健排程搜尋參數(緊急插單情境).....	118
表 4.69 干擾對象各項指標值(緊急插單情境).....	119
表 4.70 穩健排程結果(緊急插單情境).....	119
表 4.71 最適一般排程解結果.....	119
表 4.72 緊急插單例子.....	120
表 4.73 一般排程解 vs 穩健排程解(緊急插單情境).....	120

## 圖目錄

圖 1.1 研究流程圖.....	4
圖 2.1 重排程架構.....	12
圖 2.2 一般遺傳演算法運作流程.....	19
圖 2.3 分支圖.....	22
圖 3.1 多目標平行機台零工式工廠重排程架構.....	30
圖 3.2 多目標函數架構圖.....	31
圖 3.3 交期之模糊隸屬函數.....	36
圖 3.4 零工式工廠平行機台與訂單批量分割示意圖.....	37
圖 3.5 機器整備時間示意圖.....	38
圖 3.6 上半部基因編碼表示.....	40
圖 3.7 下半部基因編碼表示.....	41
圖 3.8 上半部基因初始族群示意圖.....	41
圖 3.9 下半部基因初始族群示意圖.....	42
圖 3.10 上半部基因交配示意圖.....	46
圖 3.11 上半部突變示意圖.....	47
圖 3.12 途程間的批量分割關係(情況 1).....	49
圖 3.13 途程間的批量分割關係(情況 2).....	50
圖 3.14 途程間的批量分割關係(情況 3).....	50
圖 3.15 途程間的批量分割關係(情況 4).....	50
圖 3.16 途程間的批量分割關係(情況 5).....	51
圖 3.17 途程間的批量分割關係(情況 6).....	51
圖 3.18 途程間的批量分割關係(情況 7).....	51
圖 3.19 途程間的批量分割關係(情況 8).....	52
圖 3.20 混合式遺傳演算法(HGA)運作流程圖.....	56
圖 3.21 穩健解示意圖.....	65
圖 3.22 個體解測試性干擾機制示意圖.....	68
圖 3.23 穩健排程 HGA 流程圖.....	69
圖 4.1 整體最適適應函數值趨勢圖.....	77
圖 4.2 缺料階段二重排-GA 求解過程.....	98
圖 4.3 緊急插單 GA 演算過程.....	106
圖 4.4 HGA 穩健排程適應函數值趨勢圖(機台故障情境).....	113
圖 4.5 HGA 穩健排程適應函數值趨勢圖(物料短缺情境).....	116
圖 4.6 HGA 穩健排程適應函數值趨勢圖(緊急插單情境).....	119

# 第一章 緒論

## 1.1 研究背景與動機

在現實動態的製造環境中，通常會存在許多無法預期的突發狀況或變數，使得原先規劃的生產排程計劃無法持續有效的實施，這些突發狀況諸如緊急插單、機台當機、物料短缺、工人缺席與品質問題等等，對於生產排程皆有重大的負面影響，排程規劃人員為了因應這些突發的排程中斷事件，常常需要修正或更新既有的排程計畫，這些修正或更新既有排程的動作稱之為重排程。由於現實快速的生產步調，排程規劃人員面對這些突發的中斷事件，除了需要重新規劃出一個具高生產績效的排程之外，更需要在時間成本效益的考量下，快速的規劃出重排程結果，以便立即對此突發的干擾狀況做出回應，使生產現場馬上恢復正常的作業狀態。

傳統上大多數的排程研究，是在於尋求一個最佳解(optimal)或是近似最佳(near optimal)解的演算法。然而，在動態的生產環境中，排程結果時常需要修正或重排規劃，一個最佳化的排程可能因為實際生產環境的變動而難以實施，因此傳統上的最佳解排程，於動態生產環境的實質意義也許不大；相反的，一個能夠容忍或降低中斷事件負面影響的排程，也許較一般最佳解之排程來得有價值。因此，在動態的生產環境中，排程求解除了須考量目標函數值的表現外，也應該注重該排程解是否具備彈性或負面事件的容忍程度。

現今產品的生產型態從以往單一產品大量生產，改變為數量較少而樣式變化多的生產型態。由生產固定數量轉變為接到顧客訂單後，才開始生產的模式，因此零工式的生產隨著經濟需求型態的轉變而變得日益普遍且重要。隨著彈性製造的生產時代來臨，機器的功能變得更多樣且更有效率。因此典型零工式生產排程中，加工步驟指定特定機器執行的限制較不符合實際生產狀況。當加工步驟執行時，若有較多替代機器可供使用，則可以減少許多加工時間或避免等待同一台機器的狀況產生，亦即增加生產製造上的彈性。

然而在現實的製造環境中，決策者所在乎的往往並非是單一的目標，而是彼此相互衝突的多重目標。但是，在多目標排程的文獻中，大都皆僅是考量生產製造上的多重績效衡量，鮮少將公司的策略、顧客重要性所應

考量的因素納入。如此一來，雖然生產製造部門所制定的排程有良好的現場績效，但以營運策略、市場行銷的觀點來看卻不盡理想，往往會導致生產與行銷兩部門的衝突，尤其是在充滿相互矛盾因素的環境中，此問題更顯重要。

本研究中擬以零工式生產型態為基礎，加入平行機器的考量，並允許作業批量分割，使一作業可同時在多部機器處理。此外，同時考量多目標之排程績效指標，除了考量生產現場績效方面的定量因素之外，並將策略類型的定性因素納入考量，構成一完整的多目標排程架構。於重排程研究中，將以特定的排程中斷事件為例，如機台當機、物料短缺與緊急插單等，發展零工式平行機台環境下的重排程方法。此外，為了發展求解出一個對於中斷事件有足夠容忍力或彈性的排程，本研究亦將穩健性(Robustness)與穩定性(Stability)兩項指標，納入排程的考量。

## 1.2 研究目的

雖然有關重排程方法的研究非常的多，但是大多數的研究皆是在流程式工廠或是零工式單一機台的工廠架構下研究，甚少在零工式平行機台的生產型態發展重排程的方法，由於平行機台的考量，重排程時需要考慮批量分割與各平行機台指派的問題，而非單純只是零工式單一機台下的排序問題，因此零工式平行機台的重排程方法與考量也較上述兩類複雜許多。本研究擬以特定的排程中斷事件為例，發展各個中斷事件下的重排程方法。此外，於一般排程的架構下，加入排程穩健性與穩定性的考量，以發展求解出一個對於中斷事件有足夠容忍力或彈性的排程，以降低中斷事件或干擾因素對於生產規劃的負面影響。

基於上述，本研究的重點如下：

1. 於零工式工廠平行機台生產型態下，發展重排程方法與穩健排程(robust schedule)的求解。研究中的重排程方式包括部分重排法(partial rescheduling)與完全重排法(complete rescheduling)，探討之排程中斷因素包含機台故障、物料短缺與緊急插單等。
2. 考量零工式平行機台型態下的作業批量分割與機台指派問題。並以多目標架構結合排程量性因素(交期、製距、機器使用率)、質性因素(公司策略、市場考量、顧客歷史交易等)。

3. 結合排程穩健性指標(Robustness)、穩定性指標(Stability)與一般排程下的多目標，發展求解出一個對於中斷事件有足夠容忍力或彈性的穩健排程，以降低中斷事件或干擾因素對於生產規劃的負面影響。

### 1.3 研究範圍

Vieira(2003)等學者提出了非常完整的重排程架構，其中又分為重排程環境、重排程策略(或政策)與重排程方法三大面，本文重排程研究主要針對重排程的方法進行發展，因此對於重排程的策略面並無探討。

關於重排程環境面。現場排程若以機器數目配合工作劃分，可將排程問題分成數種不同複雜程度的情形(林, 1994)。其複雜程度依序如下：

1. 單機-單階
2. 多機-多階
3. 多階-流程式工廠(Flow shop)
4. 多階-零工式工廠(Job shop)

這四種生產方式以零工式工廠(Job shop)類型最為複雜，屬於NP-hard(Pinedo, 1995)的問題。本研究只以Job shop生產類型的排程問題做為研究範圍，並加入平行機台的考量，允許加工作業批量分割。

### 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，實驗的資料為一假設的Job shop排程問題，並沒有運用實際工廠的資料作為測試。以此排程問題對本研究所提的重排程方法，在多目標衡量下進行實驗，最後再依據實驗數據與結果作分析討論。本研究主要可分為四個步驟：

1. 於多目標零工式平行機台的生產環境下，建立一般的作業排程規劃。
2. 以一般的作業排程規劃方法為基礎，根據不同的排程中斷事件，發展個別的重排程方法。
3. 發展具穩健性與穩定性之排程的求解方法。
4. 以電腦進行實驗，根據以上步驟各方法進行系統驗證。

## 1.5 論文架構

本研究論文的內容共分為五章：第一章說明本研究之動機、目的、範圍、假設、方法與步驟等相關內容；第二章說則針對本論文內容所涉及之相關文獻加以探討。包括遺傳演算法、禁忌搜尋法、多目標問題、重排程、穩健排程之相關文獻；第三章則根據本論文之目的與文獻探討結果，首先建立本研究中一般排程的方法架構，再設計重排程方法與穩健排程之求解方式；第四章為本研究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。

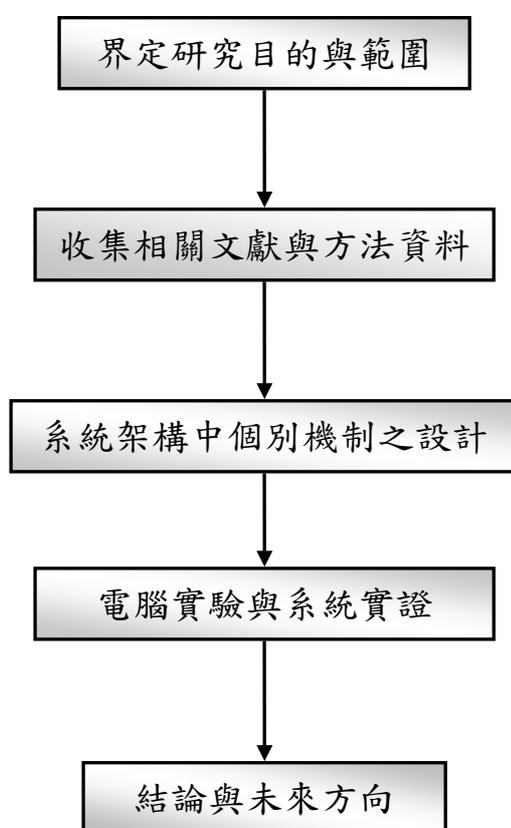


圖 1.1 研究流程圖

## 第二章 文獻探討

本章分為五個部份，第一節探討排程問題之分類與多目標排程問題；第二節描述重排程問題並探討相關研究整理，其中包含重排程方法與穩健排程等相關研究；第三節說明平行機器排程問題與相關文獻；第四節與第五節則說明遺傳演算法與禁忌搜尋法於排程上的應用。

### 2.1 排程問題之描述

排程問題有許多種類型，針對不同的型態排程問題，其問題模式也不盡相同。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。

#### 2.1.1 排程問題之分類

在林(1994)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，常見的交期滿足、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(1966)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(1966)

### 2.1.2 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)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; Daniels, 1994; Neppalli et al., 1996; Murata et al., 1996; Ishibuchi and Murata, 1998; Kim et al., 1998; Min et al., 1998; Chang and Lo, 2001)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)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)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); Min et al. (1998)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); Murata et al. (1996); Neppalli et al. (1996)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); Murata et al. (1994)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); Rajendran and Ziegler (1999); Danneberg et al. (1999); Rajendran and Ziegler (2003); Allahverdi (2003); Lin and Liao (2003)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)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)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)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)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)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 Ensore Ham, NEH)啟發式解法求解。Ghrayeb (2003)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)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)	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)	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)	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)	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) Min et al. (1998)	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)	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)	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

表 2.2 多目標排程文獻整理 (續)

Chang and Lo (2001)	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)	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)	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)	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)	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)	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)	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 1,w
Ishibuchi et al.(2003)	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)	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)	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)	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

表 2.2 多目標排程文獻整理 (續)

Allahverdi(2004)	流程式排程	DCH/SAH/MNE H/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)	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

資料來源：黃(2005)

## 2.2 重排程問題之描述

根據 Vieira et al. (2003)的定義，重排程(rescheduling)是指更新現有的生產排程，以便對於變動或干擾因素作出回應，這些因素包括加入新的訂單作業、機器故障與緊急插單...等等。

### 2.2.1 重排程因素與問題架構

工廠生產排程計畫之擬定，很少有情形是只排一次便能獲得滿意的結果，一般而言需要反覆數次才能完成(吳鴻輝&李榮貴，1997)。其次一個可行之排程計畫一旦決定，幾乎會立即遭遇不可抗拒的改變而要求重新調整排程計畫之狀況，例如緊急插單或是機器故障等，這些會造成原定排程計畫不得不調整的因素，稱之為重排程因素(rescheduling factors)。Vieira et al. (2003)的研究裡指出常見的重排程因素(表 2.3)。在一動態之工廠環境下，這些重排程因素使得重排程不但是無法避免的而且是很花時間(人及電腦)或成本的工作，因此重排程問題有必要深入討論。在吳鴻輝與李榮貴(1997)的研究裡指出，在執行重排程時，基本上是隱含了排程評估(evaluation)、應變方案設計(solution)及排程修正(revision)等三個階段(steps)，其中表 2.4 為常見的重排程方案。

Vieira et al. (2003)提出重排程問題的架構(如圖 2.1)，關於重排程問題的研究，大致可分為三個面向：(1)重排程環境(rescheduling environments)；(2)重排程策略(rescheduling strategies)；(3)重排程方法(rescheduling methods)。

表 2.3 常見的重排程因素

1. Machine Failure
2. Urgent (rush, or ‘hot’) job arrival
3. Job cancellation
4. Due date change (delay or advance)
5. Delay in the arrival or shortage of materials
6. Change in job priority
7. Rework or quality problems
8. Over- or underestimation of process time
9. Operator absenteeism

(資料來源：Vieira, et al., 2003)

表 2.4 常見的重排程方案

1. 加快製程速度
2. 增加人手
3. 加班
4. 加工中(in-process)外包
5. 使用替代途程(機器)
6. 延後交期
7. 作業重疊
8. 作業分割
9. 訂單分割
10. 作業間閒置時間縮短
11. 相依設置時間之壓縮
12. 使用不同法則重排

(資料來源：吳鴻輝&李榮貴，1997)

Rescheduling environments				
Static (finite set of jobs)		Dynamic (infinite set of jobs)		
Deterministic (all information given)	Stochastic (some information uncertain)	No arrival variability (cyclic production)	Arrival variability (flow shop)	Process flow variability (job shop)

Rescheduling strategies				
Dynamic (no schedule)		Predictive-reactive (generate and update)		
Dispatching rules	Control-theoretic	Rescheduling policies		
		Periodic	Event-driven	Hybrid

Rescheduling methods				
Schedule generation		Schedule repair		
Nominal schedules	Robust schedules	Right-shift rescheduling	Partial rescheduling	Complete regeneration

圖 2.1 重排程架構

### 2.2.2 重排程方法

Vieira et al. (2003)主要探討的重排程方法包含兩大類，第一類是既有生產排程修正(Schedule repair)的方法；第二類則是產生穩健生產排程(Robust schedule)的方法。本文就排程修正方法與穩健排程之文獻整理說明如下。

#### 1. 排程修正(Schedule repair)

排程修正方法又細分為三種方式，第一種是 right shift 重排程方法，主要是延後尚未執行的工作，以便騰出處理排程中斷的時間，直到生產排程再度變得可行為止。第二種則是部分重排(partial rescheduling)方法，此種方法只重排有受到排程中斷事件影響的工作，盡量使重排程的結果和初始排程的結果異動最小。第三種種類型則是完全重排方法(Complete regeneration)，也就是完全重新產生新的生產排程，包括沒有受到排程中斷影響的工作，雖然可以獲得較佳的排程結果，但主要的缺點是太過於浪費運算資源與回應時間太久。

本文中，關於排程修正方法的文獻(Bierwirth and Mattfeld, 1999; Akturk and Gorgulu, 1999; Guo and Nonaka, 1999; 林虹谷, 2002; Rangsaritratamee et al., 2003; Jensen, 2003; Azizoglu and Alagoz, 2005; Hu et al., 2005)，整

理如表 2.5。

表 2.5 排程修正方法文獻整理

作者	排程型態	排程修正方法	研究概要與貢獻
Bierwirth & Mattfeld (1999)	零工式排程	完全重排	考量新訂單到達後的重排程問題，以重新利用前初始解的方式，由GA演算法求解重排程。
Akturk & Gorgulu(1999)	流程式排程	部分重排	考慮機台當機情境，以媒合(match-up)方式修復初始排程。
Guo & Nonaka(1999)	流程式排程	部分重排	以機台當機為例發展重排程方法，其中重排程方針是針對不同的當機時程長度來決定，研究目標主要在於降低機台故障對於排程的影響。
林虹谷 (2002)	流程式排程 & 零工式排程	完全重排	將重排程問題表現成限制條件滿足問題(CSP)，再利用各種相關的CSP求解方法，如Search Algorithms，求解重排程問題。
Rangsaritratsamee et al. (2003)	零工式排程	完全重排	建立重排程方法，以 GA 演算法求解，同時考量排程效率性指標(efficiency)與穩定性指標(stability)。
Jensen (2003)	零工式排程	right-shif、 部分重排 & 完全重排	以機台當機為例，利用數種不同重排程方法進行重排，並比較重排後的績效指標。
Azizoglu & Alagoz (2005)	單階 平行機台	完全重排	考量機台當機問題，其中以總流程時間為效率目標，以重排程後受到干擾的作業數量為穩定性目標，重排程方式以啟發式解法求得最佳解。
Hu et al. (2005)	流程式排程	完全重排	考量多筆緊急訂單的重排程問題，其中多筆緊急訂單必須和前排程中尚未處理的作業一起納入重排，求解過程以總流程時間的加權平均值為目標函數，而緊急訂單之權重則較其他訂單作業來得高。

## 2. 穩健排程(Robust Schedule)

求解或規劃穩健排程的主要目的是在於，讓排程遭遇變異或中斷事件

時，仍然能維持好的排程系統績效，使一個具有穩健性或彈性的排程能夠吸收變異或排程中斷造成的負面影響。有關於排程穩健性(Robustness)的定義，Dooley and Mahmoodi (1992)；Jensen (2003)等學者的研究中定義排程穩健性為系統遭遇變異時仍可運作良好的能力。Velagapudi (1992)定義排程穩健性為排程可消除(eliminate)或最小化(minimize)變異的能力。Sevaux and Sörensen (2002)指出穩健性相似於彈性，一具有彈性的排程應該較容易修正，例如：將初始排程依變異情況重排修正成為一個新的高品質排程。Sörensen (2002)定義排程品質的穩健性為排程目標函數值的敏感度。舉例來說，若生產活動發生若干干擾或變異時，排程仍然能維持高的生產績效(目標函數值)，則稱此排程具有穩健性。Artigues et al. (2005) 指出排程系統的穩健性可定義為其可吸收預期中斷事件的能力。

關於穩健排程建立方法與文獻中，Leon et al. (1994) 利用 GA 演算法產生穩健的排程，最小化預期的延遲(delay)或製距(makespan)。Byeon et al. (1998) 將一零工式排程問題分解成數個子問題，再以啟發式演算法分別求解各個部份排程(partial schedule)。Wu et al. (1999) 使用 branch and bound 演算法處理部分排程(partial schedule)相關的 disjunctive graph。Mehta & Uzsoy (1998) 使用 shifting bottleneck 演算法，建立預測式排程，在其間插入閒置時間(idle time)以減小排程中斷的衝擊。Herrmann(1999) 利用 GA 演算法最佳化最差(worst-case)的績效生產排程。Shafaei & Brunn (1999) 研究在動態的零工式工廠建立強健排程的法則(rules)。Sevaux & Sörensen (2002) 於單機排程問題中，以 GA 演算法求解穩健排程，其中各作業所需零件的到廠日是變動的。Sörensen (2002) 定義評估穩健排程的目標函數，以禁忌搜尋法(tabu search)求解。Jensen (2003) 於零工式排程問題中，定義穩健性衡量指標，考量排程的彈性與穩健性，利用 GA 演算法求解低製距的穩健排程。

### 2.3 平行機器排程

平行機器排程問題之作業方式，將以往生產或處理的工件僅能經過單一的機器處理才可完成，變更為同時有多部機器可供使用，而每一工件之處理可在任一機器上執行。此種生產型態廣泛出現於各種產業之多條平行生產線或彈性製造中心，例如在工廠多條生產線、服務業速食店的收銀台等都可視為一平行機器的生產方式。

針對平行機器排程問題，總時程最小化、總流程時間最小化和總延遲

時間最小化，均是屬於 NP-hard 問題的範圍(Baker, 1984)。現有文獻大多探討相同機器或等效機器之排程問題，就問題複雜度而言，不相關機器之問題較相同或等效機器之問題更為困難。

利用平行機器來加工作業的生產型態目前已被廣泛地運用於各種產業中，因此有關平行機器的作業排程問題一直受到相當的重視。所謂平行機器排程是在探討工作中心(work center)中會設置一部以上功能相同的相似機器的排程問題，每部機器皆可執行同一加工作業，排程的目的便是要妥善地利用多部機器，將工件安排到各個機器中，並決定工件的加工順序，以達到所希望績效指標的最佳化。

### 2.3.1 平行機器的類型

Baker(1974)定義平行機器為「生產的工件，僅需經過單一加工途程即能完成，而同時有多部機器可供利用，此即為平行機器的生產型態」。平行機器依其機器間之功能差異程度大致上分為三類(Cheng及Diamond, 1995)：

1. 相同機器 (identical machine)：所有的機器均相同，同一工作在各個機器上所需的加工時間皆相等。此類問題是屬於較典型的平行機器排程問題，考量生產現場中的機器都一樣。
2. 非等效機器 (uniform machine)：同一工作在各個機器上的加工時間會因機器加工速度而不同。此類型的問題考量到機器加工速度的差異，例如工廠採購新型的機器後，新型與舊型機器同時為同一製程加工時，由於新舊機器的性能不同，造成加工速度效率上的差異，便衍生出此類的問題。
3. 不相關機器 (unrelated machine)：同一工作在各個機器加工時，工作在不同機器上所需的加工時間不相等，其加工時間並無任何關聯，而機器對不同的工作的加工速率也互有差異。此類的問題包括某些工作只限於在某些機器上加工；某些工作在某些機器上的加工速率會優於其他機器，處理此類問題時，通常難度會比較高。

### 2.3.2 單階平行機器排程

關於單階平行機器排程的文獻整理如表2.6。Guinet(1995)先發展一套啟發式解法，然後再利用模擬退火法改善平均延遲時間最小化或加權延遲時

間總和之排程問題。Serafini(1996)機器的類型上探討工作可以分割，求解以最大加權延遲時間最小化為績效指標之問題。Piersma及Van(1996)針對不相關機器以局部搜尋法（塔布搜尋法、模擬退火法、遺傳演算法），求解總時程最小化的排程問題。Suresh及Chaudhuri(1996)考量了最大完工時間與最大延遲時間最小化兩個績效指標，他們發展出以塔布搜尋法為基礎的啟發式解法。Cheng等(1996)則對不同的工作給予不同的權重值，以區域搜尋法和遺傳演算法求解，使加權的最大延遲完成時間之和最小化。Gürsel等(1997)則考慮機器有整備時間的問題，並針對一工作可以分割至不同機器上加工之情況求解。Tamimi等(1996)針對總加權延遲時間最小化排程問題，以遺傳演算法求解。Srivastava(1998)則以最大完工時間最小化為績效指標，而發展出一個以塔布搜尋法為基礎的啟發式解法。Min及Cheng(1999)以遺傳演算法，求解總完工時間最小化為績效指標的問題。Serifođlu及Ulusoy(1999)考量加權的工作延遲及提早完成時間之和最小化，並假設工作都有不同的到達時間、加工時間、設定時間與交期，發展以遺傳演算法為架構的求解方法。林(2001)則發展啟發式演算法與分枝界限法來求解不相關機器的總加權延遲時間最小化問題。

### 2.3.3 多階平行機器排程

關於多階段平行機器的排程研究中，大多是探討流程式生產型態，而其求解方法以使用最佳解和啟發式解的技巧為主。多階平行機器排程問題為一典型的NP-complete問題(Garey及Johnson, 1979)(Gupta, 1988)。Nowicki及Czeslaw(1998)用塔布搜尋法求解，使總排程時間最小化；Sridhar及Rajendran(1993)則用模擬退火法求取最小的總流程時間。Jacques等(1996)用分枝界限法求工作數不大時之解。Brah等(1991)發展了具平行機器之流程式生產型態的基本數學模型及其延伸來解其他的排程問題。Santos等(1995)建立了多階平行機器流程式排程問題中，最小總排程時間的全域最小界限（Global Lower Bound）。而針對多階平行機器流程式排程問題所發展出的啟發式法則，比較著名的有Nawaz(1983)的The Nawaz Heuristic與Campbell等人的The Campbell, Dudek, and Smith (CDS) Procedure，及Hundal和Rajgopal(1988)與Santos等(1995)的啟發式法則。

表 2.6 單階平行機器排程文獻整理

作者	年份	平行機器類型	求解方法	績效衡量指標
Guinet, A.	1995	非等效機器	啟發式解法 模擬退火法	平均延遲時間最小化
Serafini, P.	1996	非等效機器		最大加權延遲時間最小化
Piersma, N., and Van Dijk, W.	1996	不相關機器	局部搜尋法 (塔布搜尋法、 模擬退火法、 遺傳演算法)	總時程最小化
Suresh, V. and Chaudhuri, D.	1996	不相關機器	塔布搜尋法為基 礎的啟發式解法	最大完工時間最小化 最大延遲時間最小化
Cheng, R. and Gen, M.	1997	相同機器	區域搜尋法 遺傳演算法	加權的最大延遲完成時間 之和最小化
Gürsel, A. S., et al.	1997	相同機器	最佳化方法	延遲工作數最小化
Tamimi, S. A. and Rajan, V. N.	1997	非等效機器	遺傳演算法	總加權延遲時間最小化
Azizoglu, M. and Kirca, O.	1998	相同機器	分枝界限演算法	總加權延遲時間最小化
Srivastava, B.	1998	不相關機器	塔布搜尋法為基 礎的啟發式解法	最大完工時間最小化
Min, L. and Cheng, W.	1999	相同機器	遺傳演算法	總完工時間最小化
Serifođlu, F. and Ulusoy, G.	1999	相同機器	遺傳演算法	加權的工作延遲及 提早完成時間之和最小化
林暘桂	2001	不相關機器	啟發式演算法 分枝界限演算法	總加權延遲時間最小化

資料來源：鍾(2004)

## 2.4 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(1989)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(1994)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將不同的問題特徵（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(1989)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。Murat(1994)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(1994)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(1989)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 2.2。

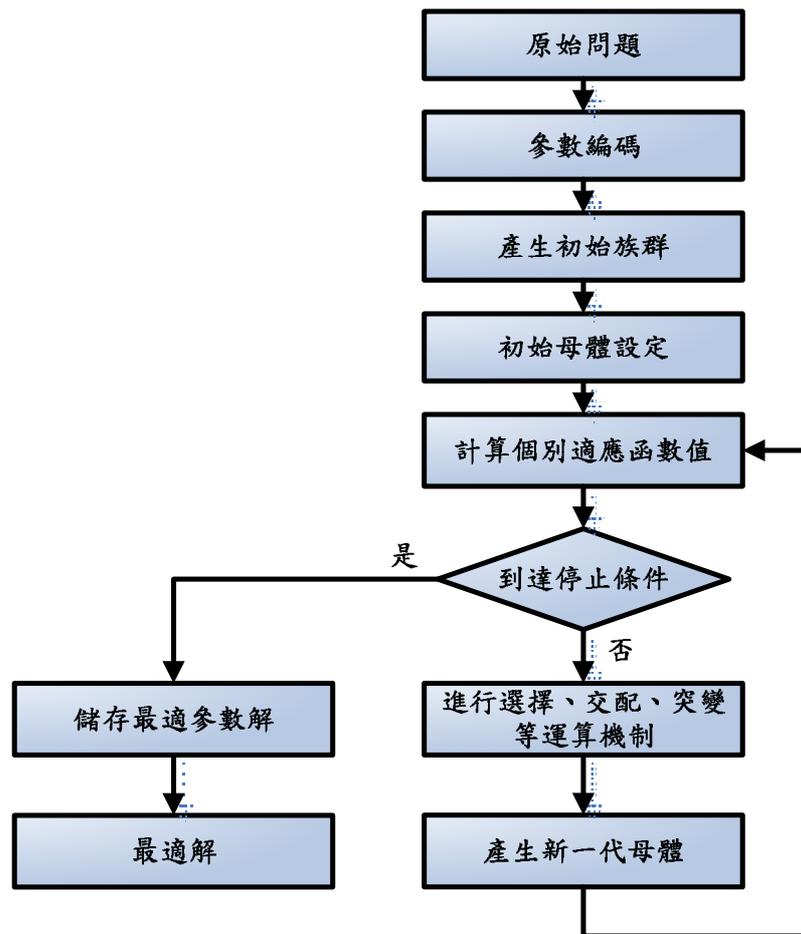


圖 2.2 一般遺傳演算法運作流程

### 2.4.1 常見 Job shop 排程問題的編碼表示法

在 Cheng(1997)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(  $m_1, m_2, m_3$  ) :  $X_{121}$   $X_{122}$   $X_{123} = 0$  1 0

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

訂單( 2, 3 )在機器(  $m_1, m_2, m_3$  ) :  $X_{231}$   $X_{232}$   $X_{233} = 1$  1 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 2.3）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形成一循環(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 2.3 為例，其基因如下：

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

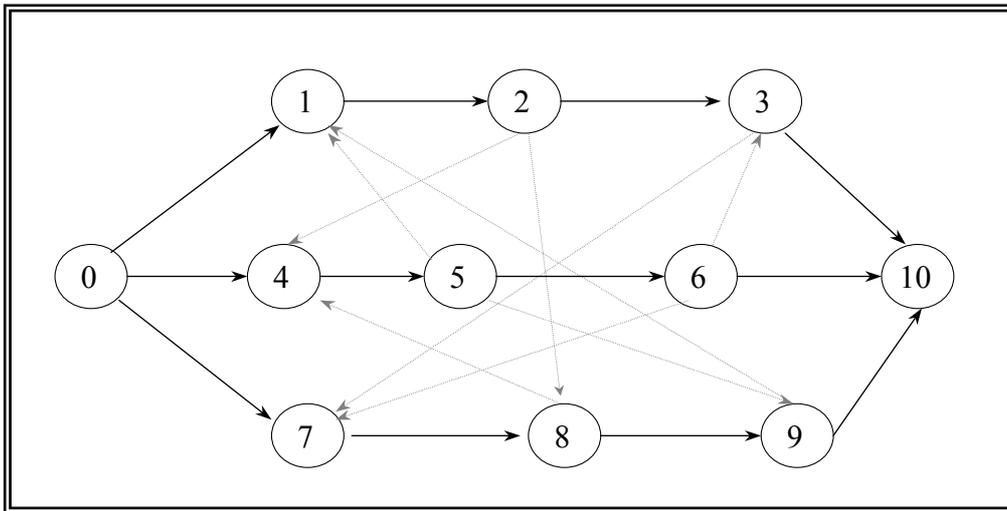


圖 2.3 分支圖

## 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 1.09 1.88 2.66 2.91 2.01 3.23 3.21 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：

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

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。

## 2.4.2 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 Fang (1994)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.均勻交配(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 ]

在吳信儀(1996)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 ]

### 2.4.3 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)。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)以 GA 為基礎發展一啟發式搜尋技術解決 Job shop 的排程問題。Pinedo et al. (1995)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)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)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。Murata et al. (1996)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)利用隨機法則，發展出一套方法，並應用在汽車工業。Liaw (2000)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)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)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 2.7：

表 2.7 遺傳演算法之優缺點歸納表

優點	<p><b>1. 多點同步搜尋</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. 使用適應函數</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. 機率式的搜尋</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. 基因編碼</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>
缺點	<p><b>1. 運算子的設計問題</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. 重複搜尋的問題。</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>

## 2.5 禁忌搜尋法

### 2.5.1 禁忌搜尋簡介

禁忌搜尋法又名塔布搜尋法，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(1989)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(1990)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 list)，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(1990)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 criterion)：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更佳的解。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。

### 2.5.2 禁忌搜尋在排程上之應用

Wesley(1995)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(1996)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(2000)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。

### 第三章 多目標平行機台零工式工廠重排程架構

本研究主要於零工式平行機台的架構下，建構重排程的方法與穩健排程的求解，系統以多目標適應函數同時考量質性與量性目標，以混合式遺傳演算法求解一般排程。在重排程中，系統以機台故障、緊急插單、物料短缺三種情境作為排程中斷事件，受中斷後的排程以部份重排法與完全重排法進行重排，期望以部份重排法快速修正排程規劃，以完全重排法取得最適目標函數值的重排程解。

在穩健排程中，以混合式遺傳演算法配合本研究的測試性干擾機制，以求得穩健排程解，期望增加排程對於中斷事件的容忍程度。穩健解於排程上的涵義是指，當生產活動發生若干干擾或變異時，目標函數仍能維持高的數值。在實際生產的動態環境中，干擾情況時常發生，排程結果變更頻繁，若排程經過中斷干擾並重排後，仍然可維持不錯的績效指標，則此排程對於生產績效的維持穩定可有很大的幫助。

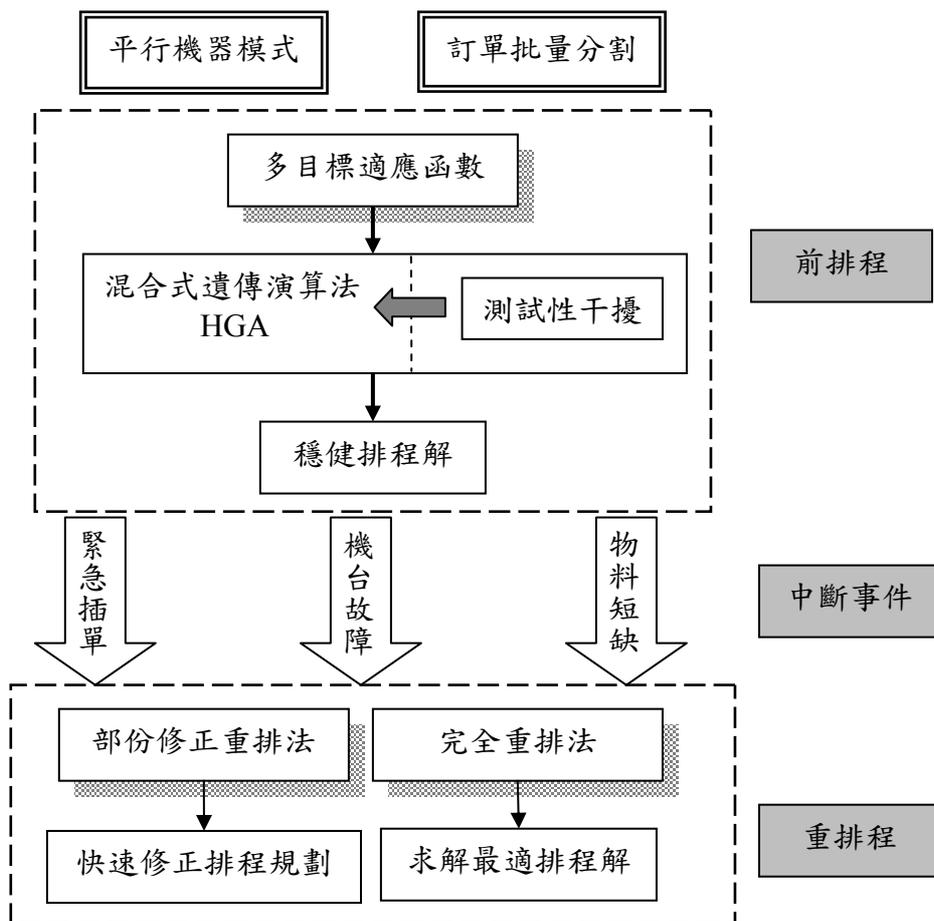


圖 3.1 多目標平行機台零工式工廠重排程架構

本研究各項詳細流程與方法於本章各節說明。3.1 節介紹本研究的基本環境，說明本研究的多目標模式架構與零工式平行機台生產環境；3.2 節說明本系統的基本排程方法與步驟流程；3.3 節說明排程經歷機台故障、物料短缺與緊急插單等中斷事件後，如何以部份修正重排法或完全重排法進行重排程。3.4 節說明於求解過程中加入測試性干擾機制，搜尋具穩健性與穩定性的穩健排程解，提升排程對於中斷事件的容忍程度。

### 3.1 多目標模式與零工式平行機台生產環境

本節介紹本研究的基本環境架構，首先說明系統結合量性因素與質性因素的多目標模式，3.1.1 小節說明定性因素模式架構；3.1.2 小節說明定量因素模式架構；3.1.3 小節說明平行機台零工式工廠的模式架構。

在現實的排程環境中，生管排程人員所需考量的因素非常的多，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質性因子，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。本研究於排程目標上，結合量性因素與質性因素。量性排程因素依據排程結果（製距、機器使用率、交期滿足度），利用 AHP 法則算出相關權重，制訂本研究之量性多目標適應函數。質性排程因素主要依據排程規劃人員評估各訂單間相對重要性，利用 AHP 法則算出相關權重，決定出訂單的優先順序，並依此順序訂出排程時違背此順序的懲罰函數。之後再依據此兩方面函數決定本研究多目標函數。

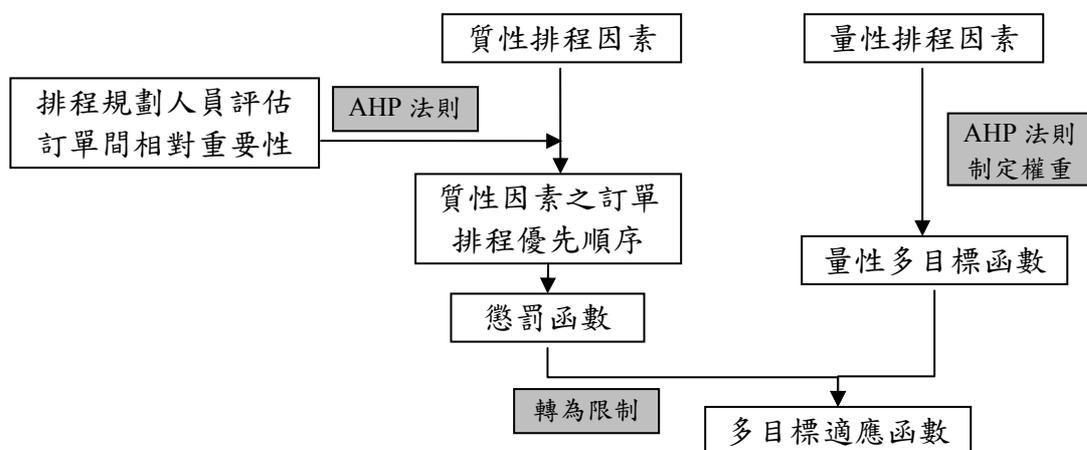


圖 3.2 多目標函數架構圖

### 3.1.1 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。

#### 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(1977)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 3.1。

表 3.1 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 3.2。

表 3.2 各因素之權重制訂

因素	$F_1$	$F_2$	$F_3$	$F_4$	特徵值 $\lambda_{\max}=4.11$	權重 $W_i$
$F_1$	1	1/3	2	7	0.3773	0.24
$F_2$	3	1	5	9	0.8976	0.58
$F_3$	1/2	1/5	1	5	0.2192	0.14
$F_4$	1/7	1/9	1/5	1	0.0628	0.04

步驟五：規劃人員在因素  $F_i$  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  $F_i$  下的評估值( $E_{ij}$ )，如表 3.3。其中  $E_{ij}$  表示訂單  $j$  在因素  $i$  的評估值。

表 3.3 各方案之評估值

$F_i$	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 $E_{ij}$
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( $E_{ij}$ )。

步驟七：計算各訂單的加權值，並加以排序，如表 3.4。

表 3.4 各方案之加權排序

$F_i$	$F_1$	$F_2$	$F_3$	$F_4$	加權值	排序
$W_i$	0.24	0.58	0.14	0.04		
評估值	$E_{1j}$	$E_{2j}$	$E_{3j}$	$E_{4j}$		
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，之後利用此 AHP 所求得的質性訂單順序，配合遺傳演算法或混合式遺傳演算法求解出的訂單順序，計算出懲罰函數值，將此懲罰函數值轉換為目標函數值中的限制。懲罰函數詳細計算與整體適應函數的設計於 3.2 節中說明。

### 3.1.2 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。

## 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 3.1

$$\frac{\min MS}{MS_x} \quad \text{式 (3.1)}$$

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。

## 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 3.3。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求出交期滿足度，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：

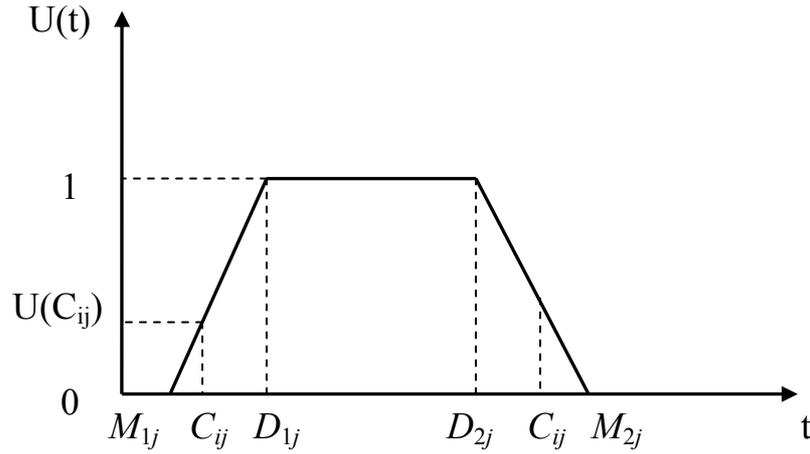


圖 3.3 交期之模糊隸屬函數

$$\left\{ \begin{array}{ll} U(C_{ij}) = \frac{C_{ij} - M_{1j}}{D_{1j} - M_{1j}} & , \quad M_{1j} \leq C_{ij} \leq D_{1j} \\ U(C_{ij}) = 1 & , \quad D_{1j} \leq C_{ij} \leq D_{2j} \\ U(C_{ij}) = \frac{M_{2j} - C_{ij}}{M_{2j} - D_{2j}} & , \quad D_{2j} \leq C_{ij} \leq M_{2j} \\ U(C_{ij}) = 0 & , \quad C_{ij} \leq M_{1j} \text{ or } C_{ij} \geq M_{2j} \end{array} \right. \quad \text{式 (3.2)}$$

$$Y_i = \frac{\sum_{j=1}^n U(C_{ij})}{n} \quad \text{式 (3.3)}$$

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

$D_{1j}$ ：訂單  $j$  之最早達交時間

$D_{2j}$ ：訂單  $j$  之最晚達交時間

$M_{1j}$ ：訂單  $j$  之交期下限

$M_{2j}$ ：訂單  $j$  之交期上限

$Y_i$ ：個體解  $i$  之交期評估值

### 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。

#### 3.1.3 零工式工廠平行機台生產環境

傳統的零工式工廠(Job shop)排程問題研究中，定義每一階作業所需的加工機器種類均只有一台機器。然而現今工廠的生產型態卻不是這樣，為了能夠提高產能及減少瓶頸工作站的負荷，均會在工作中心(work center)內設置功能相同的平行機器，訂單在不同工作中心間流動，在各工作中心內進行不同的加工程序(operation)，藉著訂單作業的批量分割，一作業可同時在多部機器處理，而縮短各作業之完成時間，達到交期準確的目標。本研究之零工式工廠平行機台的生產環境示意圖如圖 3.4，圖中工作中心內劃影線之圓圈，代表被排入該批量作業生產的機器。

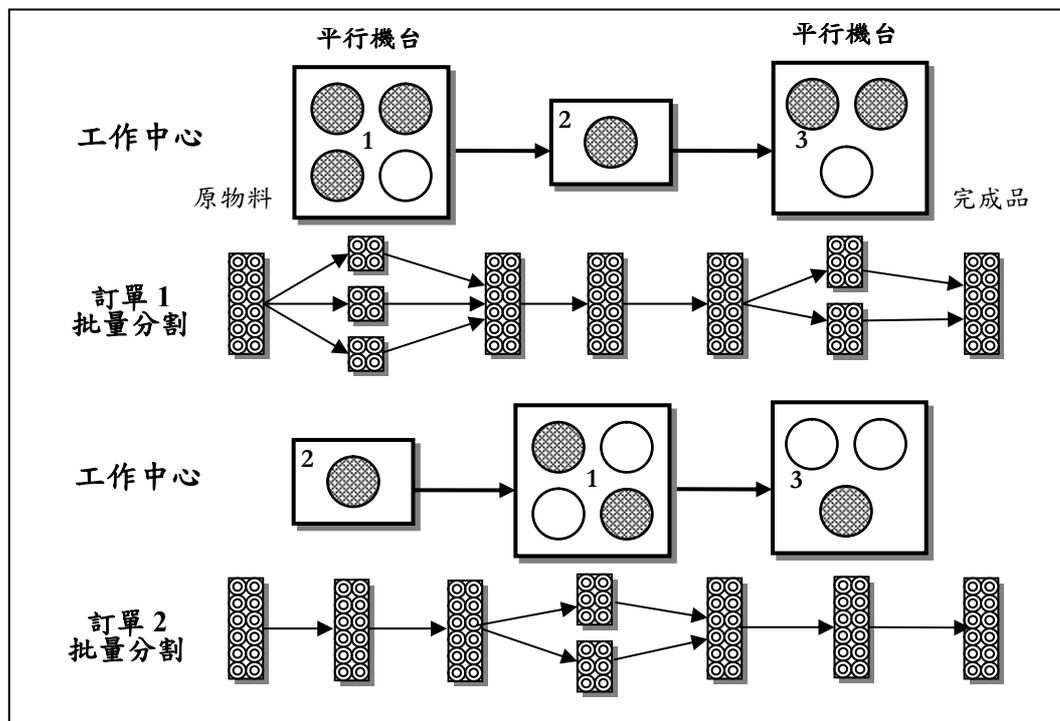


圖 3.4 零工式工廠平行機台與訂單批量分割示意圖

現實生產環境中，每一個作業開始生產前可能皆需經過換模具或是相關機器整備的處理，因此本研究將作業的整備時間與加工時間分開計算，其機器整備與作業加工時間的示意如圖 3.5。

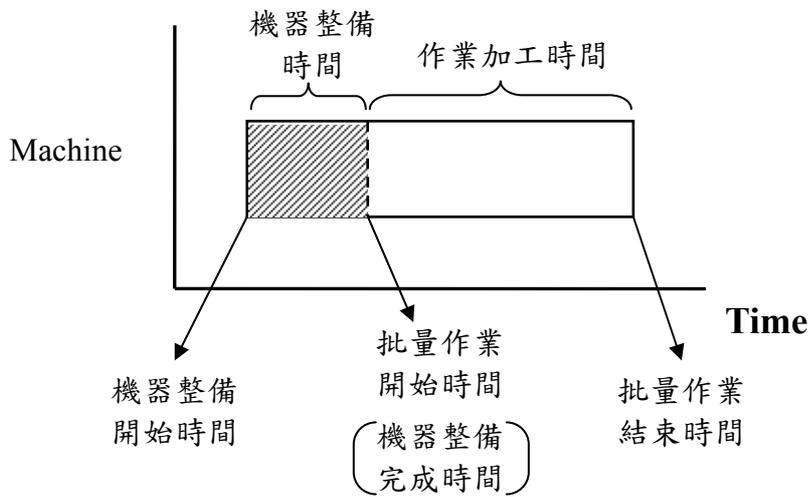


圖 3.5 機器整備時間示意圖

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與機器整備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。

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，但在多階平行機器且允許批量分割的環境之下，每個批量均須對所有平行機器進行判斷，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，並考慮為數眾多的平行機器問題，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. 每一個作業開始前皆需要機器整備時間。
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. 在加工機器上，某些時段已有加工作業在其上加工中。
10. 若該作業於目前工作中心所完成的批量之總單位數，大於下一途程分割之批量大小，則可先將此完成的批量運送至下工作中心進行加工，運送時間忽略不計。
11. 在具有多部機器的工作中心中，每部平行機器皆被視為相同機器。
12. 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。

## 3.2 排程演算法

零工式工廠的生產環境中，由於每個訂單皆有多個途程，且每個作業途程加工時間、各訂單交期時間也不同，因此各訂單作業的加工先後順序決定了一個排程績效的好壞，所以排程規劃人員最主要的工作即是在於決定每個訂單作業於每個機台的加工先後順序。而本研究中平行機台的架構，更增加了排程的複雜度，排程規劃人員除了須決定各作業的先後順序之外，更須規劃出該作業須分割的批量數與加工機台的指定。

本研究的排程演算法是以遺傳演算法為基礎，於演算法中首先定義各染色體基因的編碼，配合本研究所設計的適應函數，以遺傳演算法的各機制演化求解。過程中每次適應函數計算前，皆需要將基因碼解碼為實際排程資訊才能計算，解碼方法則以本研究的排程法則進行解碼。此外，為了提升遺傳演算法的求解品質，本研究結合禁忌搜尋法與遺傳演算法來求解排程解。遺傳演算法的運作於本節的 3.2.1 小節說明，解碼排程法則於 3.2.2 小節說明。

### 3.2.1 遺傳演算法的運作

本小節說明研究中關於遺傳演算法的設計及其運算過程，其中遺傳演算法個體分為上半部與下半部基因，本小節首先就編碼方式說明如下：

#### 1. 上半部基因編碼

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

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。上半部基因數目等於所有訂單作業數總和。上半部基因的編碼示意圖如圖 3.6。

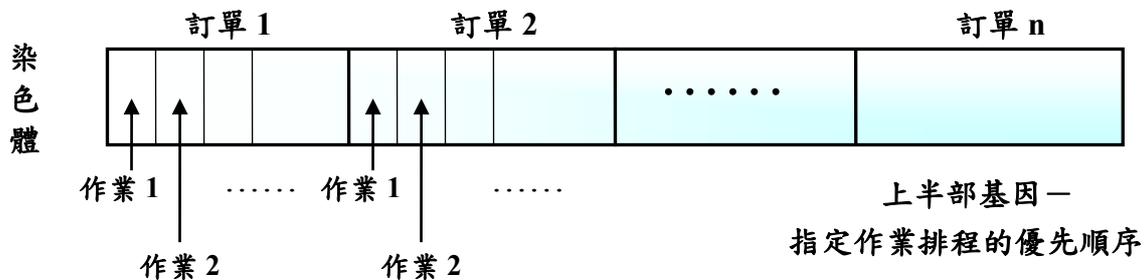


圖 3.6 上半部基因編碼表示

## 2. 下半部基因編碼

個體中下半部的基因，每一基因代表該批量作業是否於該平行機台加工，若該批量作業於此機台加工則以 1 表示，若沒有則以 0 表示。舉例來說，若訂單 1 的作業 1 之基因碼為 011，則表示訂單 1 的作業 1 將分割成兩個批量作業加工，並於其所屬途程工作中心的機台 2 和機台 3 進行加工。

若某訂單的某項作業之基因碼為 111，則表示此作業將分割成三個批量作業加工，並於其所屬途程工作中心的機台 1、機台 2 與機台 3 進行加工。下半部基因的編碼示意圖如圖 3.7，每個訂單每項作業的基因碼皆以此表示。而每項作業的基因長度  $k$ ，則由所有工作中心中最大的平行機台數來決定。例如工作中心 3 擁有最多的平行機台，數目為 4 台，則所有訂單所有作業的基因長度皆為 4。

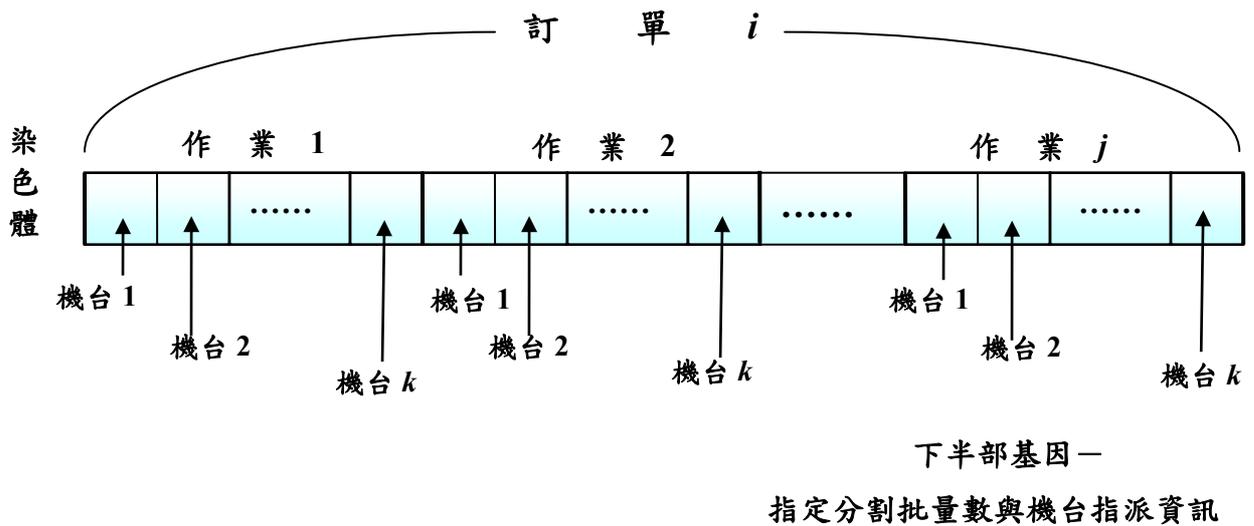


圖 3.7 下半部基因編碼表示

### 3. 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。假設所有訂單的總作業數共為  $N$  個作業，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 3.8。



圖 3.8 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 的互換之後的個體即為一理解。

下半部基因代表的是作業的分割批量數與排程的機台佔有資訊。假設平行機台數最大值為 4，以圖 3.9 為例，初始解的產生上，每個機台的指派隨機以 0 或 1 的數值填入。0 代表該作業無排入此機台生產，1 代表該作業

有排入此機台生產。以訂單 2 的作業 2 來看，其基因碼為[0101]，表示該作業將分成兩個批量生產，分別於其途程工作中心的機台 2 與機台 4 加工。

	作業 1				作業 2				...	作業 m			
訂單 1													
訂單 2					0	1	0	1					
...													
訂單 n									...				

隨機方式  
填入 0 或 1

圖 3.9 下半部基因初始族群示意圖

同樣地，以此方式產生下半部機因的初始族群，會有不理解的現象發生，其不理解的產生有兩種情況。情況一，舉例來說，當某作業的基因碼為[0111]，但此作業途程的工作中心只擁有 2 台平行機台，此時則將不存在的機台碼刪除，修正其基因碼成[0100]。情況二，舉例來說，假設某訂單的總批量為 120，其作業 1 的上機批量限制為 35，當其基因碼為[1111]時，代表該作業要分為 4 個批量加工，此時每台機器上的加工批量變成  $120/4=30$ ，沒有達到上機批量限制，此為一不理解，修正方式則是隨機挑選一機台，將其基因碼修改為 0，即逐一減少分割批數，直到符合上機批量限制為止。在初始解產生後，上述兩種情況皆可能發生，因此在修正上，統一先將不存在的機台碼刪除，再檢查其是否符合上機批量限制，若不符合則依照上述方法修正。

#### 4. 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後，即可算出其與質性訂單順序的差異，並以其作為懲罰函數值。

#### 懲罰函數之制訂

在遺傳演算法運作的過程中，我們是以 3.1.1 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則是先計算排序後的訂單順序與定性因素訂單順序之間的加權後差異平方總和，再將其除以可能的加權後最大差異平方總和，以正規化之。而最大差異平方總和則是以最大違反定性順序的訂單順序所計算出的。計算取加權之原因，則是為了讓重要的訂單更顯示其重要程度，不重要的訂單在排程時若違反定性順序也無妨。詳細的懲罰函數計算步驟舉例如下：

假設目前有 5 張訂單，這些訂單經 AHP 求得之權重如表 3.5，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，加權差異平方和之計算如表 3.7。

表 3.5 各訂單之加權值

$F_i$	$F_1$	$F_2$	$F_3$	$F_4$	加權值	排序
訂單 1	0.1061	0.15	0.15	0.35	0.1745	4
訂單 2	0.0668	0.20	0.25	0.25	0.1770	3
訂單 3	0.4055	0.30	0.10	0.15	0.2913	1
訂單 4	0.1648	0.15	0.20	0.10	0.1586	5
訂單 5	0.2568	0.20	0.30	0.15	0.2256	2

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 加權後差異平方總和之計算

訂單編號	1	2	3	4	5	加權差異平方總和
訂單加權值	0.1745	0.1770	0.2913	0.1586	0.2256	
定性因素訂單順序	4	3	1	5	2	
定量因素訂單順序	1	2	3	4	5	
差異平方	$3^2$	$1^2$	$(-2)^2$	$1^2$	$(-3)^2$	
加權差異平方	1.5705	0.1770	1.1652	0.1586	2.0304	5.1017

加權差異平方的總和為  $1.5705+0.1770+1.1652+0.1586+2.0304=5.1017$

表 3.8 加權後最大差異平方總和之計算

訂單編號	1	2	3	4	5	加權最大差異平方總和
訂單加權值	0.1745	0.1770	0.2913	0.1586	0.2256	
定性因素訂單順序	4	3	1	5	2	
最大違反訂單順序	2	3	5	1	4	
差異平方	$2^2$	$0^2$	$(-4)^2$	$4^2$	$(-2)^2$	
加權差異平方	0.6980	0	4.6608	2.5376	0.9024	8.7988

依照定性因素的訂單順序，找出最大違反定性順序的訂單順序，以求出最大差異平方和。本例依表 3.5 的定性順序，找出的最大違反訂單順序為：

( 訂單 4 ) → ( 訂單 1 ) → ( 訂單 2 ) → ( 訂單 5 ) → ( 訂單 3 )

以此最大違反的訂單順序與質性因素訂單順序，求出加權後最大差異平方的總和，如表 3.8。

將加權後差異平方總和除以加權後最大差異平方總和，以正規化後，所得懲罰函數值如下：

$$5.1017 / 8.7988 = 0.5798$$

本研究利用 AHP 法計算出定性因素與定量因素權重，整體適應函數值計算如下式

$$f(x) = W_1 \cdot (\sum \alpha_i \cdot f_i) + W_2 \cdot (1 - p(x))$$

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。

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

## 5. 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，如下

$$P(x_i) = \frac{f(x_i)}{\sum_i f(x_i)}$$

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，本研究中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。

## 6. 交配運算子設計

由於染色體中包含了訂單之作業排程的優先順序與其加工途程所指定的工作中心中佔有的平行機台資訊，因此一般常用的交配運算子已無法使

用。本研究的交配運算分為上半部基因的交配與下半部基因的交配。上半部基因屬於排序數值，需採用排序問題的交配運算子，本研究引用 Bean(1994)所提出的 Random Key Representation 修正重新編碼與交配方法並加以修改；而下半部基因的交配則以一般的兩點交配處理。上半部基因的交配步驟如下：

- 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：在選取到的染色體上的每個基因，以 0 至 1 的均勻分配隨機產生一數值，依照原基因由小大到重新代表之，即重新以一個 0 至 1 的隨機數編碼為此基因值。
- 步驟三：以兩點交配方式，隨機選取兩點作為交配點，互換兩交配點之間的基因值。
- 步驟四：將染色體各基因值由小到大重新排序編號，即步驟二重新編碼後的解碼。完成此兩條染色體的交配運算。
- 步驟五：重複步驟二~四，直到所有子代染色體產生完畢。

本研究上半部基因的交配舉例如下圖 3.10：

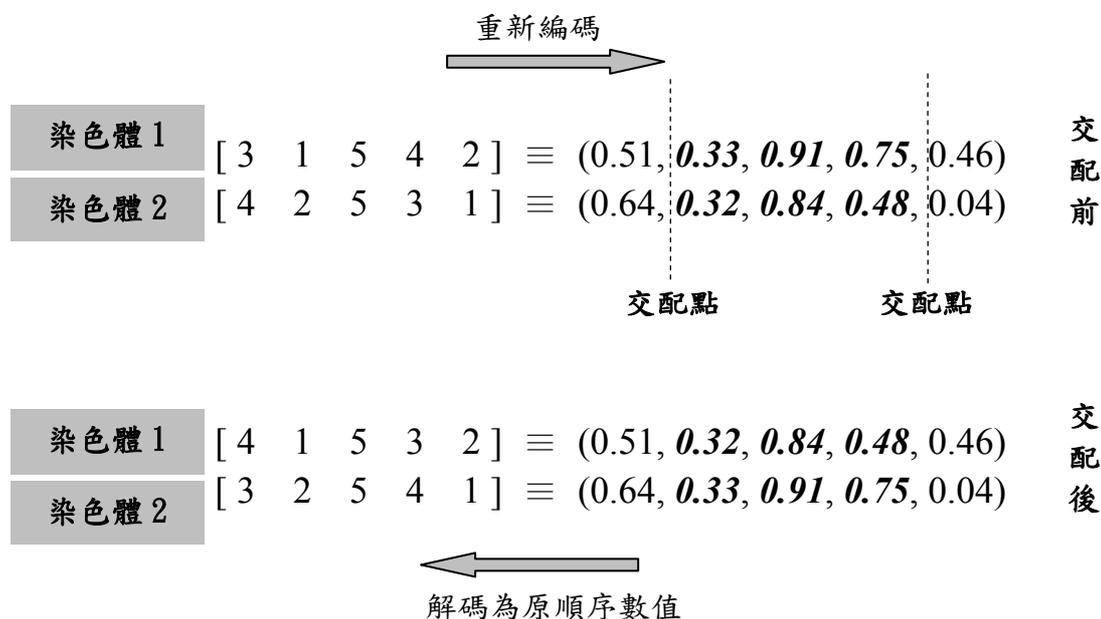


圖 3.10 上半部基因交配示意圖

## 7. 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平行機器資訊，因此本研究採取以下突變方法，分為上半部基因的突變與下半部基因的突變。

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。

下圖 3.11 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。



圖 3.11 上半部突變示意圖

下半部基因的突變仍是對每一基因均給予突變機率來決定是否需要進行突變，若該基因決定突變，則基因碼由原本的 0 變為 1，或是由原本的 1 變為 0。

### 3.2.2 排程法則

本小結排程法則即為基因演算法的解碼流程。在本排程演算法中，一組染色體即是代表一個排程解。在每次計算適應函數值之前，需要將基因碼解碼為實際代表的排程，此動作即是以排程法則依照各作業基因所指定的排程優先順序與所佔平行機台數兩種資訊，進行作業排程。排程法則說明如下。

每個作業經由下半部基因的指定，便可得知分割批量數與指派加工的平行機台。在此將經過批量分割後的作業，稱為「批量作業」，批量作業是在同一個工作中心裡的不同平行機器上做加工。每個批量作業，於指派加工的平行機台上，判斷其空閒時區能否容納此批量加工時間。在解碼流程中，每個作業皆依照其基因所指定的排程優先順序進行下述的排程法則步驟，在排程步驟中則會使用到下半部基因所指定的分割批量數與指派加工的平行機台兩項資訊。

排程法則與步驟如下：

- 步驟一：判斷欲排之批量作業是否為訂單加工途程中的第一個作業，若是則至步驟二，若否則至步驟六。
- 步驟二：將該批量作業排入其基因所指定加工的平行機台。判斷該平行機台目前有無安排其他作業，若已有安排其他作業則至步驟三。若無，則直接將此批量作業排入，計算該批量作業的作業開始時間與作業結束時間，此批量作業完成排程，至步驟十二。
- 步驟三：尋找平行加工機器中的最早空閒時間區段。
- 步驟四：依據平行加工機器之空閒狀態與此批量作業的機器整備時間，計算此批量作業的開始時間與結束時間。
- 步驟五：判斷批量作業之結束時間與平行加工機器之下一作業的機器整備開始時間，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，執行步驟四。若無衝突產生，則此批量作業完成排程，至步驟十二。
- 步驟六：欲排之作業非訂單加工途程中的第一個作業時，首先找出途程中前一作業各分割批量的作業結束時間。
- 步驟七：依照途程中前一作業各分割批量的作業結束時間，計算欲排作業的每批分割批量最早可開始的整備時間。
- 步驟八：若執行此批量作業之平行加工機器目前有安排其他作業，則執行步驟九，否則依照步驟七求出的該批最早可開始的機器整備時間，計算此批量作業的作業開始時間與作業結束時間，完成此批量作業之排程，至步驟十二。
- 步驟九：以步驟七求出的該批最早可開始的機器整備時間為起點，尋找平行加工機器中的最早空閒時間區段。
- 步驟十：依據平行加工機器之空閒狀態與此批最早可開始的機器整備時間，計算此批量作業的開始時間與結束時間。
- 步驟十一：判斷批量作業的結束時間與平行加工機器中下一作業的機器整備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，執行步驟九。若無衝突產生，則執行步驟十二。
- 步驟十二：判斷該作業所有欲排的分割批量是否皆完成批量作業的排程。若否則至步驟一，直至該欲排作業的所有分割批量作業皆排完。

上述步驟七中，需要計算欲排作業的每批分割批量最早可開始的整備時間。若欲排作業非訂單加工途程中的第一個作業時，其最早可開始整備的時間將受到上一個途程作業結束時間的限制，基本上批量作業若完成加工，則馬上運送至下一途程站進行加工。若途程間的作業批量皆為 1(無分割)，則該作業於下一途程站最早可開始整備的時間則是此站作業結束的時間(運送時間不計)。但若作業於途程站間的分割批量數不同，就需考慮目前已完成的批量數目是否足夠供應下一途程站的上機批量數目，若不足夠，就需等待下一批或更多批完成後，在一起送至下一途程站機台。

舉例來說，某訂單須生產的數量為 120 單位，其途程 1 的分割批量為 3 批，途程 2 的分割批量為 2 批，因此途程 1 作業於每個機台上的加工數量為 40，途程 2 作業於每個機台上的加工數量則為 60。當途程 1 的第一批完成批量欲運送至途程 2 加工時，由於無法滿足供應途程 2 批量作業的上機數 60( $40 < 60$ )，因此需要再等到途程 1 的第二批完成時，總數目 80，才可完全供應途程 2 第一批欲生產的批量作業( $80 > 60$ )。

途程站與站之間的分割批量數關係，決定作業於下一途程站的整備開始時間，多種情況舉例如下圖 3.12 至圖 3.19，其中批量作業以長方條表示之，於長方條右下角的數字代表該批量作業於途程 1 的作業結束時間。

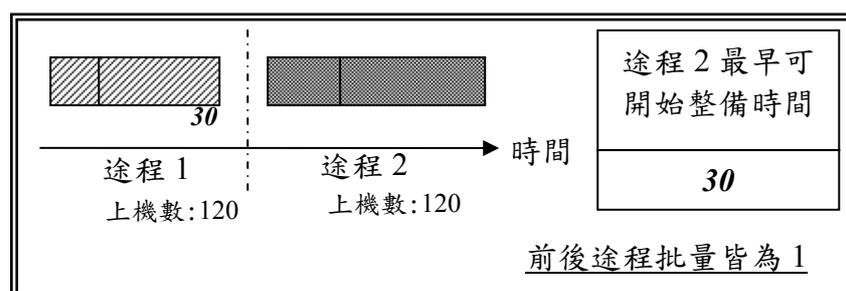


圖 3.12 途程間的批量分割關係(情況 1)

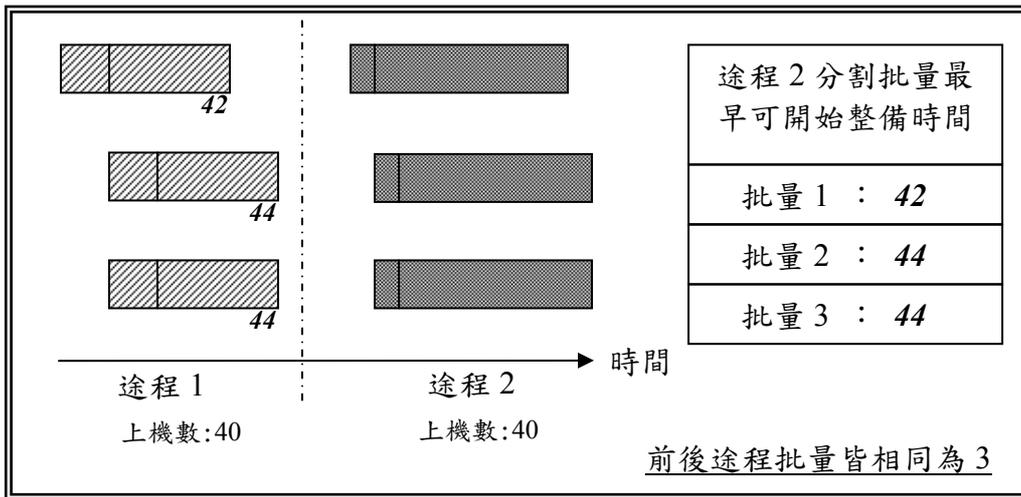


圖 3.13 途程間的批量分割關係(情況 2)

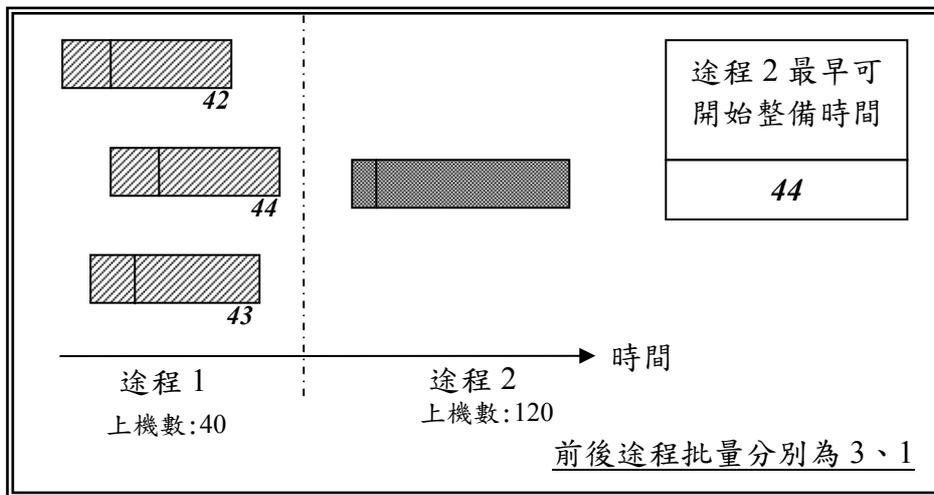


圖 3.14 途程間的批量分割關係(情況 3)

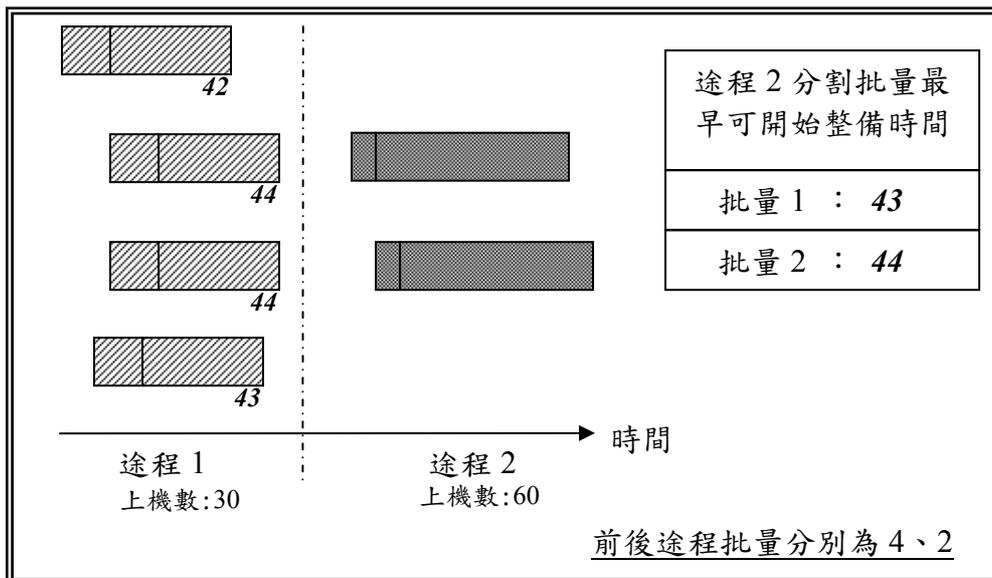


圖 3.15 途程間的批量分割關係(情況 4)

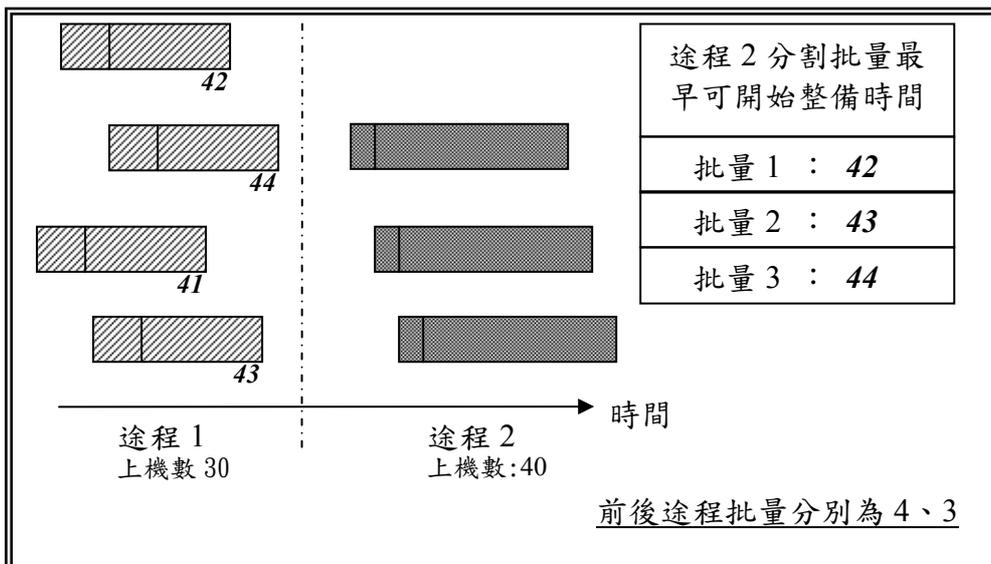


圖 3.16 途程間的批量分割關係(情況 5)

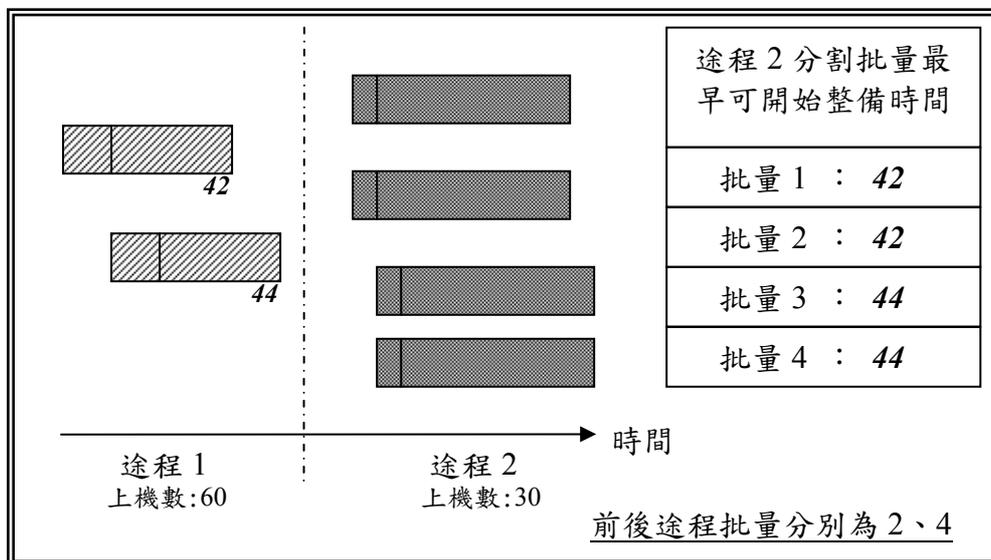


圖 3.17 途程間的批量分割關係(情況 6)

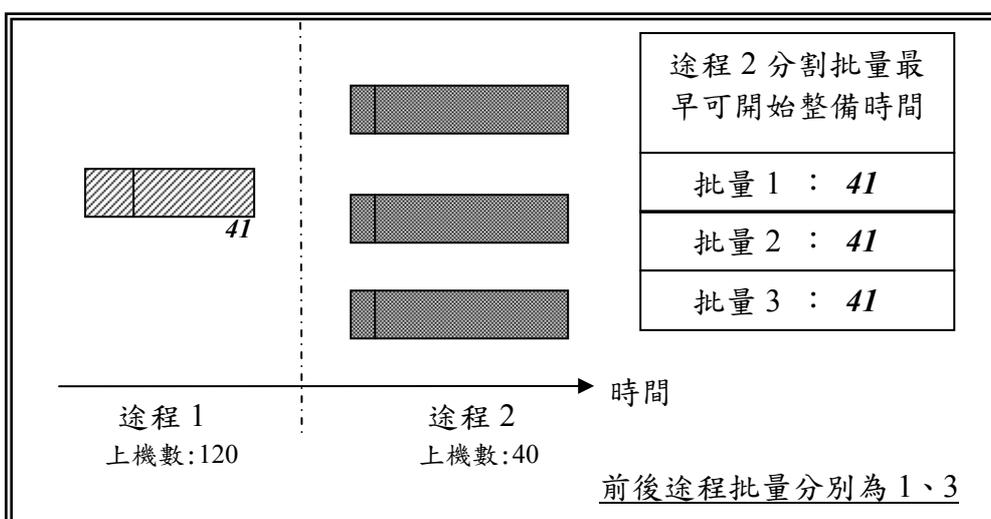


圖 3.18 途程間的批量分割關係(情況 7)

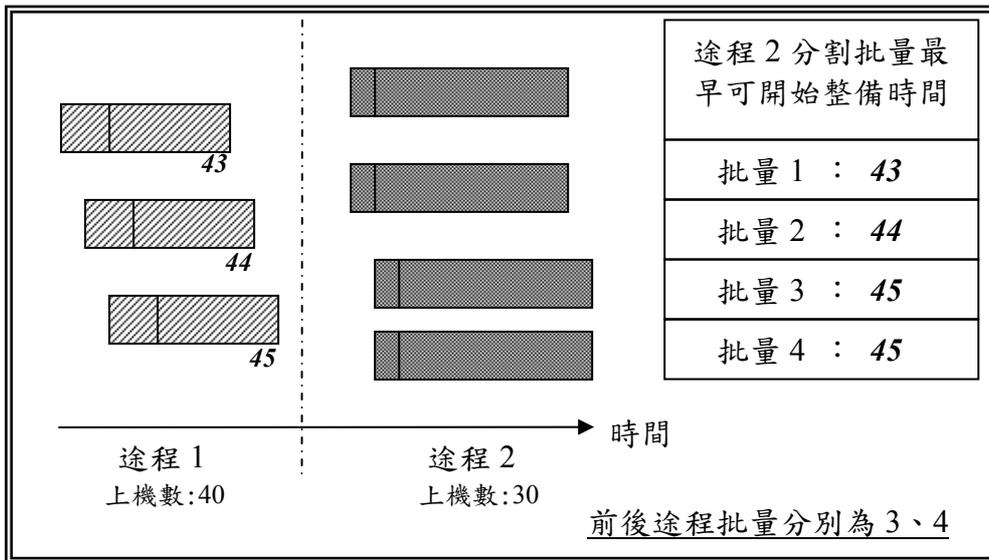


圖 3.19 途程間的批量分割關係(情況 8)

為了方便判斷各分割批量作業的開始整備時間，本研究設計以下判斷方法，說明如下。

### 批量作業開始時間判斷

定義  $B_1$  為此欲排作業於前一途程的分割批量數。

定義  $B_2$  為此欲排作業於目前途程的分割批量數。

定義  $P_1$ 、 $P_2$  與  $i$  初始值為 1。

分割批量作業的開始整備時間判斷方法如下步驟：

步驟 1：判斷  $P_1 / B_1$  是否大於或等於  $P_2 / B_2$ 。若否則至步驟 2。若是，則本作業第  $i$  個批量的最早可開始時間為前途程第  $P_1$  個批量作業的作業結束時間。將  $P_2$  值加 1。將  $i$  值加 1。至步驟 3。

步驟 2：將  $P_1$  值加 1。重新至步驟 1。

步驟 3：若該作業所有分割批量的開始整備時間皆判斷完畢則停止，否則至步驟 1。

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

本節主要說明系統如何結合禁忌搜尋法與遺傳演算法，提高系統搜尋品質，其主要是將傳統 GA 求出的一個解，再由禁忌搜尋法來進行鄰域搜尋，以找出精鍊解。

### 禁忌搜尋法

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（HGA）來求解排程解，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 3.9，其適應性函數  $G(I)=10$ 。

表 3.9 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 3.10  $S_1$  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 3.11  $S_2$  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 3.12 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  $\{(6, 12), (3, 6), (12, 3)\}$ 。

### 3. 當 $z=4$

表 3.13 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  $\{(3, 6), (12, 3), (14, 20)\}$ ，而因為此鄰域解的目標函數  $11 > 10$  (原個體的適應性函數)，所以原個體將會被替換。持續進行至  $z=7$ 。

本節中混合式遺傳演算法(HGA)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。結合禁忌搜尋後的混合式遺傳演算法(HGA)運作流程如圖 3.20。

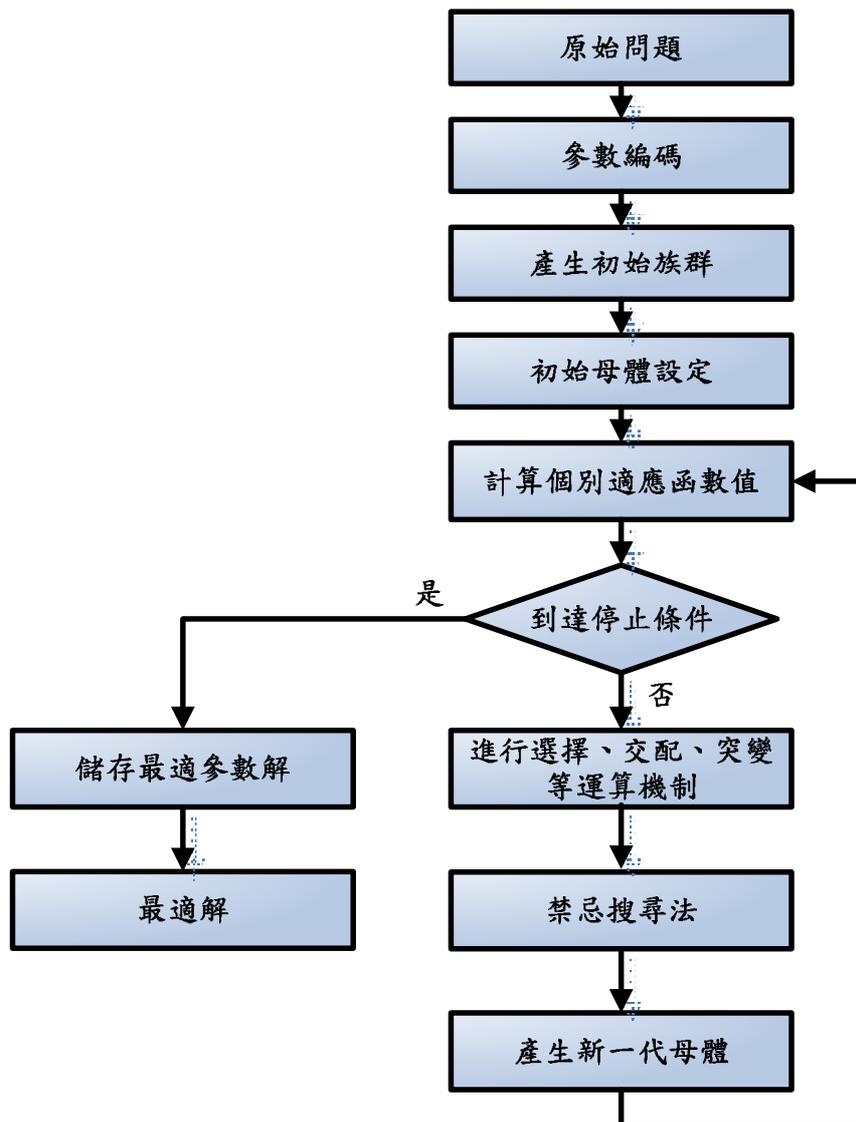


圖 3.20 混合式遺傳演算法(HGA)運作流程圖

### 3.3 重排程方法

在實際的生產環境中，排程規劃人員通常會週期性的安排生產排程，排程週期的長短則視該工廠情況而自行訂定，可能為一周或兩周。然而所規劃的初始排程，可能因為現實狀況的突變或是種種無法預期的因素干擾，造成原先初始排程必須要做修正改變，現場作業才能持續生產。我們稱上述這些突發的狀況或干擾因素為排程的「中斷事件」，更新初始生產排程以便對於變動或干擾因素作出回應的動作稱為「重排程」，而原先的初始排程於本研究則稱之為「前排程」。

中斷事件的發生可能原因眾多，普遍可舉例如：緊急插單、機台故障、物料短缺、抽單、變更交期、作業進度落後或超前、訂貨數量變更等等。其中緊急插單、機台故障與物料短缺三項對於生產系統的負面影響極大，於重排程的議題上極具代表性。因此，本文以緊急插單、機台故障與物料短缺三項作為發展本研究重排程方法時考慮的中斷事件。以下簡述此三項中斷事件的造成原因與其對生產現場的影響。

#### 機台故障

雖然多數工廠中都有例行保養與預防保養，但是無可避免的機器還是會發生偶發性故障當機，一旦機器發生故障，其對現場的影響是立即的。機器發生故障時，需用到該機器的生產作業都需要延後到機器修復為止，才能再進行。生產排程人員需要預估機器恢復正常運轉所需的時間，再依此資訊作重排程的動作。

#### 緊急插單

所謂緊急插單指的是該張訂單原先並未安排在本次的排程週期內生產，而因為某些原因必須要優先處理該張訂單，盡快安排生產其所指定之產品的相關作業。這些原因可能是訂單來自於交易頻繁的顧客，基於顧客關係給予優先處理、或是承接該張訂單將有額外的收益、訂單的數量或金額達到公司所認定的標準，給予優先生產的權利。緊急插單發生時，排程規劃人員必須重新評定該單的質性因素，決定此插單的重要程度，在於以重排。

## 物料短缺

在一般的工廠中，缺料也是常見造成現場中斷的因素，雖然缺料與當機對於現場的影響都是立即性，但是以缺料而言，其直接影響的是該缺料訂單的作業以及其後續途程的作業，而其他作業仍可以進行不會受到直接影響。缺料發生時的調整方式是先取消缺料的訂單，其他不直接受到影響的作業則往前時段遞補，以節省生產時間，等到該缺料的物料或零件到達時再將其重新插入排程。

在重排程的方法上，大致可分為部份重排法與完全重排法，部分重排法的精神在於，盡量只對被影響到的作業來進行重排修正，好處是可降低中斷干擾所造成的影響，並且可較快速的產生一個重排程解。完全重排法指的是完全重新產生一個新的生產排程，其中包括未受到中斷事件影響的作業，以此方法進行重排程可獲得較佳品質的排程結果，但是相對耗費的運算資源與時間也較多。

在本研究中，對於機台故障與物料短缺兩種中斷事件，以部份重排法進行重排。部份重排法規劃時，以原本前排程排序時的作業排程優先順序為基礎，由於此順序是由遺傳演算法求得的最適解，因此保留此排程優先順序，並直接利用在重排程上。此法的精神在於維持原前排程排序時的順序，以維持一定品質的排程解，並且降低重排造成的干擾，使大部分的作業仍然能維持於原本的排程相對位置，且由於此法不需經過遺傳演算法重新演化求解，因此重排過程快速許多。

緊急插單的中斷事件由於涉及到較多的質性因素改變，且訂單作業數也增加改變了，因此重排方法上，先重新評定加入緊急插單後的質性因素，再以完全重排法進行重排，以求得較好品質的排程解。

各中斷事件的重排程詳細方法如下。

### 3.3.1 機器故障的重排程

本研究中，機台故障的重排程以部份修正重排法來進行重排。在每一次的正常周期排程規劃中，以 GA 或 HGA 演化求解出的解，包含了上半部基因與下半部基因，這些基因值經過解碼程序後即是一規劃排程。本機台故障的部份修正重排法，主要精神在於快速產生一個高生產績效(排程目標函數值)的重排程，方法上主要是保留原前排程的基因值，依排程中斷情況將此基因值作適當修正，再將需要重排的作業依此基因重新解碼即可得重排程解。由於一般情況下，前排程基因都是以演算法求解出的最適解，因此我們保留住原前排程解的基因特性來重新規劃排程，可取得與原前排程目標函數值相近的解，幫助維持重排結果的品質。

機器故障部分重排法的各步驟如下。

#### 機器故障重排步驟

1. 紀錄故障機台與故障發生時間點(breakdown time)。

在此故障機台中，於故障時間點上正在處理的作業為「中斷作業」。

2. 由現場人員或專業維修人員預估機台維修時間，即機台當機狀態持續時間(breakdown duration)。
3. 將所有「開始整備時間早於 breakdown time」的批量作業從此重排程問題中移除。這些批量作業已生產完成或是已開始上機生產，不受此機台故障的中斷事件影響。其他剩餘的批量作業就是需要重排的作業。
4. 計算此故障機台，能再重新開始運作的時間點(Breakdown end time)，即機台修復時間點。

機台修復時間點 = 故障發生時間點 + 機台維修時間

5. 計算該中斷作業於機台修復後的作業重新「開始時間」與作業「結束時間」。

中斷作業的重新開始時間 = 機台修復時間點

中斷作業的結束時間 = 機台修復時間點 + (中斷作業於前排程的結束時間-故障發生時間點)

6. 取得需重排作業的作業排程優先順序。

從原前排程解中擷取需重排作業的上半部基因值，將上半部基因值依大小相對順序重新修正排序。舉例如下：

- (1) 假設表 3.14 為前排程中各作業的上半部基因，其中需要重排的作業代號是 1、3、5、6、7、8、9。

表 3.14 前排程作業上半部基因

作業代號	1	2	3	4	5	6	7	8	9	10
上半部基因	6	5	10	4	7	8	1	9	2	3

- (2) 擷取需重排作業的上半部基因值如下：

表 3.15 需重排作業上半部基因

作業代號	1	3	5	6	7	8	9
上半部基因	6	10	7	8	1	9	2

- (3) 依照各基因值大小的相對順序重新排序如表 3.16。表 3.16 即為進行重排時各作業的作業排程優先順序。

表 3.16 需重排作業修正後的上半部基因

作業代號	1	3	5	6	7	8	9
上半部基因	3	7	4	5	1	6	2

7. 取得需重排作業的分割批量數與機台指派資訊。

從原前排程解中擷取需重排作業的下半部基因值，擷取方式同上半部基因，但不改變下半部基因值，即需重排作業的下半部基因值與前排程解相同。

8. 將上述步驟所得的作業排程優先順序、作業分割批量數與機台指派資訊，代入演算法的解碼流程(排程法則與步驟)，即可得重排程解。

機台故障重排程的實地案例運作與各步驟結果將會於第四章做詳細說明。

### 3.3.2 物料短缺的重排程

本研究中，物料短缺的重排程方法大致上與上述機台故障的重排方法類似，方法精神在於保留前排程優良的基因特性，作法上同樣保留前排程的上半部基因與下半部基因，但由於該缺料訂單的作業時程已延遲，因此考慮增加或改變此缺料作業的分割批量與加工機台數，以此趕工生產。

在物料短缺的重排規劃中，將分成兩個階段執行：第一階段是該缺料尚未到達前，先取消缺料的訂單，其他不直接受到影響的作業則往前時段遞補，以節省生產時間。第二階段是該缺料補齊到達工廠後，將此缺料定單尚未完成的作業重新插入排程。

物料短缺修正重排法的兩階段重排分別如下。

#### 階段一物料短缺重排步驟

1. 紀錄發現缺料現象的時間點、缺料作業代號。假設發現缺料現象的第一時間即馬上進行重排程。
2. 由現場人員或專業採購人員預估此缺料補齊到達工廠的時間。
3. 將前排程中「作業開始整備時間」小於「發現缺料的時間點」的所有批量作業從重排程問題中移除，這些批量作業已生產完成或是已開始上機生產，不受此缺料事件的影響。
4. 將該缺料訂單的作業也從此階段一重排程中移除。
5. 取得階段一需重排作業的作業排程優先順序。依照原前排程的作業排程優先順序(即演算法中的上半部基因)，在不改變相對順序的原則下進行修正，過程同上節機台故障修正重排的步驟6。
6. 取得階段一需重排作業的分割批量數與機台指派資訊(即演算法中的下半部基因)。過程同上節機台故障修正重排的步驟7。
7. 將上述步驟所得的作業排程優先順序、作業分割批量數與機台指派資訊，代入演算法的解碼流程(排程法則與步驟)，即可得階段一的缺料重排程解。

在該缺料尚未到達工廠之前，生產現場以此階段一重排程結果為排程依據，持續進行生產。

在該缺料作業的物料到達後，立即將該缺料訂單尚未完成的作業重新插入排程中。階段二重排方式大致與階段一相同，主要也是將修正後的作業排程優先順序代入解碼程序求解，但由於缺料訂單的作業時程已延遲，為了加快其生產速度，傳統作法上排程人員可能將該缺料訂單作業的分割批量數設為最大，即給予最多的機台資源來趕工生產。雖然此作法對於該缺料訂單的交期滿足是有幫助的，但是相對於其他訂單的交期考量，給予缺料單作業最多的機台資源可能會造成其他訂單作業的生產時程必須往後遞延，間接降低其他訂單的達交率，進而影響到整體目標函數值。為了決定較客觀的訂單作業分割批量數，使重排時能取得較佳的整體目標函數值，而非只單純考量缺料訂單的交期，我們對於需重排作業的分割批量數與機台指派值(下半部基因)以 GA 搜尋並重排，GA 求解時各作業的排程優先順序(上半部基因)則維持不動。物料短缺階段二重排程的步驟如下。

### 階段二物料短缺重排步驟

1. 將不需重排的作業移除。確定該缺料到達工廠的時間點後，將目前排程中「作業開始整備時間」早於「缺料到達工廠時間點」的所有批量作業從階段二重排問題中移除，這些作業已生產完成或是已開始上機生產。
2. 取得階段二重排程時的作業排程優先順序。於缺料訂單的作業時程已延遲，因此，我們將其作業排程的優先順序設為最高；其他作業的順序則在此缺料訂單作業之後，但仍依照前排程的作業排程優先順序，以不改變相對順序的原則下進行修正。修正過程同機台故障重排的步驟 6。
3. 將上步驟所得的作業排程優先順序，代入 GA 演算法中求解重排程。以 GA 搜尋演化下半部基因，以找出對目標函數值幫助最大的作業分割批量數與機台指派值，並依此重排。求解時我們設定下列限制：
  - (1) 上半部基因固定不變(為上步驟所求得的基因值)。
  - (2) 限制缺料作業的分割批量數不可小於原前排程的分割數。其他作業則無限制。

設定限制(2)的原因是因為，若再降低缺料作業的分割批數，只會造成缺料作業的交期更往後延。

完成上述步驟 3 後，即可求得物料短缺的階段二重排程結果。詳細的實例運作與各步驟結果將會於第四章做說明。

### 3.3.3 緊急插單的重排程

本研究中，緊急插單的重排程是以完全重排法的方式進行重排。雖然緊急插單的中斷現象亦可以使用上小節的部份修正重排法，但是由於排程多了一筆緊急插單的考量，定性因素必須重新評估，且總訂單作業數也改變了，因此以完全重排法進行重排，以求得品質較高的重排程解。此外，不同於機台故障或缺料現象對於現場的立即性影響，緊急插單的重排規劃較適合使用求解運算時間較長的完全重排法。

一般傳統的插單重排，都是將該筆插單的重要程度視為最高，排程的優先順序為第一，此作法對於該筆插單的排程結果幫助最大，但是以公司的整體利益而言可能不是最好的作法。將插單強制排入第一優先生產，可能造成其他訂單的交期延遲，間接造成利潤的損失。因此，插單重排應該以整體公司利益來考量，重新評估該單的定性因素來決定其排程優先程度，再將該插單與其他的訂單一起以演算法求解，以取得整體目標函數值較高的排程解。

本研究緊急插單的重排程步驟如下。

#### 緊急插單重排步驟

1. 取得緊急訂單的資料，包括得知緊急訂單的時間、訂單交期、作業途程站、數量、各作業整備時間、各作業加工時間、上機批量限制等等。
2. 將原排程中「作業開始整備時間」早於「得知緊急訂單時間」的所有批量作業從重排程問題中移除，這些批量作業已生產完成或是已開始上機生產，不受此緊急插單的中斷事件影響。
3. 以 AHP 法，重新修正加入該插單後的定性因素評估結果。取得定性因素考量下的訂單排程優先順序與各訂單的權重值。
4. 將需要重排的作業，配合步驟 3 的定性評估結果，重新以 GA 或 HGA 演化求解，即可得此緊急插單的重排程。

緊急插單重排程的詳細實例運作與各步驟結果將會於第四章做說明。

### 3.4 穩健排程(Robust Schedule)

一般而言，排程解的目標函數值若不會因為中斷事件的干擾而下降太多，則稱此排程為穩健排程。現實生產環境中，常因為機台當機、缺料、插單、抽單或產品品質等等問題，造成生產作業中斷，迫使排程人員必須重新規劃排程。雖然排程在中斷後，排程人員會依適當方法進行重排，但由於中斷事件的負面影響，生產績效往往會大幅下降，不如原先排程所規劃。因此，在正常的週期排程規劃中，若可事先規劃出一個具有穩健性(Robustness)的排程，則生產作業若遭遇中斷事件，整體生產仍可維持高的績效指標。

不同於一般穩健排程所考慮的目標函數下降問題，重排程時通常需考量到重排的結果是否具穩定性(Stability)。由於重排程的計畫是承接原本前排程的規劃來完成的，因此若重排程的結果與原前排程差異太大，許多生產後勤的準備作業可能會無法及時完成。例如備料計畫、生產工具的準備長前置時間的關鍵原料或組件，可能皆需要非常長時間來進行事前準備。若排程重新規劃時做了大幅度更動，可能會造成上述許多後勤作業無法及時配合，則此時重新規劃的排程也無法使用。

由上述可知，重排程結果的穩定性好或壞，是決定於重排程的方法，因此，穩定性(Stability)的考量通常是在於重排程階段，一般情況下的週期排程階段則不考慮穩定性。相反的，穩健性(Robustness)的考量則是在一般週期排程階段，以求出的穩健排程來降低中斷事件造成的目標函數下降問題。由於穩健性與穩定性是在不同階段考量的因素，因此多數關於穩健性或穩定性的排程研究皆是單一考量其中一個。本研究則是將穩健性與穩定性同時考量，於排程規劃求解時，事先預估該排程經過中斷事件並重排後，所得的目標函數值情況與穩定性表現。本文中，結合穩健性與穩定性之排程，統一以「穩健排程」的名稱代表之。

在說明本研究的穩健排程求解方法之前，先於下述中重新闡述穩健性指標與穩定性指標兩者於本研究中的定義與計算方式。

#### 3.4.1 穩健性指標(Robustness)

穩健性於排程上的涵義是指，當生產活動發生若干干擾或變異時，排程仍然能維持高的生產績效，即高的目標函數值。相反的，一般排程通常是尋求最佳解，而此最佳解雖然有較高的目標函數值，但是若生產活動發

生變異或干擾，目標函數值通常會大幅度下降。以下圖 3.21 非連續型的函數為例， $x$  等於 13 時即是上述所稱的穩健解，雖然此解之目標函數值不比  $x$  等於 5 所求得的最佳解來得高，但是此解的穩健性則較高，例如：當  $x$  等於 13 時， $x$  向左或向右偏移一單位，其所得之目標函數值仍是 3；相反的， $x$  等於 5 時所求得的目标函數值為 5，但是若  $x$  向左或向右偏移一單位，目標函數值則掉落到 0。

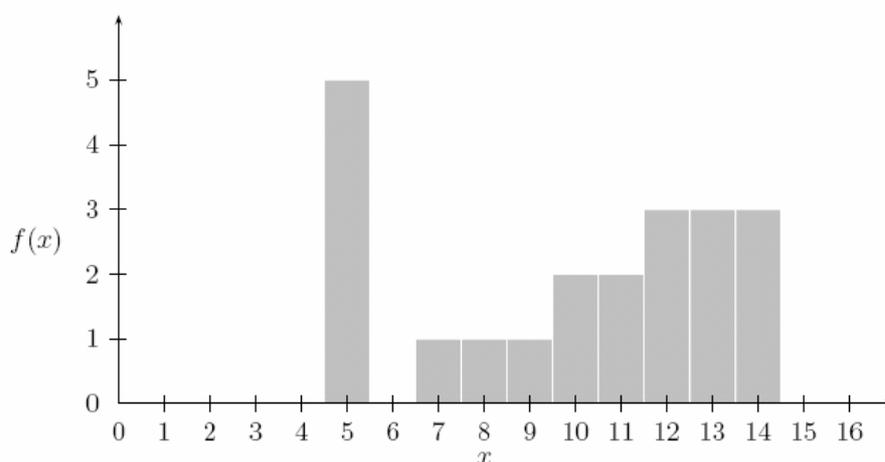


圖 3.21 穩健解示意圖

由上述說明可知，穩健性衡量的是一個解的目標函數值經歷干擾後的情況為何，因此，本研究中定義一個排程的穩健性指標為：該排程經過中斷事件並且重排後所得到的目標函數值。

假設一排程  $S$  的原目標函數值為  $f_v$ ，經過中斷事件干擾後，將該排程進行重排規劃，重排後得到排程  $S'$  之目標函數值為  $f_v'$ ，則稱  $f_v'$  為前排程  $S$  的穩健性指標。以圖 3.21 為例，解  $x=13$  在往右一單位的干擾下，目標函數值仍為 3，因此解  $x=13$  在往右一單位干擾下的穩健性指標為 3；相反的  $x=5$  穩健性指標為 0。

### 3.4.2 穩定性指標(Stability)

根據 Rangaritratsameet al. (2004)，穩定性於排程上的涵義是指，重排規劃後的排程與原本前排程之間，各作業起訖時間上的差異程度。衡量排程穩定性的主要目的，在於避免排程經過重新規劃後做了太大幅度的更動，造成許多備料或訂購作業無法配合。舉例來說，若某作業經過重排後，作業開始時間提早了許多，這時可能造成該作業生產所需的物料來不及備

足，使得此作業無法依重排程結果規劃的時間進行生產；相反地，若作業經過重排後，作業開始時間延後了許多，此時可能造成原先用於該作業生產的物料或半成品，大量堆放於現場，造成不當庫存的累積、生產現場混亂等等。而除了作業開始時間的差異需要衡量之外，作業結束時間的差異也需衡量，舉例來說，若重排後某作業的開始時間與前排程差異不大，但作業結束時間卻有很大的差異，則表示該作業經過重排後，各批量作業間的生產時程太過離散，此狀況亦會產生現場不當的庫存累積，造成生產現場混亂。

由上述的說明，我們定義本研究中穩定性指標的計算方式，其中必須先計算作業時間總差異值，再將其正規化即可得穩定性指標值，計算方式如下：

$Tdiff_x$ ：個體解  $x$  的作業時間總差異。衡量前排程與重排程結果之間，各作業起訖時間的差異量。

$T_{O_{ij}}$ ：前排程中，訂單  $i$  作業  $j$  的各批量作業間，最早的作業開始整備時間

$T'_{O_{ij}}$ ：重排程結果中，訂單  $i$  作業  $j$  的各批量作業間，最早的作業開始整備時間

$E_{O_{ij}}$ ：前排程中，訂單  $i$  作業  $j$  的各批量作業間，最晚的作業結束時間

$E'_{O_{ij}}$ ：重排程結果中，訂單  $i$  作業  $j$  的各批量作業間，最晚的作業結束時間

$$Tdiff_x = \frac{\left( \sum_i \sum_j |T'_{O_{ij}} - T_{O_{ij}}| + \sum_i \sum_j |E'_{O_{ij}} - E_{O_{ij}}| \right)}{2}$$

因此，若此作業時間總差異值愈小，則代表該排程穩定性愈佳，我們以作業時間總差異的大小來衡量個體解的穩定性，如同製距指標一樣，需要先進行正規化，才能與其他值作加權的運算。

穩定性指標值的正規化如下：

$$stb_x = \frac{\min Tdiff + C}{Tdiff_x + C}$$

其中，

$\min Tdiff$ ：到目前為止，搜尋過程中最小的作業時間總差異

$stb_x$ ：個體解  $x$  的穩定性指標值

$C$ ：正數參數值

上式中， $C$  為一個正數的參數值，由於搜尋過程中  $\min T_{diff}$  可能為 0，因此為了避免無法進行除法運算，分子與分母皆加上一個正數值  $C$ 。此運算並不會改變個體解的穩定性評估，以相對的角度而言，在同一代中的個體，作業時間總差異值低者，穩定性指標值恆高；作業時間總差異值高者，穩定性指標恆低，對於演算法中的機制不會造成明顯的影響。特別需注意的是， $C$  值參數的設定，需視排程問題的單位時間尺度而定，不宜太大或太小，本研究之  $C$  值的設定為 50。

### 3.4.3 穩健排程方法

為了在一般週期排程規劃的階段，就事先排出一個具有穩健性與穩定性的排程解，我們於正常排程的方法中(如 3.2 節)，再加入測試性干擾的機制，事先以某些特定的生產中斷事件，來測試該個體(排程解)的穩健性與穩定性指標的表現，然後再依照此兩項指標與原先目標函數值的綜合評估，找出同時具有穩健性與穩定性的排程，該穩健排程則對求解中設定的中斷事件提供吸收干擾的功能，日後若此穩健排程解真的經歷該中斷事件，則重排後的結果亦可保持較高的目標函數值與穩定性表現。

實際生產環境的中斷事件非常多，排程規劃人員可依照其需求，將日常生產活動中最嚴重或最常發生的中斷事件，納入測試性干擾機制中，在排程求解過程中來測試各個體是否對於此中斷事件能有好的穩健與穩定性表現。這些中斷事件如同 3.3 節介紹的機台故障、插單與缺料等等。本研究的穩健排程求解示意如下圖 3.22。其中測試性干擾機制只是提供事先測試的功能，並不改變原個體解的結構。

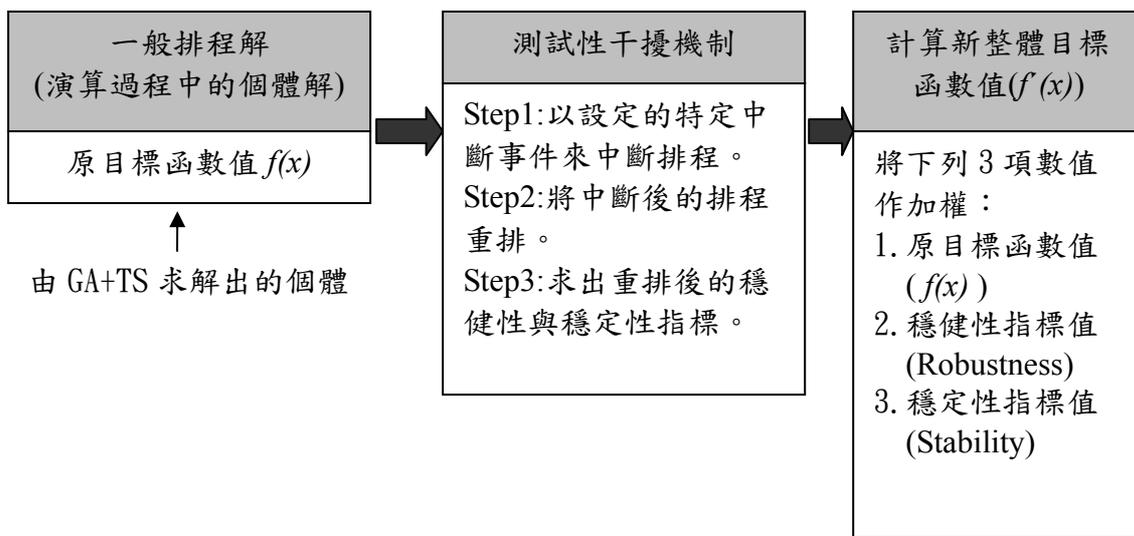


圖 3.22 個體解測試性干擾機制示意圖

在穩健排程求解過程中，個體是重新以一個新的目標函數值來衡量其優劣，此新的目標函數值包括了原先目標函數值(一般排程下)、穩健性指標值與穩定性指標值三項。將三項作加權即為此新的目標函數值，而各項指標值的權重為參數，可依求解品質而自行設定。

求解穩健排程的運算中，新的目標函數如下：

$$f(x)' = a \cdot f(x) + b \cdot stb_x + c \cdot rbn_x$$

其中，

$f(x)$ ：個體解  $x$  的原目標函數值，即一般排程下的目標函數值

$rbn_x$ ：個體解  $x$  的穩健性指標值(Robustness)

$stb_x$ ：個體解  $x$  的穩定性指標值(Stability)

整體穩健排程的求解過程，主要是將測試性干擾機制納入求解一般排程使用的 HGA(如 3.2.3 節)，整體穩健排程的混合式遺傳演算法流程如圖 3.23。其中由於原目標函數值的評估也是非常重要的，因此只擷取原目標函數值較大的前  $k$  個進行測試，可不必對於所有個體進行測試，一方面也幫助提升演算法的搜尋速度，避免不需要的測試運算。 $k$  值參數之設定，則可視求解出的品質而自行調整，本研究中每代個體數為 20， $k$  值大致設定為 2~5。

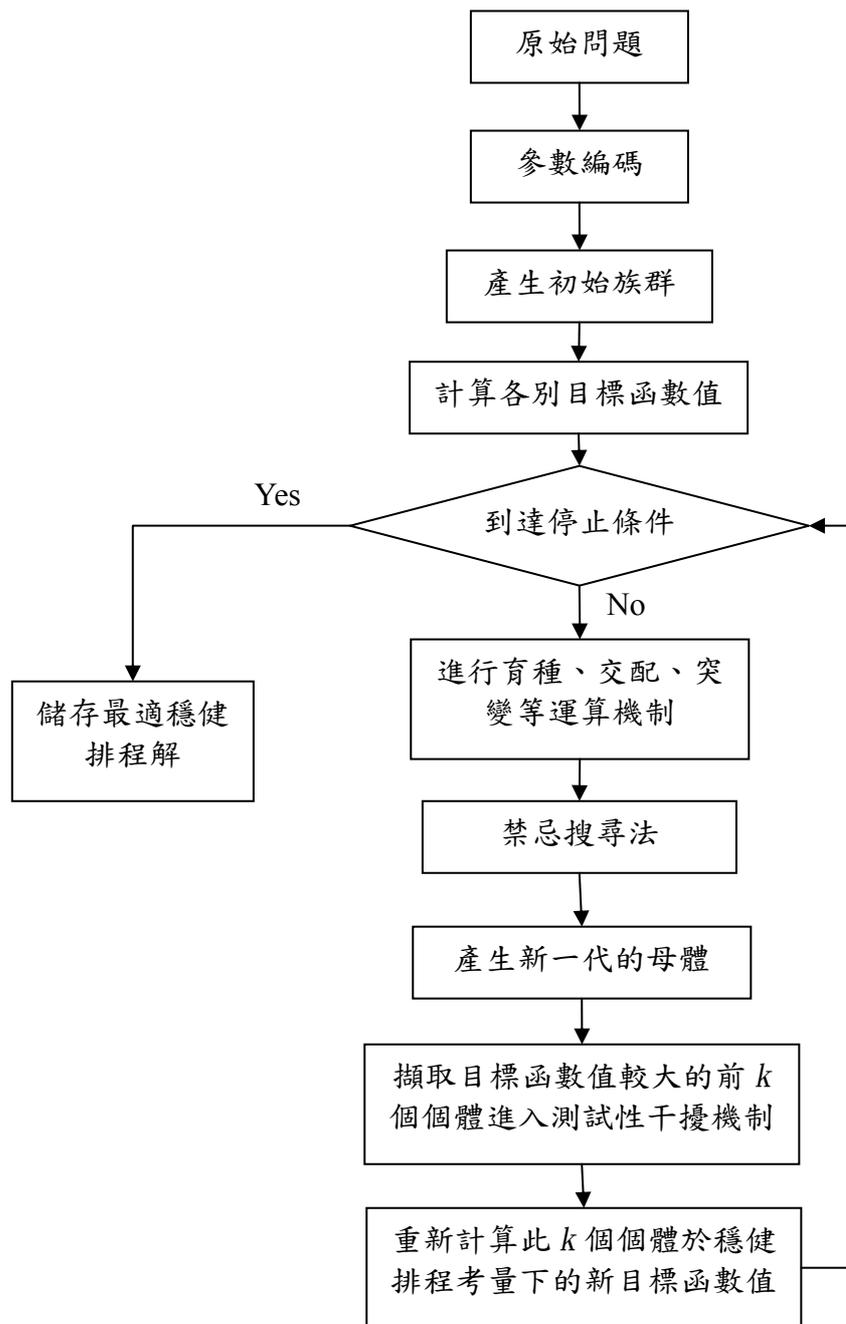


圖 3.23 穩健排程 HGA 流程圖

## 第四章 系統實證

本章節將以零工式工廠的排程問題為例，依據第三章所述的系統架構來進行驗證。首先於 4.1 節中說明實證問題之排程環境假設。4.2 節說明定性因素評估與各績效指標權重值訂定過程。4.3 節說明以 GA 與 HGA 求得一般排程(非穩健)的結果。4.4 節說明機台故障、物料短缺與緊急插單三種中斷事件的重排程步驟結果。4.5 節說明求解過程中加入測試性干擾機制，求得穩健排程之步驟結果，並驗證穩健排程的效用。

### 4.1 實證問題說明

在實證中，零工式工廠的排程環境假設為：現場有 8 個不同的工作中心，每一個工作中心有 1~3 台不等的平行機器，平行機器的類型屬於相同機器，本次排程規劃週期中有 10 張訂單，每一訂單均代表一種產品。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梯形模糊數。每張訂單的作業數並不相同。每一種類機器的平行機器數如表 4.1，訂單的資料如表 4.2。

表 4.1 各工作中心之平行機台數目

工作中心	1	2	3	4	5	6	7	8
平行機台數目	3	2	2	1	3	2	3	3

表 4.2 訂單資料

訂單	單位數	作業	加工 工作中心	機器 整備時間	單位 加工時間	上機 批量限制	最早 交期時間	最可能 交期時間	最晚 交期時間
1	120	1	3	1	0.183	55	55	65	75
		2	4	1.8	0.067	75			
		3	7	1.2	0.175	57			
		4	1	0.7	0.125	40			
		5	6	2.8	0.108	83			
		6	8	0.5	0.167	48			
2	85	1	8	0.5	0.271	22	90	100	110
		2	3	0.4	0.200	35			
		3	1	0.3	0.165	27			
		4	2	0.5	0.188	37			

表4.2 訂單資料 (續)

		5	5	1	0.200	55			
		6	7	0.3	0.224	27			
3	200	1	3	0.4	0.105	38	106	116	126
		2	2	2.2	0.090	89			
		3	4	4	0.035	186			
		4	5	0.5	0.080	38			
		5	8	2.5	0.120	100			
4	140	1	6	1	0.079	51	123	133	143
		2	5	0.3	0.121	41			
		3	4	3	0.071	98			
		4	2	3.4	0.136	118			
		5	7	1.3	0.129	62			
		6	1	0.5	0.114	44			
5	90	1	1	0.3	0.167	30	52	62	72
		2	3	0.4	0.211	39			
		3	7	0.3	0.211	27			
		4	5	0.2	0.200	20			
6	135	1	1	3.2	0.126	103	74	84	94
		2	4	1.6	0.067	75			
		3	2	0.5	0.126	44			
		4	8	1.1	0.178	56			
		5	5	0.3	0.141	28			
		6	6	3	0.089	90			
7	155	1	2	0.3	0.097	31	77	87	97
		2	7	0.5	0.129	47			
		3	1	0.7	0.142	49			
		4	8	0.4	0.135	36			
		5	5	0.2	0.148	15			
8	170	1	4	3.5	0.082	146	111	121	131
		2	6	1	0.088	57			
		3	7	0.6	0.100	50			
		4	2	1.8	0.100	80			
		5	3	1.5	0.082	73			
		6	8	0.8	0.129	54			
9	210	1	2	1.5	0.043	70	82	92	102
		2	1	0.3	0.090	22			
		3	7	1.3	0.095	63			
		4	5	2	0.033	90			
		5	4	4	0.071	196			
		6	3	3.2	0.105	92			
10	105	1	8	0.4	0.124	32	48	58	68
		2	5	1.2	0.105	67			
		3	3	1	0.143	49			

表4.2 訂單資料 (續)

	4	2	0.5	0.076	39			
	5	6	0.6	0.190	47			

## 4.2 定性因素評估

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 4.3、表 4.4、表 4.5)，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 4.6、表 4.7、表 4.8、表 4.9)。最後計算出各訂單在定性因素下的加權值 (表 4.10)，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

由表 4.3 與 4.5 可得，遺傳演算法進行時的適應性函數如下：

$$\text{fitness} = 0.75(0.28 \text{ 製距} + 0.65 \text{ 交期滿足} + 0.07 \text{ 機器使用率}) + 0.25(1 - \text{懲罰函數值})$$

表 4.3 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 4.4 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 4.5 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

表 4.6 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 4.7 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 4.8 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 4.9 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 4.10 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
權重	0.47	0.16	0.30	0.07		
Order_1	0.1554	0.0852	0.1839	0.0414	0.1447	2
Order_2	0.0792	0.0598	0.0497	0.056	0.0656	8
Order_3	0.175	0.17	0.0712	0.1349	0.1403	3
Order_4	0.1363	0.2454	0.1452	0.2403	0.1637	1
Order_5	0.0762	0.0598	0.0232	0.054	0.0562	9
Order_6	0.0656	0.0416	0.0878	0.0869	0.0699	7
Order_7	0.0635	0.17	0.0372	0.1349	0.0776	6
Order_8	0.0605	0.0328	0.2515	0.0328	0.1114	5
Order_9	0.1085	0.1094	0.1145	0.1913	0.1162	4
Order_10	0.0798	0.026	0.0358	0.0275	0.0544	10

表 4.11 定性評估之訂單順序結果

定單順序	1	2	3	4	5	6	7	8	9	10
訂單編號	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

### 4.3 一般排程結果

本小節首先以 4.1 小節與 4.2 小節的例子，配合傳統遺傳演算法(GA)與混合式遺傳演算法(HGA)來說明本系統的細部排程結果。本小節求解過程尚未加入測試性干擾，本文以「一般排程」稱之。穩健排程結果則於 4.5 節說明。

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 3.2.1 節中已提及。其他相關設定如表 4.12 及表 4.13 所示。

表 4.12 遺傳演算法(GA)中參數之設定值

參數	參數值
群體大小	20
演算代數	400
交配方式	Random key representation
上半部基因突變機率	0.005
下半部基因突變機率	0.005

表 4.13 混合式遺傳演算法(HGA)中參數之設定值

參數	參數值
群體大小	20
演算代數	400
交配方式	Random key representation
上半部基因突變機率	0.005
下半部基因突變機率	0.005
禁忌搜尋次數	4

以表 4.12 與表 4.13 參數求解後，過程如下圖 4.1。分析兩者求解過程，在 150 代時則開始穩定收斂，只有 HGA 在 350 代後才有微幅上升。整體結果可看出，在相同求解代數條件下，可知混合式遺傳演算法(HGA) 應用於零工式平行機台排程時，所得到的結果差異明顯要比傳統遺傳演算法(GA)較佳。

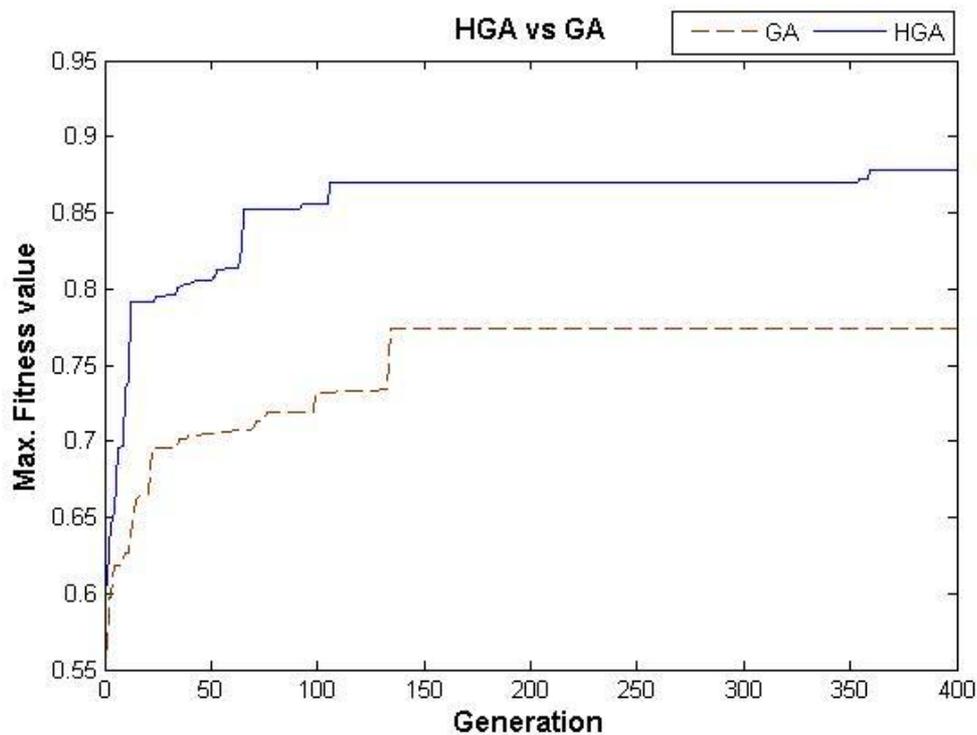


圖 4.1 整體最適適應函數值趨勢圖

表 4.14 整體適應函數值與搜尋時間

演算法		GA	HGA
量性 目標函數值	正規化後製距	0.9158	0.9513
	交期滿足度	0.6830	0.8855
	機器使用率	0.4756	0.4797
懲罰函數值		0.1048	0.1145
整體適應函數值		0.7741	0.8780
搜尋時間(秒)		521.47	4472.80

表 4.15 製距值比較

演算法	GA	HGA
搜尋過程中最短之製距	114.2	117.3
最適解製距	124.7	123.3
最適解正規化後之製距	0.9158	0.9513

表 4.16 訂單交期滿足(GA)

訂單編號	1	2	3	4	5	6	7	8	9	10	平均
交期	65	100	116	133	62	84	87	121	92	58	
完工時間	96.8	105. 7	111. 3	124. 7	68.1	95.7	90.7	115. 3	116. 7	106. 3	
交期滿足度	0	1	1	1	1	0.83	1	1	0	0	<b>0.683</b>

表 4.17 訂單交期滿足(HGA)

訂單編號	1	2	3	4	5	6	7	8	9	10	平均
交期	65	100	116	133	62	84	87	121	92	58	
完工時間	71	107.1	104.9	123.3	68.5	94.3	102.1	113	96.1	73.5	
交期滿足度	1	1	0.945	1	1	0.97	0.49	1	1	0.45	<b>0.8855</b>

表 4.18 機器使用率(GA)

工作中心	工作中心 1			工作中心 2		工作中心 3		工作中心 4	工作中心 5	
機台編號	機器 1-1	機器 1-2	機器 1-3	機器 2-1	機器 2-2	機器 3-1	機器 3-2	機器 4-1	機器 5-1	機器 5-2
使用率	0.2566	0.3007	0.8291	0.6369	0.6185	0.6084	0.5056	0.6146	0.4356	0.4055
工作中心	工作中心 6			工作中心 7			工作中心 8			
機台編號	機器 5-3	機器 6-1	機器 6-2	機器 7-1	機器 7-2	機器 7-3	機器 8-1	機器 8-2	機器 8-3	平均使用率
使用率	0.5366	0.3293	0.4133	0.4977	0.4116	0.2680	0.4082	0.4507	0.5088	0.4756

表 4.19 機器使用率(HGA)

工作中心	工作中心 1			工作中心 2		工作中心 3		工作中心 4	工作中心 5	
機台編號	機器 1-1	機器 1-2	機器 1-3	機器 2-1	機器 2-2	機器 3-1	機器 3-2	機器 4-1	機器 5-1	機器 5-2
使用率	0.2878	0.2595	0.4095	0.6427	0.6017	0.7388	0.5882	0.7692	0.5773	0.3346
工作中心	工作中心 6			工作中心 7			工作中心 8			
機台編號	機器 5-3	機器 6-1	機器 6-2	機器 7-1	機器 7-2	機器 7-3	機器 8-1	機器 8-2	機器 8-3	平均使用率
使用率	0.3689	0.3058	0.5567	0.4128	0.3839	0.5574	0.4674	0.4872	0.3657	0.4797

表 4.20 訂單的平均基因順序(GA)

訂單編號	1	2	3	4	5	6	7	8	9	10
訂單基因順序總和	138	183	138	162	126	100	152	174	172	195
訂單基因平均順序	23	30.5	27.6	27	31.5	16.67	30.4	29	28.67	39
排序	2	8	4	3	9	1	7	6	5	10

表 4.21 訂單的平均基因順序(HGA)

訂單編號	1	2	3	4	5	6	7	8	9	10
訂單基因順序總和	91	197	145	172	133	120	162	188	119	213
訂單基因平均順序	15.17	32.83	29	28.67	33.25	20	32.4	31.33	19.83	42.6
排序	1	8	5	4	9	3	7	6	2	10

本次搜尋結果所得之訂單順序與 4.2 節中定性因素下訂單順序差異如表 4.22。

表 4.22 搜尋結果所得之訂單順序與定性因素下訂單順序差異

訂單編號	1	2	3	4	5	6	7	8	9	10
訂單加權值(AHP)	0.1447	0.0656	0.1403	0.1637	0.0562	0.0699	0.0776	0.1114	0.1162	0.0544
定性因素訂單順序	2	8	3	1	9	7	6	5	4	10
最大違反訂單順序	9	3	8	10	2	4	5	6	7	1
最大違反差異平方	49	25	25	81	49	9	1	1	9	81
加權後之最大違反差異平方	7.0903	1.64	3.5075	13.2597	2.7538	0.6291	0.0776	0.1114	1.0458	4.4064
搜尋之訂單順序(GA)	2	8	4	3	9	1	7	6	5	10
排序差異平方(GA)	0	0	1	4	0	36	1	1	1	0
加權後差異平方(GA)	0	0	0.1403	0.6548	0	2.5164	0.0776	0.1114	0.1162	0
搜尋之訂單順序(HGA)	1	8	5	4	9	3	7	6	2	10
排序差異平方(HGA)	1	0	4	9	0	16	1	1	4	0
加權後差異平方(HGA)	0.1447	0	0.5612	1.4733	0	1.1184	0.0776	0.1114	0.4648	0

總差異量正規化(GA)：

$$(0.1403+0.6548+2.5164+0.0776+0.1114+0.1162)/(7.0903+1.64+3.5075+13.2597+2.7538+0.6291+0.0776+0.1114+1.0458+4.4064)= 0.1048$$

總差異量正規化(HGA)：

$$(0.1447+0.5612+1.4733+1.1184+0.0776+0.1114+0.4648)/(7.0903+1.64+3.5075+13.2597+2.7538+0.6291+0.0776+0.1114+1.0458+4.4064)=0.1145$$

結合上述各指標值的結果，計算整體目標函數值如下：

$$\text{適應函數(GA)}=0.75 \cdot (0.28 \cdot 0.9158 + 0.65 \cdot 0.6830 + 0.07 \cdot 0.4756) + 0.25 \cdot (1-0.1048)=0.744$$

$$\text{適應函數(HGA)}=0.75 \cdot (0.28 \cdot 0.9513 + 0.65 \cdot 0.8855 + 0.07 \cdot 0.4797) + 0.25 \cdot (1-0.1145)=0.878$$

表 4.23 訂單排程結果(GA)

訂單	作業	加工 工作中心	加工 機器數	加工 批量	機器 整備時間	作業 開始時間	作業 結束時間
1	1	3	1	120	0	1	23
	2	4	1	120	30.8	32.6	40.6
	3	7	2	60	40.6	41.8	52.3
				60	40.6	41.8	52.3
	4	1	2	60	52.3	53	60.5
				60	52.3	53	60.5
5	6	1	120	60.5	63.3	76.3	
6	8	1	120	76.3	76.8	96.8	
2	1	8	3	28	0	0.5	8.1667
				28	0	0.5	8.1667
				28	0	0.5	8.1667
	2	3	2	43	10.9	11.3	19.8
				43	33.9	34.3	42.8
	3	1	2	43	19.8	20.1	27.1
				43	42.8	43.1	50.1
	4	2	2	43	32.5	33	41
				43	59.5	60	68
	5	5	1	85	81.1	82.1	99.1
	6	7	3	28	99.1	99.4	105.73
				28	99.1	99.4	105.73
28				99.1	99.4	105.73	
3	1	3	2	100	0	0.4	10.9
				100	23	23.4	33.9
	2	2	2	100	21.3	23.5	32.5
				100	48.3	50.5	59.5
	3	4	1	200	59.5	63.5	70.5
	4	5	1	200	80.7	81.2	89.2
5	8	2	100	79	81.5	93.5	
				96.8	99.3	111.3	
4	1	6	1	140	33.5	34.5	45.5
	2	5	3	47	45.5	45.8	51.467
				47	45.5	45.8	51.467
				47	45.5	45.8	51.467
	3	4	1	140	70.5	73.5	83.5
	4	2	1	140	83.5	86.9	105.9
	5	7	2	70	105.9	107.2	116.2
				70	105.9	107.2	116.2
6	1	2	70	116.2	116.7	124.7	
			70	116.2	116.7	124.7	
5	1	1	1	90	0	0.3	15.3
	2	3	1	90	19.8	20.2	39.2
	3	7	3	30	42.4	42.7	49.033
30				52.3	52.6	58.933	

表 4.23 訂單排程結果(GA)

				30	52.3	52.6	58.933
	4	5	2	45	58.933	59.133	68.133
				45	58.933	59.133	68.133
6	1	1	1	135	0	3.2	20.2
	2	4	1	135	20.2	21.8	30.8
	3	2	1	135	30.8	31.3	48.3
	4	8	2	68	48.3	49.4	61.4
				68	48.3	49.4	61.4
	5	5	1	135	61.4	61.7	80.7
6	6	1	135	80.7	83.7	95.7	
7	1	2	1	155	0	0.3	15.3
	2	7	1	155	15.3	15.8	35.8
	3	1	1	155	35.8	36.5	58.5
	4	8	2	78	58.5	58.9	69.4
				78	61.4	61.8	72.3
	5	5	2	78	69.4	69.6	81.1
78				79	79.2	90.7	
8	1	4	1	170	0	3.5	17.5
	2	6	1	170	17.5	18.5	33.5
	3	7	3	57	33.5	34.1	39.767
				57	49.033	49.633	55.3
				57	58.933	59.533	65.2
	4	2	1	170	65.2	67	84
	5	3	2	85	84	85.5	92.5
				85	84	85.5	92.5
6	8	1	170	92.5	93.3	115.3	
9	1	2	2	105	0	1.5	6
				105	15.3	16.8	21.3
	2	1	2	105	6	6.3	15.8
				105	21.3	21.6	31.1
	3	7	2	105	15.8	17.1	27.1
				105	31.1	32.4	42.4
	4	5	2	105	27.1	29.1	32.6
				105	51.467	53.467	56.967
	5	4	1	210	83.5	87.5	102.5
	6	3	2	105	102.5	105.7	116.7
105				102.5	105.7	116.7	
10	1	8	2	53	8.1667	8.5667	15.067
				53	8.1667	8.5667	15.067
	2	5	1	105	15.067	16.267	27.267
	3	3	1	105	39.2	40.2	55.2
	4	2	1	105	68	68.5	76.5
	5	6	2	53	76.5	77.1	87.1
53				95.7	96.3	106.3	

表 4.24 機器之作業排程(GA)

工作中心	機器編號	加工順序	加工作業 (訂單-作業)	整備 開始時間	作業 開始時間	作業 結束時間
1	1	1	9-2	6	6.3	15.8
		2	2-3	19.8	20.1	27.1
		3	1-4	52.3	53	60.5
		4	4-6	116.2	116.7	124.7
	2	1	5-1	0	0.3	15.3
		2	2-3	42.8	43.1	50.1
		3	1-4	52.3	53	60.5
		4	4-6	116.2	116.7	124.7
	3	1	6-1	0	3.2	20.2
		2	9-2	21.3	21.6	31.1
		3	7-3	35.8	36.5	58.5
	2	1	1	7-1	0	0.3
2			9-1	15.3	16.8	21.3
3			3-2	21.3	23.5	32.5
4			2-4	32.5	33	41
5			8-4	65.2	67	84
2		1	9-1	0	1.5	6
		2	6-3	30.8	31.3	48.3
		3	3-2	48.3	50.5	59.5
		4	2-4	59.5	60	68
		5	10-4	68	68.5	76.5
		6	4-4	83.5	86.9	105.9
3		1	1	3-1	0	0.4
	2		2-2	10.9	11.3	19.8
	3		5-2	19.8	20.2	39.2
	4		10-3	39.2	40.2	55.2
	5		8-5	84	85.5	92.5
	6		9-6	102.5	105.7	116.7
	2	1	1-1	0	1	23
		2	3-1	23	23.4	33.9
		3	2-2	33.9	34.3	42.8
		5	9-6	102.5	105.7	116.7
4	1	1	8-1	0	3.5	17.5
		2	6-2	20.2	21.8	30.8
		3	1-2	30.8	32.6	40.6
		4	3-3	59.5	63.5	70.5
		5	4-3	70.5	73.5	83.5
		6	9-5	83.5	87.5	102.5
5	1	1	4-2	45.5	45.8	51.467
		2	5-4	58.933	59.133	68.133
		3	7-5	69.4	69.6	81.1
		4	2-5	81.1	82.1	99.1
	2	1	9-4	27.1	29.1	32.6
		2	4-2	45.5	45.8	51.467
		3	6-5	61.4	61.7	80.7

表 4.24 機器之作業排程(GA)

	3	4	3-4	80.7	81.2	89.2	
		1	10-2	15.067	16.267	27.267	
		2	4-2	45.5	45.8	51.467	
		3	9-4	51.467	53.467	56.967	
		4	5-4	58.933	59.133	68.133	
		5	3-4	70.5	71	79	
		6	7-5	79	79.2	90.7	
6	1	1	1-5	60.5	63.3	76.3	
		2	6-6	80.7	83.7	95.7	
		3	10-5	95.7	96.3	106.3	
	2	1	8-2	17.5	18.5	33.5	
		2	4-1	33.5	34.5	45.5	
		3	10-5	76.5	77.1	87.1	
7	1	1	7-2	15.3	15.8	35.8	
		2	1-3	40.6	41.8	52.3	
		3	5-3	52.3	52.6	58.933	
		4	8-3	58.933	59.533	65.2	
		5	2-6	99.1	99.4	105.73	
		6	4-5	105.9	107.2	116.2	
	2	1	9-3	15.8	17.1	27.1	
		2	8-3	33.5	34.1	39.767	
		3	1-3	40.6	41.8	52.3	
		4	5-3	52.3	52.6	58.933	
		5	2-6	99.1	99.4	105.73	
		6	4-5	105.9	107.2	116.2	
	3	1	9-3	31.1	32.4	42.4	
		2	5-3	42.4	42.7	49.033	
		3	8-3	49.033	49.633	55.3	
		4	2-6	99.1	99.4	105.73	
	8	1	1	2-1	0	0.5	8.1667
			2	10-1	8.1667	8.5667	15.067
3			6-4	48.3	49.4	61.4	
4			3-5	79	81.5	93.5	
2		1	2-1	0	0.5	8.1667	
		2	7-4	58.5	58.9	69.4	
		3	1-6	76.3	76.8	96.8	
		4	3-5	96.8	99.3	111.3	
3		1	2-1	0	0.5	8.1667	
		2	10-1	8.1667	8.5667	15.067	
		3	6-4	48.3	49.4	61.4	
		4	7-4	61.4	61.8	72.3	
		5	8-6	92.5	93.3	115.3	

表 4.25 訂單排程結果(HGA)

訂單	作業	加工 工作中心	加工 機器數	加工 批量	機器 整備時間	作業 開始時間	作業 結束時間
1	1	3	2	60	0	1	12
				60	0	1	12
	2	4	1	120	17.5	19.3	27.3
	3	7	2	60	27.3	28.5	39
				60	27.3	28.5	39
	4	1	3	40	39	39.7	44.7
				40	39	39.7	44.7
				40	39	39.7	44.7
	5	6	1	120	44.7	47.5	60.5
	6	8	2	60	60.5	61	71
60				60.5	61	71	
2	1	8	1	85	0	0.5	23.5
	2	3	2	43	23.5	23.9	32.4
				43	42.3	42.7	51.2
	3	1	3	28	32.4	32.7	37.367
				28	51.2	51.5	56.167
	4	2	2	43	56.167	56.667	64.667
				43	56.167	56.667	64.667
	5	5	1	85	79.3	80.3	97.3
	6	7	2	43	97.3	97.6	107.1
				43	97.3	97.6	107.1
3	1	3	2	100	12	12.4	22.9
				100	12	12.4	22.9
	2	2	2	100	22.9	25.1	34.1
				100	22.9	25.1	34.1
	3	4	1	200	70.9	74.9	81.9
	4	5	2	100	81.9	82.4	90.4
				100	81.9	82.4	90.4
	5	8	2	100	90.4	92.9	104.9
100				90.4	92.9	104.9	
4	1	6	2	70	0	1	6.5
				70	0	1	6.5
	2	5	3	47	6.5	6.8	12.467
				47	6.5	6.8	12.467
				47	6.5	6.8	12.467
	3	4	1	140	37.9	40.9	50.9
	4	2	1	140	84.8	88.2	107.2
	5	7	2	70	107.2	108.5	117.5
				70	107.2	108.5	117.5
	6	1	3	47	117.5	118	123.33
47				117.5	118	123.33	
47				117.5	118	123.33	
5	1	1	1	90	0	0.3	15.3

表 4.25 訂單排程結果(HGA)

	2	3	1	90	22.9	23.3	42.3
	3	7	2	45	42.3	42.6	52.1
				45	49.5	49.8	59.3
	4	5	2	45	52.1	52.3	61.3
45				59.3	59.5	68.5	
6	1	1	1	135	0	3.2	20.2
	2	4	1	135	27.3	28.9	37.9
	3	2	2	68	37.9	38.4	46.9
				68	37.9	38.4	46.9
	4	8	2	68	46.9	48	60
				68	46.9	48	60
5	5	1	135	60	60.3	79.3	
6	6	1	135	79.3	82.3	94.3	
7	1	2	2	78	0	0.3	7.8
				78	10.5	10.8	18.3
	2	7	2	78	39	39.5	49.5
				78	46.4	46.9	56.9
	3	1	2	78	49.5	50.2	61.2
				78	56.9	57.6	68.6
	4	8	2	78	68.6	69	79.5
				78	71	71.4	81.9
5	5	2	78	90.4	90.6	102.1	
			78	90.4	90.6	102.1	
8	1	4	1	170	0	3.5	17.5
	2	6	1	170	17.5	18.5	33.5
	3	7	1	170	56.9	57.5	74.5
	4	2	2	85	74.5	76.3	84.8
				85	74.5	76.3	84.8
	5	3	1	170	84.8	86.3	100.3
	6	8	3	57	100.3	101.1	108.43
57				104.9	105.7	113.03	
			57	104.9	105.7	113.03	
9	1	2	1	210	0	1.5	10.5
	2	1	2	105	10.5	10.8	20.3
				105	15.3	15.6	25.1
	3	7	1	210	25.1	26.4	46.4
	4	5	2	105	46.4	48.4	51.9
				105	46.4	48.4	51.9
5	4	1	210	51.9	55.9	70.9	
6	3	1	210	70.9	74.1	96.1	
10	1	8	1	105	0	0.4	13.4
	2	5	1	105	13.4	14.6	25.6
	3	3	1	105	32.4	33.4	48.4
	4	2	2	53	48.4	48.9	52.9
				53	48.4	48.9	52.9
5	6	1	105	52.9	53.5	73.5	

表 4.26 機器之作業排程(HGA)

工作中心	機器編號	加工順序	加工作業 (訂單-作業)	整備 開始時間	作業 開始時間	作業 結束時間	
1	1	1	9-2	10.5	10.8	20.3	
		2	2-3	32.4	32.7	37.367	
		3	1-4	39	39.7	44.7	
		4	7-3	49.5	50.2	61.2	
		5	4-6	117.5	118	123.33	
	2	1	6-1	0	3.2	20.2	
		2	1-4	39	39.7	44.7	
		3	2-3	51.2	51.5	56.167	
		4	4-6	117.5	118	123.33	
	3	1	5-1	0	0.3	15.3	
		2	9-2	15.3	15.6	25.1	
		3	1-4	39	39.7	44.7	
		4	2-3	51.2	51.5	56.167	
		5	7-3	56.9	57.6	68.6	
		6	4-6	117.5	118	123.33	
	2	1	1	9-1	0	1.5	10.5
			2	7-1	10.5	10.8	18.3
			3	3-2	22.9	25.1	34.1
4			6-3	37.9	38.4	46.9	
5			10-4	48.4	48.9	52.9	
6			2-4	56.167	56.667	64.667	
7			8-4	74.5	76.3	84.8	
2		1	7-1	0	0.3	7.8	
		2	3-2	22.9	25.1	34.1	
		3	6-3	37.9	38.4	46.9	
		4	10-4	48.4	48.9	52.9	
		5	2-4	56.167	56.667	64.667	
		6	8-4	74.5	76.3	84.8	
		7	4-4	84.8	88.2	107.2	
3	1	1	1-1	0	1	12	
		2	3-1	12	12.4	22.9	
		3	5-2	22.9	23.3	42.3	
		4	2-2	42.3	42.7	51.2	
		5	9-6	70.9	74.1	96.1	
	2	1	1-1	0	1	12	
		2	3-1	12	12.4	22.9	
		3	2-2	23.5	23.9	32.4	
		4	10-3	32.4	33.4	48.4	
		5	8-5	84.8	86.3	100.3	
4	1	1	8-1	0	3.5	17.5	
		2	1-2	17.5	19.3	27.3	
		3	6-2	27.3	28.9	37.9	
		4	4-3	37.9	40.9	50.9	
		5	9-5	51.9	55.9	70.9	
		6	3-3	70.9	74.9	81.9	
5	1	1	4-2	6.5	6.8	12.467	

表 4.26 機器之作業排程(HGA)

		2	10-2	13.4	14.6	25.6	
		3	9-4	46.4	48.4	51.9	
		4	6-5	60	60.3	79.3	
		5	2-5	79.3	80.3	97.3	
	2	1	4-2	6.5	6.8	12.467	
		2	5-4	52.1	52.3	61.3	
		3	3-4	81.9	82.4	90.4	
		4	7-5	90.4	90.6	102.1	
	3	1	4-2	6.5	6.8	12.467	
		2	9-4	46.4	48.4	51.9	
		3	5-4	59.3	59.5	68.5	
		4	3-4	81.9	82.4	90.4	
		5	7-5	90.4	90.6	102.1	
6	1	1	4-1	0	1	6.5	
		2	1-5	44.7	47.5	60.5	
	2	1	4-1	0	1	6.5	
		2	8-2	17.5	18.5	33.5	
		3	10-5	52.9	53.5	73.5	
	4	6-6	79.3	82.3	94.3		
7	1	1	1-3	27.3	28.5	39	
		2	7-2	39	39.5	49.5	
		3	5-3	49.5	49.8	59.3	
		4	2-6	97.3	97.6	107.1	
		5	4-5	107.2	108.5	117.5	
	2	1	1-3	27.3	28.5	39	
		2	5-3	42.3	42.6	52.1	
	3	1	9-3	25.1	26.4	46.4	
		2	7-2	46.4	46.9	56.9	
		3	8-3	56.9	57.5	74.5	
		4	2-6	97.3	97.6	107.1	
		5	4-5	107.2	108.5	117.5	
	8	1	1	10-1	0	0.4	13.4
			2	1-6	60.5	61	71
			3	7-4	71	71.4	81.9
4			3-5	90.4	92.9	104.9	
5			8-6	104.9	105.7	113.03	
2		1	2-1	0	0.5	23.5	
		2	6-4	46.9	48	60	
		3	7-4	68.6	69	79.5	
		4	8-6	100.3	101.1	108.43	
3		1	6-4	46.9	48	60	
		2	1-6	60.5	61	71	
		3	3-5	90.4	92.9	104.9	
		4	8-6	104.9	105.7	113.03	

## 4.4 重排程結果

本小節說明本系統於機台故障、缺料、緊急插單三種情況下的重排程步驟與結果。本小節說明此三種重排程時，以 4.3 節混合式遺傳演算法(HGA)所求得之排程結果為「前排程」(參照表 4.25 與表 4.26)。在未發生中斷事件前，生產作業皆依照此「前排程」生產；發生中斷事件後，以本系統重排程的方式進行重排，重排完成後，生產現場則依此重排後的排程結果持續作業生產。

### 4.4.1 機台故障重排程

本小節說明機台故障重排程與結果，其重排方法依照 3.3.1 小節的機台故障修正式重排法進行重排。機台故障的情境，以隨機方式從 8 種工作中心的所有平行機台中選取一台作為故障機台，本例選取工作中心 1 編號 3 的平行機台做為故障機台；以均勻分配隨機從此機台的工作時程中選取一時間點作為機台當機的時間點，本例之機台故障時間點為「23.8」；令此故障時間點上的作業為「中斷作業」，本例的中斷作業為訂單 9 的作業 2；機台故障狀態的持續時間指的是機台維修所需時間，通常依現場人員預估修復時間，本例則指定為 8。本例隨機假設的機台故障情境如下表 4.27 所示：

表 4.27 機台故障情境假設

機台故障情境假設		
故障機台	工作中心	1
	平行機台編號	3
機台故障時間點	23.8	
中斷作業	訂單 9 的作業 2 的第二個批量作業	
機台故障狀態持續時間 (機台維修時間)	8	

機台故障的重排步驟結果如下。

#### 步驟 1、將不需重排的作業移除

將「作業開始整備時間」早於機台故障時間點「23.8」的所有批量作業從重排程問題中移除，這些批量作業已生產完成或是已開始上機生產，不受此機台故障的中斷事件影響。移除的批量作業以(訂單-作業)表示如下：(1-1)、(1-2)、(2-1)、(2-2) 的批量二、(3-1)、(3-2)、(4-1)、(4-2)、(5-1)、(5-2)、(6-1)、(7-1)、(8-1)、(8-2)、(9-1)、(9-2)、(10-1)、(10-2)。

特別注意的是，作業 2-2 並不是完全從重排程問題中移除，因為作業 2-2 第二個批量的「作業開始整備時間」為 42.3 (大於 23.8)，因此作業 2-2 的第二個批量仍然需要納入重排。

### 步驟 2、修正中斷作業的重新開始時間與作業結束時間

修正中斷作業於機台修復後的重新開始時間與此中斷作業的作業結束時間。

重新開始時間： $23.8+8=31.8$

修正後的作業結束時間： $31.8+(25.1-23.8)=33.1$

### 步驟 3、取得重排程時的作業排程優先順序

取得需重排作業的作業排程優先順序。依照原前排程的作業排程優先順序(即演算法中的上半部基因)，在不改變相對順序的原則下進行修正，結果如下：

原前排程的作業排程優先順序如下表 4.28。

表 4.28 前排程各作業的排程優先順序

作業	1-1	1-2	1-3	1-4	1-5	1-6	2-1	2-2	2-3	2-4
順序	2	7	11	15	27	29	3	30	34	37
作業	2-5	2-6	3-1	3-2	3-3	3-4	3-5	4-1	4-2	4-3
順序	45	48	8	17	38	40	42	5	6	31
作業	4-4	4-5	4-6	5-1	5-2	5-3	5-4	6-1	6-2	6-3
順序	39	41	50	9	23	49	52	4	12	16
作業	6-4	6-5	6-6	7-1	7-2	7-3	7-4	7-5	8-1	8-2
順序	20	22	46	13	24	35	43	47	1	21
作業	8-3	8-4	8-5	8-6	9-1	9-2	9-3	9-4	9-5	9-6
順序	25	36	51	54	10	14	18	19	26	32
作業	10-1	10-2	10-3	10-4	10-5					
順序	28	33	44	53	55					

需重排的作業，於原前排程的作業排程優先順序如下表 4.29。

表 4.29 需重排作業原先的排程順序

作業	1-3	1-4	1-5	1-6	2-2	2-3	2-4	2-5	2-6	3-3
順序	11	15	27	29	30	34	37	45	48	38
作業	3-4	3-5	4-3	4-4	4-5	4-6	5-3	5-4	6-2	6-3
順序	40	42	31	39	41	50	49	52	12	16
作業	6-4	6-5	6-6	7-2	7-3	7-4	7-5	8-3	8-4	8-5
順序	20	22	46	24	35	43	47	25	36	51
作業	8-6	9-3	9-4	9-5	9-6	10-3	10-4	10-5		
順序	54	18	19	26	32	44	53	55		

修正後，重排程的作業排程優先順序如下表 4.30。

表 4.30 需重排作業的排程順序

作業	1-3	1-4	1-5	1-6	2-2	2-3	2-4	2-5	2-6	3-3
順序	1	3	12	13	14	17	20	28	31	21
作業	3-4	3-5	4-3	4-4	4-5	4-6	5-3	5-4	6-2	6-3
順序	23	25	15	22	24	33	32	35	2	4
作業	6-4	6-5	6-6	7-2	7-3	7-4	7-5	8-3	8-4	8-5
順序	7	8	29	9	18	26	30	10	19	34
作業	8-6	9-3	9-4	9-5	9-6	10-3	10-4	10-5		
順序	37	5	6	11	16	27	36	38		

#### 步驟 4、依照上述所得的作業排程優先順序進行作業排程

將表 4.30 重排程的作業排程優先順序代入作業排程步驟(即演算法的解碼流程步驟)，批量作業的平行機台指派(即演算法的下半部基因)則與原前排程相同，不改變指派的機台號碼與機台數。

本機台故障例子的重排程結果，以機器排程的方式呈現如下表 4.31，此重排後的目標函數值為 0.8331。其中，我們比較重排程前後的作業異動狀況，此次重排程需重排作業的總數為 38，重排後作業異動的總數為 19，以此例觀察，有一半的作業無異動，不受重排程影響，顯示依此修正重排法來進行重排，確實可保留前排程多數的作業不受重排程影響而異動。

表 4.31 機器之作業排程(機台故障重排程結果)

工作中心	機器編號	加工順序	加工作業 (訂單-作業)	整備 開始時間	作業 開始時間	作業 結束時間	
1	1	1	9-2	10.5	10.8	20.3	
		2	2-3	32.4	32.7	37.367	
		3	1-4	39	39.7	44.7	
		4	7-3	49.5	50.2	61.2	
		5	4-6	125.5	126	131.33	
	2	1	6-1	0	3.2	20.2	
		2	1-4	39	39.7	44.7	
		3	2-3	51.2	51.5	56.167	
		4	4-6	125.5	126	131.33	
	3	1	5-1	0	0.3	15.3	
		2	9-2	15.3	15.6	33.1	
		3	1-4	39	39.7	44.7	
		4	2-3	51.2	51.5	56.167	
		5	7-3	64.9	65.6	76.6	
		6	4-6	125.5	126	131.33	
	2	1	1	9-1	0	1.5	10.5
			2	7-1	10.5	10.8	18.3

表 4.31 機器之作業排程(機台故障重排程結果)

		3	3-2	22.9	25.1	34.1
		4	6-3	37.9	38.4	46.9
		5	10-4	48.4	48.9	52.9
		6	2-4	56.167	56.667	64.667
		7	8-4	82.5	84.3	92.8
		1	7-1	0	0.3	7.8
		2	3-2	22.9	25.1	34.1
	2	3	6-3	37.9	38.4	46.9
		4	10-4	48.4	48.9	52.9
		5	2-4	56.167	56.667	64.667
		6	8-4	82.5	84.3	92.8
		7	4-4	92.8	96.2	115.2
		1	1-1	0	1	12
		3	1	2	3-1	12
3	5-2			22.9	23.3	42.3
4	2-2			42.3	42.7	51.2
5	9-6			78.9	82.1	104.1
1	1-1			0	1	12
2	2		3-1	12	12.4	22.9
	3		2-2	23.5	23.9	32.4
	4		10-3	32.4	33.4	48.4
	5		8-5	92.8	94.3	108.3
	4		1	1	8-1	0
2		1-2		17.5	19.3	27.3
3		6-2		27.3	28.9	37.9
4		4-3		37.9	40.9	50.9
5		9-5		59.9	63.9	78.9
6		3-3		78.9	82.9	89.9
5	1	1	4-2	6.5	6.8	12.467
		2	10-2	13.4	14.6	25.6
		3	9-4	54.4	56.4	59.9
		4	6-5	60	60.3	79.3
		5	2-5	79.3	80.3	97.3
	2	1	4-2	6.5	6.8	12.467
		2	5-4	52.1	52.3	61.3
		3	3-4	89.9	90.4	98.4
		4	7-5	98.4	98.6	110.1
	3	1	4-2	6.5	6.8	12.467
		2	9-4	54.4	56.4	59.9
		3	5-4	59.9	60.1	69.1
		4	3-4	89.9	90.4	98.4
		5	7-5	98.4	98.6	110.1
	6	1	1	4-1	0	1
2			1-5	44.7	47.5	60.5
2		1	4-1	0	1	6.5
		2	8-2	17.5	18.5	33.5
		3	10-5	52.9	53.5	73.5
4	6-6	79.3	82.3	94.3		
7	1	1	1-3	27.3	28.5	39

表 4.31 機器之作業排程(機台故障重排程結果)

		2	7-2	39	39.5	49.5	
		3	5-3	49.5	49.8	59.3	
		4	2-6	97.3	97.6	107.1	
		5	4-5	115.2	116.5	125.5	
		1	1-3	27.3	28.5	39	
	2	2	5-3	42.3	42.6	52.1	
		1	9-3	33.1	34.4	54.4	
	3	2	7-2	54.4	54.9	64.9	
		3	8-3	64.9	65.5	82.5	
		4	2-6	97.3	97.6	107.1	
		5	4-5	115.2	116.5	125.5	
		1	10-1	0	0.4	13.4	
	8	1	2	1-6	60.5	61	71
			3	7-4	71	71.4	81.9
			4	3-5	98.4	100.9	112.9
5			8-6	112.9	113.7	121.03	
1			2-1	0	0.5	23.5	
2		2	6-4	46.9	48	60	
		3	7-4	76.6	77	87.5	
		4	8-6	108.3	109.1	116.43	
		1	6-4	46.9	48	60	
3		2	1-6	60.5	61	71	
		3	3-5	98.4	100.9	112.9	
		4	8-6	112.9	113.7	121.03	
		1	10-1	0	0.4	13.4	

#### 4.4.2 物料短缺重排程

本小節說明物料短缺的重排程與結果，其重排方法依照 3.3.2 小節的缺料修正式重排法進行重排。物料短缺的情境，以隨機方式選取一訂單的作業當作發生缺料的作業，本例以訂單 2 的第 3 個作業作為發生物料短缺的作業，生產現場欲執行此作業時，由於物料不足或是物料尚未入廠，而無法生產；此項缺料的物料到達時間則可隨意指定，本例指定物料到達時間為 53。

表 4.32 物料短缺情境假設

物料短缺情境假設		
發生缺料的作業	訂單	2
	作業	3
原始前排程中 開始整備時間	32.4	
缺料到達時間點	53	

以缺料而言，其直接影響的是該缺料訂單的作業，其他作業則不受影

響可以先行生產。在排程的調整上，則分成兩個階段。階段一是先將該缺料訂單的作業移除，其餘之訂單作業往前遞補，在該缺料尚未到達之前，生產現場依此階段一調整後的排程進行生產。階段二則是於該缺料到達的時間點後，將此缺料訂單的作業插入排程。物料短缺的兩階段重排如下。

### 缺料階段一重排程

物料短缺階段一重排程的步驟結果如下。

#### 步驟 1、將不需重排的作業移除

假設管理者於發現缺料狀況時的第一時間，就馬上進行重排程。在重排的步驟上，先將「作業開始整備時間」小於「發現缺料時間」的所有批量作業從重排程問題中移除，這些批量作業已生產完成或是已開始上機生產，不受此缺料事件的影響。本例假設管理者並未提早發現該作業缺料的情況，而是生產線員工欲將該作業上機整備的前一刻才發現缺料，此時「發現缺料時間」則為該作業於前排程第一批(最早的批量)的作業開始整備時間。本例訂單 2 作業 3 於前排程各批量的「作業開始整備時間」如下表 4.33，因此此例訂「發現缺料時間」為「32.4」。

表 4.33 訂單 2 作業 3 於前排程的「作業開始整備時間」

前排程(參照表 4.25)	
	作業開始整備時間
批量 1	32.4
批量 2	51.2
批量 3	51.2

指定「發現缺料時間」後，我們將「作業開始整備時間」小於「發現缺料時間」的所有批量作業從重排問題中移除，並將該缺料訂單的所有作業也移除，以讓其他作業往前遞補。移除後，在階段一需要修正重排的作業以(訂單-作業)表示如下表 4.34：

表 4.34 階段一需重排的作業

作業	1-4	1-5	1-6	2-2	3-3	3-4	3-5	4-3	4-4	4-5
作業	4-6	5-3	5-4	6-3	6-4	6-5	6-6	7-2	7-3	7-4
作業	7-5	8-3	8-4	8-5	8-6	9-4	9-5	9-6	10-4	10-5

#### 步驟 2、取得重排程時的作業排程優先順序

取得需重排作業的作業排程優先順序。依照原前排程的作業排程優先順序(即演算法中的上半部基因)，在不改變相對順序的原則下進行修正，過

程同上節機台故障修正重排的步驟3。本例以最後結果表示如下表 4.35：

表 4.35 階段一需重排作業的排程優先順序

作業	1-4	1-5	1-6	2-2	3-3	3-4	3-5	4-3	4-4	4-5
順序	1	9	10	11	16	18	20	12	17	19
作業	4-6	5-3	5-4	6-3	6-4	6-5	6-6	7-2	7-3	7-4
順序	25	24	27	2	4	5	22	6	14	21
作業	7-5	8-3	8-4	8-5	8-6	9-4	9-5	9-6	10-4	10-5
順序	23	7	15	26	29	3	8	13	28	30

### 步驟3、依照上述所得的作業排程優先順序進行作業排程

將表 4.35 重排程的作業排程優先順序代入作業排程步驟(即演算法的解碼流程步驟)，批量作業的平行機台指派(即演算法的下半部基因)則與原前排程相同，不改變指派的機台號碼與機台數。

本缺料例子的階段一重排程結果，以機器排程的方式呈現如下表 4.36，在該缺料的物料尚未到達之前，生產現場以此階段一重排程結果為排程依據，持續進行生產。

表 4.36 機器之作業排程(物料短缺階段一重排結果)

工作中心	機器編號	加工順序	加工作業 (訂單-作業)	整備 開始時間	作業 開始時間	作業 結束時間
1	1	1	9-2	10.5	10.8	20.3
		2	1-4	39	39.7	44.7
		3	7-3	49.5	50.2	61.2
		4	4-6	83.6	84.1	89.43
	2	1	6-1	0	3.2	20.2
		2	1-4	39	39.7	44.7
		3	4-6	84.8	85.3	90.63
	3	1	5-1	0	0.3	15.3
		2	9-2	15.3	15.6	33.1
		3	1-4	39	39.7	44.7
		4	7-3	56.9	57.6	68.6
		5	4-6	84.8	85.3	90.63
2	1	1	9-1	0	1.5	10.5
		2	7-1	10.5	10.8	18.3
		3	3-2	22.9	25.1	34.1
		4	6-3	37.9	38.4	46.9
		5	10-4	48.4	48.9	52.9
		6	8-4	74.5	76.3	84.8
	2	1	7-1	0	0.3	7.8
		2	3-2	22.9	25.1	34.1
		3	6-3	37.9	38.4	46.9
		4	4-4	50.9	54.3	73.3
		5	8-4	74.5	76.3	84.8
		6	10-4	84.8	85.3	89.3

表 4.36 機器之作業排程(物料短缺階段一重排結果)

3	1	1	1-1	0	1	12	
		2	3-1	12	12.4	22.9	
		3	5-2	22.9	23.3	42.3	
		4	2-2	42.3	42.7	51.2	
		5	9-6	70.9	74.1	96.1	
	2	1	1-1	0	1	12	
		2	3-1	12	12.4	22.9	
		3	2-2	23.5	23.9	32.4	
		4	10-3	32.4	33.4	48.4	
		5	8-5	84.8	86.3	100.3	
4	1	1	8-1	0	3.5	17.5	
		2	1-2	17.5	19.3	27.3	
		3	6-2	27.3	28.9	37.9	
		4	4-3	37.9	40.9	50.9	
		5	9-5	51.9	55.9	70.9	
		6	3-3	70.9	74.9	81.9	
5	1	1	4-2	6.5	6.8	12.467	
		2	10-2	13.4	14.6	25.6	
		3	9-4	46.4	48.4	51.9	
		4	6-5	60	60.3	79.3	
	2	1	4-2	6.5	6.8	12.467	
		2	5-4	52.1	52.3	61.3	
		3	3-4	81.9	82.4	90.4	
		4	7-5	90.4	90.6	102.1	
	3	1	4-2	6.5	6.8	12.467	
		2	9-4	46.4	48.4	51.9	
		3	5-4	59.3	59.5	68.5	
		4	3-4	81.9	82.4	90.4	
		5	7-5	90.4	90.6	102.1	
	6	1	1	4-1	0	1	6.5
			2	1-5	44.7	47.5	60.5
2		1	4-1	0	1	6.5	
		2	8-2	17.5	18.5	33.5	
		3	6-6	79.3	82.3	94.3	
4	10-5	94.3	94.9	114.9			
7	1	1	1-3	27.3	28.5	39	
		2	7-2	39	39.5	49.5	
		3	5-3	49.5	49.8	59.3	
		4	4-5	73.3	74.6	83.6	
	2	1	1-3	27.3	28.5	39	
		2	5-3	42.3	42.6	52.1	
	3	1	9-3	25.1	26.4	46.4	
		2	7-2	46.4	46.9	56.9	
		3	8-3	56.9	57.5	74.5	
		4	4-5	74.5	75.8	84.8	
	8	1	1	10-1	0	0.4	13.4
			2	1-6	60.5	61	71
3			7-4	71	71.4	81.9	
4			3-5	90.4	92.9	104.9	

表 4.36 機器之作業排程(物料短缺階段一重排結果)

		5	8-6	104.9	105.7	113.03
	2	1	2-1	0	0.5	23.5
		2	6-4	46.9	48	60
		3	7-4	68.6	69	79.5
		4	8-6	100.3	101.1	108.43
	3	1	6-4	46.9	48	60
		2	1-6	60.5	61	71
		3	3-5	90.4	92.9	104.9
		4	8-6	104.9	105.7	113.03

**缺料階段二重排程**

在該缺料作業的物料到達後，立即將該缺料訂單尚未完成的作業重新插入排程中。物料短缺階段二重排程的步驟結果如下。

**步驟 1、將不需重排的作業移除**

本例指定的物料到達時間為「53」，在時間為「53」之前，生產作業皆依照上階段所求得之階段一排程結果(表 4.36)為依據進行生產。因此於階段二的重排中，我們將「作業開始整備時間」早於「53」的所有批量作業從階段二重排程問題中移除，這些批量作業已生產完成或是已開始上機生產。

將缺料訂單尚未完成的作業納入需要重排的作業集合中，得到階段二需要修正重排的作業如下，以(訂單-作業)表示如表 4.37：

表 4.37 階段二需重排的作業

作業	1-6	3-3	3-4	3-5	4-5	4-6	5-4	6-5	6-6	7-3
作業	7-4	7-5	8-3	8-4	8-5	8-6	9-6	10-4	10-5	2-3
作業	2-4	2-5	2-6							

**步驟 2、取得重排程時的作業排程優先順序**

取得需重排作業的作業排程優先順序。由於缺料訂單的作業時程已延遲，因此，我們將其作業排程的優先順序設為最高；其他作業的順序則在此缺料訂單作業之後，但仍依照前排程的作業排程優先順序，以不改變相對順序的原則下進行修正。所得結果表示如下表 4.38：

表 4.38 階段二需重排作業的排程優先順序

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

### 步驟 3、依上述所得的作業排程優先順序，以 GA 重新求解

取得作業排程優先順序後，我們可以利用原前排程指定的作業分割批量數與機台指派資訊(下半部基因)來進行重排，但由於缺料訂單的作業時程已延遲，為了加快其生產速度，傳統作法上排程人員可能將該缺料訂單作業的分割批量數設為最大，即給予最多的機台資源來趕工生產。雖然此作法對於該缺料訂單的交期滿足是有幫助的，但是相對於其他訂單的交期考量，給予缺料單作業最多的機台資源可能會造成其他訂單作業的生產時程必須往後遞延，間接降低其他訂單的達交率，進而影響到整體目標函數值。

為了決定較客觀的訂單作業分割批量數，使重排時能取得較佳的整體目標函數值，而非只單純考量缺料訂單的交期，我們將需重排作業(表 4.37)的分割批量數與機台指派值(下半部基因)以 GA 演化求解，求解時各作業的排程優先順序(上半部基因)則維持不動。此外，在求解過程中我們限制缺料作業的分割批量數不可小於原前排程的分割數，因為若再降低缺料作業的分割批數只會造成缺料作業的交期更往後延；非缺料作業的分割批量數則不限制。

由於此時 GA 只對下半部基因進行求解，且在加入上述限制後，求解空間變得非常的小，因此求解只需少許代數即可收斂。本例設定 50 代求解，求解過程如圖 4.2，求解結果如表 4.40，GA 搜尋時間為 178.6 秒。

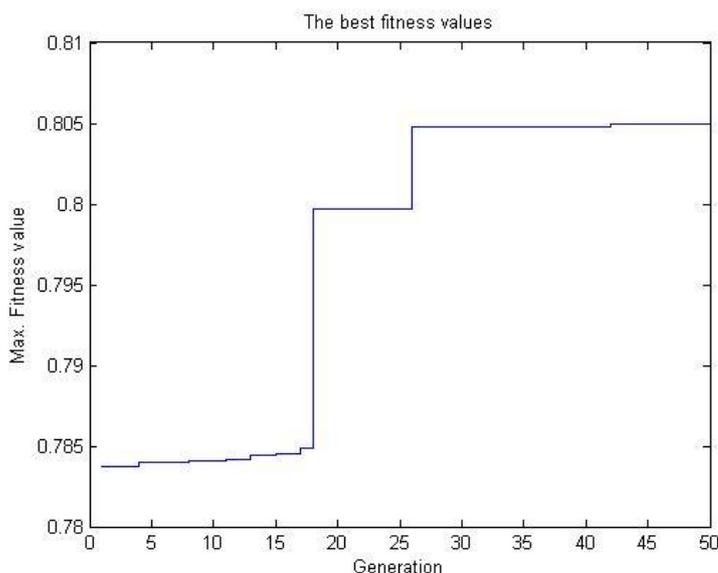


圖 4.2 缺料階段二重排-GA 求解過程

此外，本例以三種不同的分割批量數決定方式，進行重排與比較，比

較結果如表 4.39，由表中結果可知，批量分割數與機台指派值以 GA 重新求解的方式來決定並重排，確實可得到較好的重排程解。

表 4.39 缺料階段二重排-各方式結果比較

重排方式	設分割批量數與前排程相同	設定缺料作業分割批量數為最大	以 GA 重新求解分割批量數
重排後目標函數值	<b>0.7849</b>	<b>0.7567</b>	<b>0.8050</b>

本缺料例子的階段二重排程結果，以機器排程的方式呈現如下表 4.40。由重排後的結果來觀察，該缺料作業 2-3 之途程為工作中心 1 的 1 至 3 號平行機台，由階段一的排程結果(表 4.36)可知，工作中心 1 的編號 1 機台在時間 53 時正好有作業 7-3 在進行中，未了避免中斷目前正在機台上生產的作業，缺料作業 2-3 於 1 號機台的批量，將等到作業 7-3 結束後才插入；缺料作業 2-3 的另外兩個批量則於本例指定的物料到達時間「53」插入排程。

我們比較重排程前後的作業異動狀況，以缺料的階段一重排程觀察，階段一需重排作業的總數為 30，重排後作業異動的總數為 5，以此例觀察，有大多數的作業無異動，不受重排程影響，顯示依修正重排法來進行重排，確實可保留前排程多數的作業不受重排程影響而異動。以缺料的階段二重排程觀察，重排程需重排作業的總數為 23，重排後作業異動的總數為 15，以此例觀察，雖然有部份作業不受重排程影響，但是由於缺料階段二重排利用 GA 重新搜尋批量分割數與機台指派值，重排後多數的批量分割數和機台指派值與前排程相異，造成大部份作業異動的結果。

表 4.40 機器之作業排程(物料短缺階段二重排結果)

工作中心	機器編號	加工順序	加工作業(訂單-作業)	整備開始時間	作業開始時間	作業結束時間
1	1	1	9-2	10.5	10.8	20.3
		2	1-4	39	39.7	44.7
		3	7-3	49.5	50.2	61.2
		4	2-3	61.2	61.5	66.167
		5	4-6	96.2	96.7	112.7
	2	1	6-1	0	3.2	20.2
		2	1-4	39	39.7	44.7
		3	2-3	53	53.3	57.967
	3	1	5-1	0	0.3	15.3
		2	9-2	15.3	15.6	25.1
		3	1-4	39	39.7	44.7
		4	2-3	53	53.3	57.967

表 4.40 機器之作業排程(物料短缺階段二重排結果)

		5	7-3	57.967	58.667	69.667
2	1	1	9-1	0	1.5	10.5
		2	7-1	10.5	10.8	18.3
		3	3-2	22.9	25.1	34.1
		4	6-3	37.9	38.4	46.9
		5	10-4	48.4	48.9	52.9
		6	2-4	57.967	58.467	66.467
		7	8-4	76.9	78.7	87.2
	2	1	7-1	0	0.3	7.8
		2	3-2	22.9	25.1	34.1
		3	6-3	37.9	38.4	46.9
		4	4-4	50.9	54.3	73.3
		5	2-4	73.3	73.8	81.8
		6	8-4	81.8	83.6	92.1
		7	10-4	92.1	92.6	96.6
3	1	1	1-1	0	1	12
		2	3-1	12	12.4	22.9
		3	5-2	22.9	23.3	42.3
		4	2-2	42.3	42.7	51.2
		5	9-6	70.9	74.1	96.1
	2	1	1-1	0	1	12
		2	3-1	12	12.4	22.9
		3	2-2	23.5	23.9	32.4
		4	10-3	32.4	33.4	48.4
		5	8-5	92.1	93.6	107.6
4	1	1	8-1	0	3.5	17.5
		2	1-2	17.5	19.3	27.3
		3	6-2	27.3	28.9	37.9
		4	4-3	37.9	40.9	50.9
		5	9-5	51.9	55.9	70.9
		6	3-3	70.9	74.9	81.9
5	1	1	4-2	6.5	6.8	12.467
		2	10-2	13.4	14.6	25.6
		3	9-4	46.4	48.4	51.9
		4	6-5	60	60.3	79.3
		5	2-5	81.8	82.8	99.8
	2	1	4-2	6.5	6.8	12.467
		2	5-4	52.1	52.3	61.3
		3	3-4	81.9	82.4	90.4
		4	7-5	90.4	90.6	102.1
	3	1	4-2	6.5	6.8	12.467
		2	9-4	46.4	48.4	51.9
		3	5-4	59.3	59.5	68.5
		4	3-4	81.9	82.4	90.4
		5	7-5	90.4	90.6	102.1
	6	1	1	4-1	0	1
2			1-5	44.7	47.5	60.5
2		1	4-1	0	1	6.5
		2	8-2	17.5	18.5	33.5

表 4.40 機器之作業排程(物料短缺階段二重排結果)

		3	6-6	79.3	82.3	94.3
		4	10-5	96.6	97.2	117.2
7	1	1	1-3	27.3	28.5	39
		2	7-2	39	39.5	49.5
		3	5-3	49.5	49.8	59.3
		4	8-3	59.3	59.9	76.9
		5	4-5	76.9	78.2	96.2
		6	2-6	99.8	100.1	109.6
	2	1	1-3	27.3	28.5	39
		2	5-3	42.3	42.6	52.1
		3	2-6	99.8	100.1	109.6
	3	1	9-3	25.1	26.4	46.4
2		7-2	46.4	46.9	56.9	
8	1	1	10-1	0	0.4	13.4
		2	1-6	60.5	61	71
		3	3-5	90.4	92.9	104.9
		4	8-6	107.6	108.4	115.73
	2	1	2-1	0	0.5	23.5
		2	6-4	46.9	48	60
		3	1-6	60.5	61	71
		4	7-4	71	71.4	81.9
		5	3-5	90.4	92.9	104.9
		6	8-6	107.6	108.4	115.73
	3	1	6-4	46.9	48	60
		2	7-4	69.667	70.067	80.567
		3	8-6	107.6	108.4	115.73

### 4.4.3 緊急插單重排程

本小節說明機台故障重排程與結果，其重排方法依照 3.3.3 小節的緊急插單完全重排法進行重排。緊急插單的訂單資料可隨意指定，本例則指定如下表 4.41，以此插單例子來說明緊急插單重排程的步驟結果。

表 4.41 緊急插單資料

緊急訂單資料	
得知緊急訂單的時間	50
交期	98.5
總途程數	5
作業途程	1,5,3,2,6
數量	150
各作業總加工時間	17,21,12,10,16
各作業整備時間	0.6, 0.5, 1.2, 0.8, 0.5
各作業上機批量限制	48,42,134,69,72

#### 步驟 1、將不需重排的作業移除

假設在未得知緊急插單前，生產線皆依前排程順利生產，因此我們將「作業開始整備時間」早於「50」的所有批量作業從重排程問題中移除，這些批量作業已生產完成或是已開始上機生產，不受此緊急插單的中斷事件影響。移除後，需要重排的作業以(訂單-作業)表示如下表 4.42：

表 4.42 需重排的作業

作業	1-6	2-3	2-4	2-5	2-6	3-3	3-4	3-5	4-4
作業	4-5	4-6	5-4	6-5	6-6	7-3	7-4	7-5	8-3
作業	8-4	8-5	8-6	9-5	9-6	10-5	插單-1	插單-2	插單-3
作業	插單-4	插單-5							

#### 步驟 2、修正訂單的定性因素評估值

加入緊急插單後，總訂單數變為 11，由於加入緊急插單的質性考量，各訂單的定性評估值也會因此改變，因此必須以 AHP 法重新評估各訂單的質性因素。而在得知該緊急訂單的時間點 50 時，尚未有任何一個訂單完全完成，我們將該筆插單一起納入 AHP 重新評估，總共評估 11 張訂單的定性因素。重新評估的過程中，原始 10 張訂單於各因素的兩兩成對比較結果並不改變，只需考量該插單與其他訂單相互間的成對比較。重新評估的過程結果如表 4.43 至表 4.48 所示。

表 4.43 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	Order_11	$\lambda=17.0071$	評估值
Order_1	1	3	1/2	1	4	5	4	5	3	5	1	0.4111	0.1344
Order_2	1/3	1	1/3	1/2	2	4	1	3	1/2	4	1/5	0.2017	0.0660
Order_3	2	3	1	2	4	5	4	4	3	5	1	0.4600	0.1504
Order_4	1	2	1/2	1	4	5	3	4	2	5	1/3	0.3459	0.1131
Order_5	4	1/2	1/4	1/4	1	3	1/2	2	1/3	1/2	1/4	0.1978	0.0647
Order_6	5	1/4	1/5	1/5	1/3	1	1/3	1/2	1/4	1/2	1/4	0.1721	0.0563
Order_7	1/4	1	1/4	1/3	2	3	1	2	1/2	3	1/5	0.1627	0.0532
Order_8	1/5	1/3	1/4	4	1/2	2	1/2	1	1/3	1/2	1/4	0.1552	0.0508
Order_9	1/3	2	1/3	2	3	4	2	3	1	4	1/4	0.2747	0.0898
Order_10	1/5	1/4	1/5	5	2	2	1/3	2	1/4	1	1/3	0.2052	0.0671
Order_11	1	5	1	3	4	4	5	4	4	3	1	0.4714	0.1542

表 4.44 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	Order_11	$\lambda=11.3549$	評估值
Order_1	1	2	1/3	1/4	2	3	1/3	3	1/2	4	2	0.2216	0.0820
Order_2	1/2	1	1/3	1/4	1	2	1/3	2	1/2	3	1	0.1520	0.0563
Order_3	3	3	1	1/2	3	4	1	5	2	5	4	0.4417	0.1635
Order_4	4	4	2	1	4	4	2	5	3	5	5	0.6284	0.2325
Order_5	1/2	1	1/3	1/4	1	2	1/3	2	1/2	3	1	0.1520	0.0563
Order_6	1/3	1/2	1/4	1/4	1/2	1	1/4	2	1/3	2	1	0.1088	0.0403
Order_7	3	3	1	1/2	3	4	1	5	2	5	4	0.4417	0.1635
Order_8	1/3	1/2	1/5	1/5	1/2	1/2	1/5	1	1/4	2	1	0.0875	0.0324
Order_9	2	2	1/2	1/3	2	3	1/2	4	1	4	3	0.2887	0.1068
Order_10	1/4	1/3	1/5	1/5	1/3	1/2	1/5	1/2	1/4	1	1	0.0713	0.0264
Order_11	1/2	1	1/4	1/5	1	1	1/4	1	1/3	1	1	0.1086	0.0402

表 4.45 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	Order_11	$\lambda=12.3945$	評估值
Order_1	1	4	3	2	5	3	4	1/2	2	5	1	0.4297	0.1524
Order_2	1/4	1	1/2	1/3	3	1/2	2	1/5	1/3	2	1/2	0.1272	0.0451
Order_3	1/3	2	1	1/3	4	1/2	3	1/4	1/2	3	1/5	0.1681	0.0596
Order_4	1/2	3	3	1	5	2	4	1/2	2	4	2	0.3757	0.1333
Order_5	1/5	1/3	1/4	1/5	1	1/4	1/2	1/7	1/4	1/2	1/3	0.0600	0.0213
Order_6	1/3	2	2	1/2	4	1	3	1/4	1/2	4	1/2	0.2113	0.0749
Order_7	1/4	1/2	1/3	1/4	2	1/3	1	1/5	1/4	2	2	0.1393	0.0494
Order_8	2	5	4	2	7	4	5	1	3	5	2	0.6001	0.2128
Order_9	1/2	3	2	1/2	4	2	4	1/3	1	4	1/4	0.2631	0.0933
Order_10	1/5	1/2	1/3	1/4	2	1/4	2	1/5	1/4	1	1/3	0.0925	0.0328
Order_11	1	2	5	1/2	3	2	1/2	1/2	4	3	1	0.3524	0.1250

表 4.46 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	Order_11	$\lambda=11.6628$	評估值
Order_1	1	0.5	0.25	0.2	0.5	0.33	0.25	2	0.25	3	1/3	0.1017	0.0367
Order_2	2	1	0.33	0.2	1	0.2	0.33	2	0.5	3	1/3	0.1346	0.0486
Order_3	4	3	1	0.5	3	2	1	4	0.5	5	1	0.3250	0.1174
Order_4	5	5	2	1	5	3	2	5	2	7	2	0.5874	0.2122
Order_5	2	1	0.33	0.2	1	0.5	0.33	2	0.25	3	1/3	0.1291	0.0466
Order_6	3	2	0.5	0.33	2	1	0.5	3	0.33	4	1/2	0.2062	0.0745
Order_7	4	3	1	1/2	3	2	1	4	1/2	5	1/2	0.3114	0.1125
Order_8	1/2	1/2	1/4	1/5	1/2	1/3	1/4	1	1/5	2	1/3	0.0821	0.0297
Order_9	4	4	2	1/2	4	3	2	5	1	5	3	0.5028	0.1817
Order_10	1/3	1/3	1/5	1/7	1/3	1/4	1/5	2	1/5	1	1/3	0.0708	0.0256
Order_11	3	3	1	1/2	3	2	2	3	1/3	3	1	0.3166	0.1144

表 4.47 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
權重	0.47	0.16	0.30	0.07		
Order_1	0.1344	0.0820	0.1524	0.0367	0.1246	2
Order_2	0.0660	0.0563	0.0451	0.0486	0.0570	9
Order_3	0.1504	0.1635	0.0596	0.1174	0.1229	4
Order_4	0.1131	0.2325	0.1333	0.2122	0.1452	1
Order_5	0.0647	0.0563	0.0213	0.0466	0.0491	10
Order_6	0.0563	0.0403	0.0749	0.0745	0.0606	8
Order_7	0.0532	0.1635	0.0494	0.1125	0.0739	7
Order_8	0.0508	0.0324	0.2128	0.0297	0.0950	6
Order_9	0.0898	0.1068	0.0933	0.1817	0.1000	5
Order_10	0.0671	0.0264	0.0328	0.0256	0.0474	11
Order_11	0.1542	0.0402	0.1250	0.1144	0.1244	3

表 4.48 定性評估之訂單順序結果

定單順序	1	2	3	4	5	6	7	8	9	10	11
訂單編號	訂單 4	訂單 1	訂單 11	訂單 3	訂單 9	訂單 8	訂單 7	訂單 6	訂單 2	訂單 5	訂單 10

在加入該筆緊急插單後，重新以 AHP 評估各訂單的權重與順序，所得的結果如下表 4.50，原始前排程的 AHP 評估結果則如表 4.49：

表 4.49 原前排程的定性評結果

訂單編號	1	2	3	4	5	6	7	8	9	10
訂單加權值	0.1447	0.0656	0.1403	0.1637	0.0562	0.0699	0.0776	0.1114	0.1162	0.0544
訂單順序	2	8	3	1	9	7	6	5	4	10

表 4.50 加入緊急插單後的定性評估結果

訂單編號	1	2	3	4	5	6	7	8	9	10	緊急插單
訂單加權值	0.1246	0.0570	0.1229	0.1452	0.0491	0.0606	0.0739	0.0950	0.1000	0.0474	0.1244
訂單順序	2	9	4	1	10	8	7	6	5	11	3

### 步驟 3、將需重排的作業代入演算法求解

將表 4.42 需重排的作業與表 4.50 修正後的定性評估結果，帶入演算法求解即可得重排程結果，本例以 GA 進行求解，設定搜尋代數為 300 代，搜尋時間共 343.76 秒，GA 收斂過程如下圖 4.3，求得的重排程結果如表 4.54，重排程解之目標函數值為 0.8866。

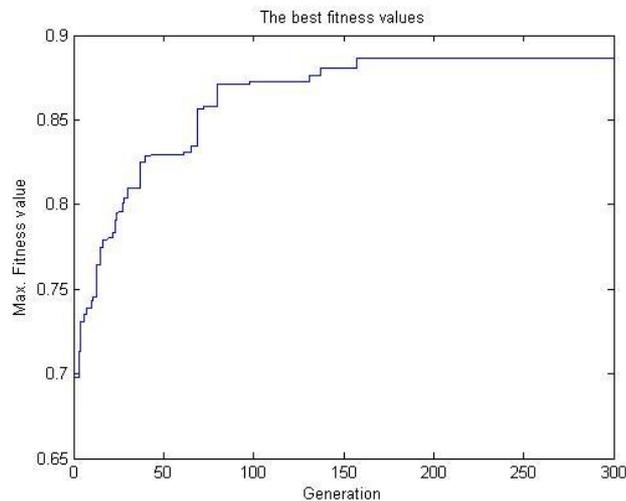


圖 4.3 緊急插單 GA 演算過程

由於現實環境的生產步調可能非常快，管能者可能被迫需要在短時間內就得到重排程結果。雖然以 HGA 可求得較好的排程結果，但鑑於時間的考量，以 GA 求解可在短時間內得到重排程結果，避免因為重排程計算時間過長，而影響整體訂單的作業時程，但若管理者無計算時間的考量，亦

可以 HGA 求得較佳的排程解，其重排的步驟方法仍同上述。

下表 4.51 為此重排程解由 GA 取得的作業排程優先順序，由此表計算出重排程後訂單平均順序如表 4.52。由表 4.52 可知，重排時該筆緊急插單的平均排程順序為 2，相較於表 4.50 定性因素評估結果所顯示的訂單優先順序 3，可知作業排程優先順序並非完全主導於定性因素的評估與懲罰函數的限制。以此例分析，由於該筆緊急插單的交期非常緊迫，而交期評估又佔目標函數值非常高的比例，因此經由 GA 求解時，在緊迫的交期限制下，可能為了提升平均交期滿足度，該筆插單的排序結果須更為優先，以此達成較高的整體目標函數值。

表 4.51 插單重排作業順序

作業	1-6	2-3	2-4	2-5	2-6	3-3	3-4	3-5	4-4
順序	13	7	9	14	29	5	18	23	3
作業	4-5	4-6	5-4	6-5	6-6	7-3	7-4	7-5	8-3
順序	6	11	16	17	21	25	27	28	2
作業	8-4	8-5	8-6	9-5	9-6	10-5	插單-1	插單-2	插單-3
順序	10	24	26	8	19	15	1	4	12
作業	插單-4	插單-5							
順序	20	22							

表 4.52 插單重排訂單平均順序

訂單編號	1	2	3	4	5	6	7	8	9	10	插單
訂單基因 順序總和	13	59	46	20	16	38	80	62	27	15	59
訂單基因 平均順序	13	14.75	15.33	6.67	16	19	26.67	15.5	13.5	15	11.8
排序	3	5	7	1	9	10	11	8	4	6	2

此外，由重排後的訂單交期滿足度觀察(表 4.53)，可知重排後該筆插單交期滿足度為 1，顯示其可準時達交；而訂單 4、5、6 則在整體目標函數的考量下犧牲了其個別的交期滿足度，但訂單 3、7、10 重排後交期滿足度反而上升。綜合上述，重排程以 GA 重新演化求解，求出之解可能會與前排程解有所差異，雖然重排後某些訂單的交期滿足度可能下降，但在考量整體目標函數下，犧牲某些較不重要的訂單，對於工廠整體利益卻可能是

大幅的提升。

表 4.53 重排後訂單交期滿足度

訂單編號	1	2	3	4	5	6	7	8	9	10	插單
原前排程 交期滿足度	1	1	0.95	1	1	0.97	0.49	1	1	0.45	
重排後 交期滿足度	1	1	1	0.49	0.92	0.96	1	1	1	0.69	1

此外，我們比較重排程前後的作業異動狀況，此次重排程需重排作業的總數為 29，重排後作業異動的總數為 27，以此例觀察，由於此緊急插單案例的重排是以 GA 進行重新演化求解，因此大多數作業的批量分割數和機台指派值皆與前排程相異，造成了重排後大多數作業異動的結果。

表 4.54 機器之作業排程(緊急插單重排結果)

工作中心	機器 編號	加工順序	加工作業 (訂單-作業)	整備 開始時間	作業 開始時間	作業 結束時間
1	1	1	9-2	10.5	10.8	20.3
		2	2-3	32.4	32.7	37.367
		3	1-4	39	39.7	44.7
		4	7-3	49.5	50.2	61.2
		5	4-6	96.2	96.7	112.7
	2	1	6-1	0	3.2	20.2
		2	1-4	39	39.7	44.7
		3	插單-1	50	50.6	59.1
		4	2-3	59.1	59.4	64.067
	3	1	5-1	0	0.3	15.3
		2	9-2	15.3	15.6	25.1
		3	1-4	39	39.7	44.7
		4	插單-1	50	50.6	59.1
		5	2-3	59.1	59.4	64.067
		6	7-3	64.067	64.767	75.767
2	1	1	9-1	0	1.5	10.5
		2	7-1	10.5	10.8	18.3
		3	3-2	22.9	25.1	34.1
		4	6-3	37.9	38.4	46.9
		5	10-4	48.4	48.9	52.9
		6	2-4	64.067	64.567	80.567
		7	8-4	80.567	82.367	90.867

表 4.54 機器之作業排程(緊急插單重排結果)

	2	8	插單-4	90.867	91.667	96.667
		1	7-1	0	0.3	7.8
		2	3-2	22.9	25.1	34.1
		3	6-3	37.9	38.4	46.9
		4	10-4	48.4	48.9	52.9
		5	4-4	52.9	56.3	75.3
		6	8-4	76.9	78.7	87.2
		7	插單-4	87.2	88	93
3	1	1	1-1	0	1	12
		2	3-1	12	12.4	22.9
		3	5-2	22.9	23.3	42.3
		4	2-2	42.3	42.7	51.2
		5	9-6	80.9	84.1	95.1
		6	8-5	95.1	96.6	103.6
	2	1	1-1	0	1	12
		2	3-1	12	12.4	22.9
		3	2-2	23.5	23.9	32.4
		4	10-3	32.4	33.4	48.4
		5	插單-3	66.6	67.8	79.8
		6	9-6	80.9	84.1	95.1
		7	8-5	95.1	96.6	103.6
	4	1	1	8-1	0	3.5
2			1-2	17.5	19.3	27.3
3			6-2	27.3	28.9	37.9
4			4-3	37.9	40.9	50.9
5			3-3	50.9	54.9	61.9
6			9-5	61.9	65.9	80.9
5	1	1	4-2	6.5	6.8	12.467
		2	10-2	13.4	14.6	25.6
		3	9-4	46.4	48.4	51.9
		4	5-4	52.1	52.3	58.3
		5	插單-2	59.1	59.6	66.6
		6	6-5	66.6	66.9	73.233
		7	3-4	73.233	73.733	79.067
		8	7-5	83.167	83.367	94.867
	2	1	4-2	6.5	6.8	12.467
		2	插單-2	59.1	59.6	66.6

表 4.54 機器之作業排程(緊急插單重排結果)

		3	5-4	66.6	66.8	72.8	
		4	6-5	72.8	73.1	79.433	
		5	2-5	80.567	81.567	98.567	
		6	3-4	98.567	99.067	104.4	
	3	1	4-2	6.5	6.8	12.467	
		2	9-4	46.4	48.4	51.9	
		3	插單-2	59.1	59.6	66.6	
		4	5-4	66.6	66.8	72.8	
		5	6-5	72.8	73.1	79.433	
		6	3-4	79.433	79.933	85.267	
		7	7-5	85.267	85.467	96.967	
	6	1	1	4-1	0	1	6.5
			2	1-5	44.7	47.5	60.5
3			10-5	60.5	61.1	71.1	
4			插單-5	93	93.5	101.5	
2		1	4-1	0	1	6.5	
		2	8-2	17.5	18.5	33.5	
		3	10-5	52.9	53.5	63.5	
		4	6-6	79.433	82.433	94.433	
		5	插單-5	96.667	97.167	105.17	
7		1	1	1-3	27.3	28.5	39
	2		7-2	39	39.5	49.5	
	3		5-3	49.5	49.8	59.3	
	4		8-3	59.3	59.9	76.9	
	5		4-5	76.9	78.2	96.2	
	2	1	1-3	27.3	28.5	39	
		2	5-3	42.3	42.6	52.1	
		3	2-6	98.567	98.867	108.37	
	3	1	9-3	25.1	26.4	46.4	
		2	7-2	46.4	46.9	56.9	
		3	2-6	98.567	98.867	108.37	
	8	1	1	10-1	0	0.4	13.4
2			1-6	60.5	61	71	
3			7-4	71	71.4	78.4	
4			3-5	85.267	87.767	99.767	
5			8-6	103.6	104.4	115.4	
2		1	2-1	0	0.5	23.5	

表 4.54 機器之作業排程(緊急插單重排結果)

		2	6-4	46.9	48	60
		3	7-4	75.767	76.167	83.167
		4	3-5	104.4	106.9	118.9
	3	1	6-4	46.9	48	60
		2	1-6	60.5	61	71
		3	7-4	75.767	76.167	83.167
		4	8-6	103.6	104.4	115.4

## 4.5 穩健排程結果

在說明完本系統的一般排程與重排程結果後，本節說明在求解過程中加入測試性干擾機制，求得穩健排程解的步驟結果。

實際生產環境中，中斷事件頻繁發生，造成排程頻繁性大幅更改，作業時程與原先規劃差異大，使得許多作業的事前準備無法及時達成，生產線績效下降。本系統的穩健排程解即是在防止上述因中斷重排造成的混亂與績效下降的問題，其於實際生產環境的功能意義，在於提供一種具有預防性功能性的排程，當穩健排程因為中斷事件而重排後，此重排後的排程結果仍然可維持高生產績效，保持現場作業的穩定性。

在穩健排程解的步驟結果說明中，依照本系統三種不同的中斷事件分為機台故障、物料短缺的穩健排程與緊急插單的穩健排程，每種穩健排程對其相對應的中斷事件提供預防性的功能，分別說明如下。

### 4.5.1 機台故障情境的穩健排程

說明機台故障的穩健排程時，我們假設該工廠於故障機台的情境如下。

#### 故障機台情境：

假設工作中心 1 的編號 3 機台老舊，缺乏維修，時常發生當機的情況，為了避免其當機影響生產績效與作業穩定性，管理人員希望所排出的排程能夠具穩健性，使此機台當機後對於作業績效、時程與現場穩定性的影響能降到最低。由該機台歷史資料的評估，所得該機台的當機可能狀況如下表 4.55。

表 4.55 工作中心 1-編號 3 機台 歷史故障狀況

工作中心 1-編號 3 機台	
機台故障時間點	符合均勻分配
機台修復時間	符合指數分配 $f(t)=1/12 \cdot \exp^{-t/12}$
機台平均修復時間	12

穩健排程解的求解過程大致上與本系統的一般排程解相同，唯一差異在於穩健排程解求解時，於禁忌搜尋法結束後再加上測試性干擾，以上述的機台故障情況來中斷其排程並重排，以評估該個體解對於某特定中斷情境的穩健性與穩定性。

在穩健排程解的目標函數計算中，以排程中斷並重排後的目標函數值來衡量穩健性，以正規化後的作業時間總差異來衡量穩定性，兩項指標之定義已於 3.4 節說明，將此兩項指標值與一般排程下的原目標函數值做加權後，即是求解穩健排程時的新整體目標函數值。各指標值的權重與求解過程的相關參數如下表 4.56，其中由於嘗試干擾的機台故障情境仍然有隨機的成份在，因此我們對每個干擾對象干擾多次，以求得較客觀的平均指標值。此干擾次數、各指標值權重、演算代數等為系統參數，可視求解品質而自行訂定。

表 4.56 穩健排程搜尋參數(機台故障情境)

參數設定		
演算法總個體數	20	
干擾對象	取一般排程下目標函數值較高的前 5 個	
干擾次數	每個干擾對象皆干擾重排 5 次	
目標函數各指標權重	原目標函數值	0.45
	平均穩健性指標值	0.35
	平均穩定性指標值	0.2
演算法求解代數	300	

結合 HGA 演算法與「故障機台情境」的干擾與重排後，以上表 4.56 的參數進行求解，搜尋過程如圖 4.4，搜尋時間約為 4891 秒。表 4.57 為找到最適穩健解之代數中 5 個干擾對象的各項指標值，其中干擾對象 5 的原目標函數值雖然非最高，但是其穩健性與穩定性指標較高，因此整體目標函數值表現較好，此干擾對象 5 即是本次搜尋得到的最適穩健排程解。此次搜尋過程中，找到的最小作業時間總差異為 0。

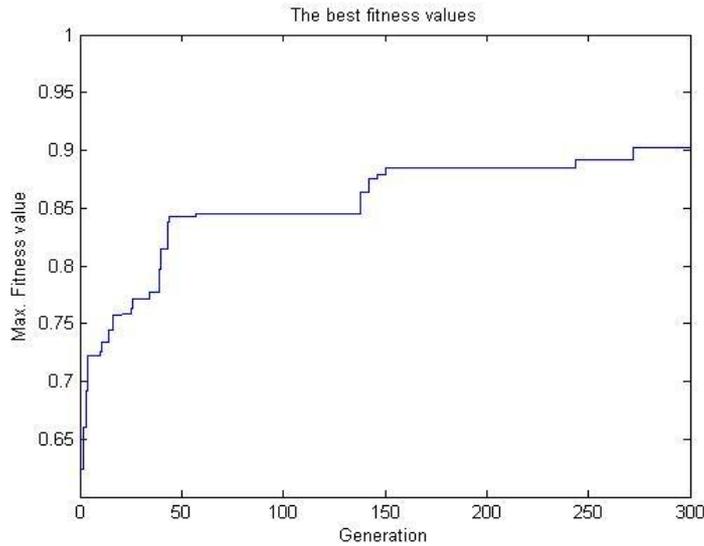


圖 4.4 HGA 穩健排程適應函數值趨勢圖(機台故障情境)

表 4.57 干擾對象各項指標值(機台故障情境)

	干擾對象 1	干擾對象 2	干擾對象 3	干擾對象 4	干擾對象 5
原目標函數值	0.9053	0.9052	0.9052	0.9052	0.9051
穩健性指標值	0.8846	0.8834	0.9016	0.9047	0.9054
穩定性指標值	0.6691	0.6040	0.6872	0.8094	0.8900
整體目標函數值	0.8508	0.8373	0.8603	0.8859	<b>0.9022</b>

表 4.58 穩健排程結果(機台故障情境)

最適穩健排程解	
原目標函數值	0.9051
平均穩健性指標值	0.9054
平均穩定性指標值	0.8900
整體目標函數值	0.9022

在上述演算法搜尋中，每次的 300 代搜尋，我們皆可同時找出原目標函數值最高的個體，與穩健排程考量下整體目標函數值最高的個體，兩者分別代表一般排程解與穩健排程解，兩者的求解過程大致相同，差別只在於評估的目標函數不同，表 4.59 即為此次 300 代搜尋中所得到的最適一般排程解結果，相較於穩健排程解結果表 4.58，此一般排程解的原目標函數值較穩健排程解的原目標函數值來得稍微高一些。

表 4.59 最適一般排程解結果

最適一般排程解	
原目標函數值	0.9057

## 穩健排程驗證

本小節以兩組排程解進行比較，分別為一般排程解與穩健排程解，以說明機台故障情境下，穩健排程解是否真能發揮功效。在比較驗證時，我們以表 4.59 的一般排程結果與表 4.58 的穩健排程結果所代表的排程來進行比較，兩組排程皆經歷相同的中斷事件(表 4.55)，比較兩組排程中斷並重排後的穩健性與穩定性。

實驗中，每組排程皆測試 5 次中斷並重排的動作，比較結果如下表 4.60。

表 4.60 一般排程解 vs 穩健排程解(機台故障情境)

實驗組次	一般排程解		穩健排程解	
	穩健性指標值	穩定性指標值	穩健性指標值	穩定性指標值
1	0.8836	0.5118	0.9054	0.9497
2	0.8897	0.6061	0.9054	0.7502
3	0.8923	0.6452	0.9051	0.6897
4	0.7938	0.2131	0.9053	0.7143
5	0.8885	0.5882	0.9054	0.7918
平均值	<b>0.8649</b>	<b>0.5129</b>	<b>0.9053</b>	<b>0.7791</b>
標準差	<b>0.0425</b>	<b>0.1745</b>	<b>0.0001</b>	<b>0.1028</b>

由上表 4.60 的分析，可看出穩健排程解歷經機台當機並重排後，穩健性指標值平均較一般排程解來得高，標準差也較一般排程解來得低許多，顯示此穩健排程經歷機台當機後，仍可維持高目標函數值，現實上的意義即是維持高生產績效；由穩定性指標來看，亦可看出穩健排程解歷經機台當機並重排後，穩定性指標值平均較高，標準差也比一般排程解來得低，顯示此穩健排程若經歷機台故障的情況，仍可表現出高穩定性，現實上的意義即是保持現場作業的穩定性，避免重新規劃排程造成的作業混亂或後勤作業的準備不及。

#### 4.5.2 物料短缺情境的穩健排程

本小節說明物料短缺情境的穩健排程，假設該工廠於物料短缺的情境如下。

##### 物料短缺情境：

假設目前所需生產的訂單產品中，訂單 2 的作業 3 所需用料為特殊料件，料件編號為 TE31J，可能因供應商歷史資料顯示出準時達交率非常低，或是供應商無法確定是否可於交期內準時達交，為了避免其缺料影響生產績效與作業穩定性，管理人員希望所排出的排程能夠具穩健性，使此缺料情況對於作業績效、時程與現場穩定性的影響能降到最低。由該物料達交與供應商歷史資料的評估，所得物料短缺可能情況如下表 4.61。

表 4.61 物料短缺可能狀況

料件編號	TE31J
平均達交率	32.5%
平均的最晚遲交時間	最晚可能遲交 12 單位時間

結合 HGA 演算法與「物料短缺情境」的干擾與重排後，以下表 4.62 的參數進行求解。

表 4.62 穩健排程搜尋參數(物料短缺情境)

參數設定		
演算法總個體數	20	
干擾對象	取一般排程下目標函數值較高的前 5 個	
干擾次數	每個干擾對象皆干擾重排 5 次	
目標函數各指標權重	原目標函數值	0.4
	平均穩健性指標值	0.4
	平均穩定性指標值	0.2
演算法求解代數	300	

本例搜尋過程如圖 4.5，搜尋時間約為 4746 秒。表 4.63 為找到最適穩健解之代數中 5 個干擾對象的各項指標值。其中干擾對象 4 的原目標函數值雖然非最高，但是其穩健性與穩定性指標較高，因此整體目標函數值表現較好，此干擾對象 4 即是本次搜尋得到的最適穩健排程解。此次搜尋過程中，找到的最小作業時間總差異為 40.57。

表 4.63 干擾對象各項指標值(物料短缺情境)

	干擾對象 1	干擾對象 2	干擾對象 3	干擾對象 4	干擾對象 5
原目標函數值	0.8529	0.8529	0.8521	0.8520	0.8520
穩健性指標值	0.7570	0.8149	0.8261	0.8260	0.8083
穩定性指標值	0.6446	0.7652	0.9102	0.9107	0.8252
整體目標函數值	0.7729	0.8202	0.8533	0.8534	0.8292

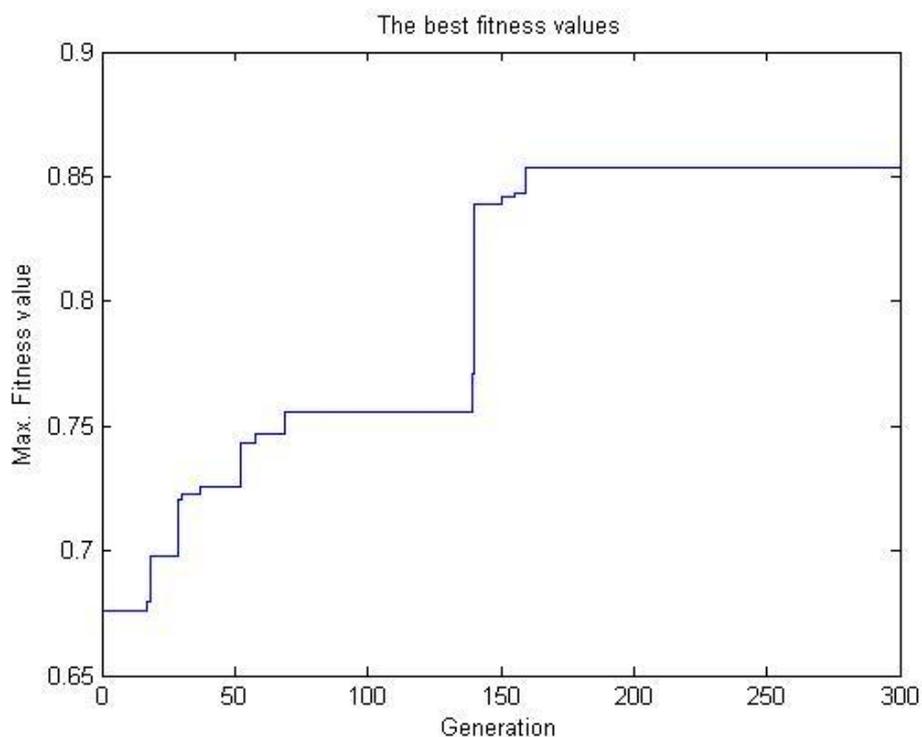


圖 4.5 HGA 穩健排程適應函數值趨勢圖(物料短缺情境)

在上述演算法搜尋中，求得之穩健排程解與一般排程解結果分別為表 4.64 與表 4.65。

表 4.64 穩健排程結果(物料短缺情境)

最適解	
原目標函數值	0.8520
平均穩健性指標值	0.8260
平均穩定性指標值	0.9107
整體目標函數值	0.8534

表 4.65 最適一般排程解結果

最適一般排程解	
原目標函數值	0.8551

## 穩健排程驗證

本小節以兩組排程解進行比較，分別為一般排程解與穩健排程解，以說明物料短缺情境下，穩健排程解是否真能發揮功效。在比較驗證時，我們以表 4.65 的一般排程結果與表 4.64 的穩健排程結果所代表的排程來進行比較，兩組排程皆經歷相同的中斷事件(表 4.61)，比較兩組排程中斷並重排後的穩健性與穩定性。

實驗中，每組排程皆測試 5 次中斷並重排的動作，比較結果如下表 4.66。

表 4.66 一般排程解 vs 穩健排程解(物料短缺情境)

實驗組次	一般排程解		穩健排程解	
	穩健性指標值	穩定性指標值	穩健性指標值	穩定性指標值
1	0.8261	0.5773	0.8292	0.9542
2	0.7287	0.4372	0.8574	0.9150
3	0.8235	0.5277	0.8291	0.9542
4	0.8647	0.6412	0.8136	0.7368
5	0.7287	0.4373	0.8618	0.9150
平均值	<b>0.8108</b>	<b>0.5241</b>	<b>0.8382</b>	<b>0.8950</b>
標準差	<b>0.0621</b>	<b>0.0889</b>	<b>0.0206</b>	<b>0.0906</b>

由上表 4.66 觀察，可看出穩健排程解歷經缺料的干擾並重排後，穩健性指標值平均較一般排程解來得高，標準差也較一般排程解來得低，顯示此穩健排程經歷機台當機後，仍可維持高目標函數值，於實際生產中的意義即是維持高生產績效；由穩定性指標值來看，亦可看出穩健排程解歷經缺料的干擾並重排後，仍可表現出高穩定性，於實際生產中的意義即是保持現場作業的穩定性，避免重新規劃排程造成的作業混亂或後勤作業的準備不及。

### 4.5.3 緊急插單情境的穩健排程

本小節說明緊急插單情境的穩健排程，假設該工廠於緊急插單的情境如下。

#### 緊急插單情境：

緊急插單為製造現場中常見的干擾因素之一。假設本例 10 筆訂單的生產時程內，極可能發生一次緊急插單的干擾。由於現場產能有限，管理者可自訂緊急插單承接政策或可接受的範圍。本例假設得知緊急插單的時間點非常接近下個排程週期時，或是所需生產的數量太龐大，無法於本週期內生產完畢，則可協議移至下個排程週期。緊急插單可承接範圍與預設資料如下表 4.67。

表 4.67 緊急插單可承接範圍(可能插單情況)

得知該插單的時間點	低於 50 則插入本週期生產 超過 50 則移至下個排程週期
該插單需生產的數量	可小於 200 單位
途程數	可小於 6 站
各途程作業的總加工時間	預設於 8~25 單位時間
各作業整備時間	預設於 0.3~3 單位時間
各作業上機批量限制	預設於 15~200
交期	預設於得知插單時間點後的 40~60 單位時間

以表 4.67 緊急插單可承接範圍作為求解穩健排程時的緊急插單干擾範圍，結合 HGA 演算法與此「緊急插單情境」的干擾與重排後，以下表 4.68 的參數進行求解。其中由於緊急插單的干擾重排是以 GA 演算法求解，因此整體穩健排程的搜尋時間倍增許多，為了降低搜尋時間，在此僅對每個干擾對象干擾重排 2 次，但不免稍微降低了求解品質與客觀性。

表 4.68 穩健排程搜尋參數(緊急插單情境)

參數設定		
演算法總個體數	20	
干擾對象	取一般排程下目標函數值較高的前 5 個	
干擾次數	每個干擾對象皆干擾重排 2 次	
目標函數各指標權重	原目標函數值	0.4
	平均穩健性指標值	0.4
	平均穩定性指標值	0.2
演算法求解代數	300	

本例搜尋過程如圖 4.6，搜尋時間約為 158650 秒。表 4.69 為找到最適穩健解之代數中 5 個干擾對象的各項指標值。表中干擾對象 4 即是本次搜尋得到的最適穩健排程解。此次搜尋過程中，找到的最小作業時間總差異為 133.72。

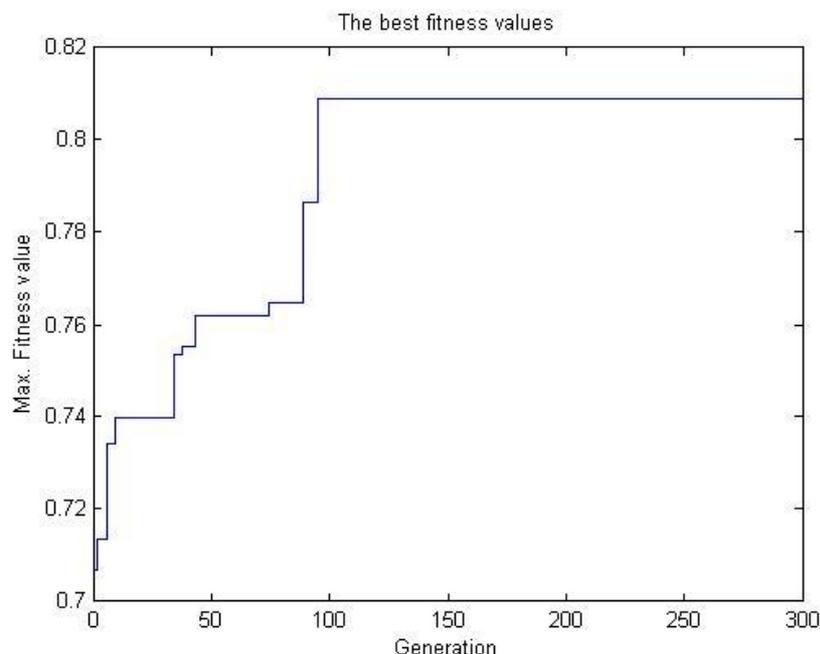


圖 4.6 HGA 穩健排程適應函數值趨勢圖(緊急插單情境)

表 4.69 干擾對象各項指標值(緊急插單情境)

	干擾對象 1	干擾對象 2	干擾對象 3	干擾對象 4	干擾對象 5
原目標函數值	0.8788	0.8718	0.8635	0.8588	0.8583
穩健性指標值	0.6910	0.7910	0.8010	0.7710	0.7610
穩定性指標值	0.5064	0.4164	0.5464	0.7864	0.7264
整體目標函數值	0.7292	0.7484	0.7751	0.8092	0.7930

表 4.70 穩健排程結果(緊急插單情境)

最適穩健解	
原目標函數值	0.8588
平均穩健性指標值	0.7710
平均穩定性指標值	0.7864
整體目標函數值	0.8092

表 4.71 最適一般排程解結果

最適一般排程解	
原目標函數值	0.8738

## 穩健排程驗證

本小節以兩組排程解進行比較，分別為一般排程解與穩健排程解，以說明緊急插單情境下，穩健排程解是否真能發揮功效。在比較驗證時，我們以表 4.71 的一般排程結果與表 4.70 的穩健排程結果所代表的排程來進行比較，兩組排程皆經歷相同的緊急插單事件(如下表 4.72)，比較兩組排程中斷並重排後的穩健性與穩定性。

表 4.72 緊急插單例子

緊急訂單資料	
得知緊急訂單的時間	50
交期	98.5
總途程數	6
作業途程	1,5,3,2,6,8
數量	200
各作業總加工時間	17,21,12,10,16,18
各作業整備時間	0.6, 0.5, 1.2, 0.8, 0.5, 0.7
各作業上機批量限制	48,42,134,69,72,150

實驗中，每組排程皆測試 5 次中斷並重排的動作，比較結果如下表 4.73。

表 4.73 一般排程解 vs 穩健排程解(緊急插單情境)

實驗組次	一般排程解		穩健排程解	
	穩健性指標值	穩定性指標值	穩健性指標值	穩定性指標值
1	0.8191	0.4492	0.8046	0.5525
2	0.8081	0.5678	0.8120	0.5855
3	0.8061	0.4440	0.8070	0.5089
4	0.7645	0.4295	0.8238	0.5437
5	0.7958	0.4583	0.8118	0.6122
平均值	<b>0.7995</b>	<b>0.4698</b>	<b>0.8118</b>	<b>0.5606</b>
標準差	<b>0.0208</b>	<b>0.0558</b>	<b>0.0074</b>	<b>0.0397</b>

由表 4.73 的比較結果分析，可看出穩健排程解歷經緊急插單的干擾並重排後，穩健性指標值平均較一般排程解來得稍高，標準差則較一般排程解來得低許多；由穩定性指標值來看，亦可看出穩健排程解歷經緊急插單的干擾並重排後，穩定性指標值雖然只維持在 0.5~0.6 左右，但是比起一般排程解，已高出了約 0.1，顯示此穩健排程的穩定性要比一般排程來得好。

### 小節結論：

由機台故障、物料短缺與緊急插單三種情境的穩健排程來觀察，各中斷情境下的穩健排程對於其中斷事件，皆表現了較高的穩健性與穩定性，顯示演算法中加入測試性干擾機制，確實可求解出某種特定情境下的穩健排程。該穩健排程經歷排程中斷事件並重排後，所得到的目標函數值較一般排程來得高，顯示穩健排程可吸收較多的負面干擾，表現出較高的穩健性(Robustness)。此外，該穩健排程在重排之後的穩定性(Stability)表現，也比一般排程來得高，顯示出穩健排程經歷重排規劃後，各項作業的起訖時間不會與前排程相差太大，對於現場作業的穩定性維持與生產後勤作業的準備，可有較大的幫助。

分析上述三種情境的穩健排程結果，由於緊急插單的重排程求解方式是利用 GA 進行完全重排法，而非機台故障或缺料情況使用的部分修正重排法，因此不論原本前排程的作業排序如何，藉由 GA 重新演化求解的機制，大多數的作業順序可能皆與原前排程不同，以表 4.73 結果來看，穩定性指標都只維持在 0.5~0.6，顯示以 GA 重排對於排程穩定性的維持似乎較無幫助。相對的，以機台故障與物料短缺的驗證結果來看(表 4.60 與表 4.66)，部分修正重排法對於穩定性的維持似乎幫助較大。然而，若要以較客觀的方法比較，仍需要以部份重排法來重排緊急插單的中斷事件，才可比較出兩方法間的差異。

## 第五章 結論與未來研究方向

### 5.1 結論

本研究於零工式工廠平行機台生產環境下，發展重排程方法與穩健排程的求解，其中考量零工式平行機台型態下的作業批量分割與機台指派問題，並以多目標架構結合排程量性因素(交期、製距、機器使用率)、質性因素(公司策略、市場考量、顧客歷史交易等)、排程穩健性指標、排程穩定性指標等。本研究探討的排程中斷事件包含機台當機、物料短缺與緊急插單等三項，分別發展建立此三項排程中斷事件的重排程方法。重排程中，對於機台當機與物料短缺的情況，以部份重排法來重新規劃，以維持前排程基因值的方式修正重排，由於不需要經過演算法多代的求解，不需搜尋時間，可快速重新規劃出一個排程解。對於緊急插單的情況，以完全重排法進行重排，求得較好品質的最適重排程解，其中包括加入緊急訂單後重新評估的定性因素修正。此外，研究中以測試性干擾機制結合混合式遺傳演算法，於演算求解過程中事先預估個體解的穩健性與穩定性，以求出一穩健排程解，由穩健排程的驗證可知，結合測試性干擾機制於演算法中，確實可找出同時具穩健性與穩定性的排程解，降低排程因中斷事件產生的負面影響，提高排程對於中斷干擾的容忍程度。在動態生產環境中，此穩健排程結果對於生產績效的維持可有很大的幫助。

根據實驗結果，本研究歸納下列幾項結論：

1. 部份修正重排法，主要是將原前排程之基因值修正後，再重新代入解碼機制，由於不需要經過演算法多代的求解，不需搜尋時間，可快速重新規劃出一個排程解，此法對於生產步調快速的工廠環境可有較大的幫助。
2. 物料短缺情況的重排程，對於缺料作業的分割批量數決定，可藉由 GA 做短暫的搜尋求解，以找出對整體目標函數值幫助最大的作業分割批量數與機台指派值，而非一味地給予延遲作業更多的機台資源，造成其他訂單的績效損失。
3. 緊急插單情況以完全重排法來進行重排，雖然此法需經過演算法的演化求解，但可求得品質較好的排程解。此外，緊急訂單的情況先以 AHP 方法重新評估整體訂單的相對重要性，再以演算法求得整體目標函數值較

好的最適解，可避免一味地將緊急訂單排入第一優先，造成其他重要訂單的績效降低。

4. 結合測試性干擾機制於演算法中，確實可找出同時具穩健性與穩定性的排程解，降低排程因中斷事件產生的負面影響，提高排程對於中斷干擾的容忍程度，並且降低排程因重排規劃造成的穩定性下降影響。在動態生產環境中，此穩健排程對於生產績效的維持可有很大的幫助。

## 5.2 未來研究方向

雖然本研究經由實驗驗證，確實可達預期之效果，但在發展本研究架構與問題實作的過程中，也發現一些可進一步探討的問題，歸納出幾項未來研究方向，如下：

1. 研究中以零工式工廠做為驗證應用之範圍，建議在後續研究可以嘗試將本研究之重排程系統架構擴展至更複雜的排程問題上，如彈性製造系統之排程問題。
2. 本研究中，各中斷事件皆單以部份重排法或完全重排法進行重排，未來應可對於每種中斷事件以兩種重排程方法同時進行分析比較，以歸納出部分重排法與完全重排法分別的優劣處與最適當的應用時機。
3. 後續研究應可利用不同的演算法來加快搜尋速度與求解品質，提升排程解的搜尋品質與效率。
4. 本研究中，每一次的重排程皆只針對單一個中斷事件的發生進行重排，未來應可研究針對多個中斷事件的重排程方法，例如：同時解決機台當機與缺料事件的重排程方法。
5. 研究中，機台當機的情況，並未探討機器日常維修與可靠度的問題，建議在後續研究可嘗試加入探討。

## 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]吳鴻輝、李榮貴，生產排程之重排程決策問題的文獻探討與分析，*Journal of the Chinese Institute of Industrial Engineers*，第14卷第2期，147-158，1997。
- [3]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。
- [4]林暘桂，「不相關平行機器總加權延遲時間最小化之排程問題」，朝陽科技大學，碩士論文，2001。
- [5]林虹谷，「製造業即時排程與重排程基礎架構」，國立高雄第一科技大學資訊管理系，碩士論文，2002。
- [6]黃俊嘉，「多廠整合型生產指派與排程系統之最佳化」，東海大學工業工程研究所，碩士論文，2005。
- [7]Adenso, D., “An SA/TA mixture algorithm for the scheduling tardiness problem”, *European Journal of Operational Research*, Vol. 88, pp516-524, 1996.
- [8]Akturk, M. Selim, and Elif Gorgulu, “Match-up scheduling under a machine breakdown”, *European Journal of Operational Research*, 112, pp.81-97, 1999.
- [9]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.
- [10]Azizoglu, M. and Alagoz, O., “Parallel-machine rescheduling with machine disruptions”, *IIE Transactions*, 37, 1113–1118, 2005.
- [11]Artigues, C., Billaut, J.-C. and Esswein, C., “Maximization of solution flexibility for robust shop scheduling”, *European Journal of Operational Research*, 165, 314-328, 2005.
- [12]Baker, K. R., *Introduction to Sequencing and Scheduling* (John Wiley & Sons, New York, NY 1974).
- [13]Baker, R.K., “Sequence rules and due-date assignments in a job shop”, *Management Science*, 30, pp.1093-1104, 1984.
- [14]Brah, S.A., Hunsucker, and J.L., “Branch and Bound Algorithm for The Flowshop with Multiple Processors,” *European Journal of Operational Research*, 51, pp. 88-99, 1991.
- [15]Bean, J.C., “Genetic Algorithms and Random Keys for Sequencing and Optimization”, *ORSA Journal of Computing*, Vol. 6, No. 2, 1994.
- [16]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.
- [17]Byeon, E.-S., Wu, S.D., Storer, R.H., “Decomposition Heuristics for Robust Job-Shop Scheduling”, *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, 1998.

- [18]Bierwirth, C. and Dirk, C. Mattfeld, "Production scheduling and rescheduling with genetic algorithms", *Evolutionary Computation*, Vol. 7, No. 1, pp1-17, 1999.
- [19]Cheng, T. C .E., Diamond, J. E., "Scheduling Two Job Classes on Parallel Machines", *IIE Transactions*, Vol. 27, pp689-693, 1995.
- [20]Cheng, R., Gen, M. and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms-1 representation", *Computers Ind. Engng.*, Vol. 30, No. 4, pp983-997, 1996.
- [21]Cheng, R. and Gen, M., "Parallel machine scheduling problems using mimetic algorithms", *Computer and Industrial Engineering*, Vol. 33, No. 3-4, pp761-764, 1997.
- [22]Chou, F.D. and Lee, C.E., "Two-machine flowshop scheduling with bicriteria problem", *Computers & Industrial Engineering*, 36, pp.549-564, 1999.
- [23]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*, Vol. 14, No. 4, July/August, 2001, p 367-384.
- [24]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.
- [25]Dooley, J.K. and Mahmoodi, F., "Identification of Robust Scheduling Heuristics: Application of Taguchi Methods in Simulation Studies", *Computers ind. Engng*, Vol. 22, No. 4, pp.359-368, 1992.
- [26]Daniels, R.L., "Incorporating performance information into multi-objective scheduling", *European Journal of Operational Research*, 77, pp.272-286, 1994.
- [27]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.
- [28]Forgaty, T. C., "Varying the probability of mutation in the genetic algorithm", *Proceedings of the 3rd International Conference on Genetic algorithms*, pp104-109, 1989.
- [29]Fang, H. L., "Genetic Algorithm in timetabling and scheduling", Ph. D. dissertation, Department of Artificial Intelligent, University of Edinburgh, 1994.
- [30]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.
- [31]Garey M.R., and Johnson D.S., "Computers and Intractability: A Guide to the Theory of NP-completeness," Freeman, San Francisco, 1979.
- [32]Gupta, J.N.D., "Two-Stage, Hybrid Flowshop Scheduling Problem," *J. Oper. Res. Soc.*, 39, pp.359-364, 1988.
- [33]Goldberg, D.E., "*Genetic Algorithm in Search, Optimization and Machine Learning*", Addison-Wesley Publishing Co., 1989.
- [34]Glover, F., "Tabu search-part I", *ORSA Journal on Computing*, Vol. 1, No. 3, pp190-206,

1989.

- [35]Glover, F., "Tabu search-part II", *ORSA Journal on Computing*, Vol. 2, No. 1, pp4-32, 1990.
- [36]Glover, F., "Tabu search : a tutorial ", *Interface*, Vol. 20, No. 4, pp74-94, 1990.
- [37]Guo, Bo and Nonaka Yasuo, "Rescheduling and optimization of schedules considering machine failures", *International Journal of Production Economics*,60-61, pp.503-513, 1990.
- [38]Guinet, A., " Scheduling independent jobs on uniform parallel machines to minimize tardiness criteria,"*Journal of Intelligent Manufacturing*, Vol. 6, No. 2, pp. 95-103, 1995.
- [39]Gürsel, A. S., Pico, F. and Santiago, A.,"Identical machine scheduling to minimize the number of tardy jobs when lot-splitting is allowed,"*Computer and Industrial Engineering*, Vol. 33, No. 1-2, pp. 277-280, 1997.
- [40]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.
- [41]Hundal, T.S., Rajgopal, J., "An Extension of Palmer's Heuristic for the Flow-shop Scheduling Problem," *International Journal of Production Research*, 26 (6), pp. 1119-1124, 1988.
- [42]Herrmann, J.W., "A Genetic Algorithm for minimax Optimization Problem", Proceedings of the 1999 Congress on Evolutionary Computation, pp.1099-1103, Washington, D.C., 1999.
- [43]Hu, Y.H, Yan, J.Q., Ye, F.F. and Yu, J.H., "Flow shop rescheduling problem under rush orders", *Journal of Zhejiang University SCIENCE*, 6A(10), 1040-1046, 2005.
- [44]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.
- [45]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.
- [46]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.
- [47]Jacques Carlier, Ismaïl Rebaï, "Two Branch and Bound Algorithms for the Permutation Flow Shop Problem," *European Journal of Operational Research*, 90, pp. 238-251, 1996.
- [48]Jensen, T.M., "Generating Robust and Flexible Job Shop Schedules Using Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 3, June 2003.
- [49]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.
- [50]Kumar, N.S.H. and Srinivasan, G., "A genetic algorithm for job shop scheduling- A case

- study”, *Computers in Industry*, 31 ,155-160, 1996.
- [51]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.
- [52]Leon, V.J., Wu, S.D., Storer, R.H., “Robustness Measures and robust scheduling for job shops, *IEE Trans*, 26(5), 32-43, 1994.
- [53]Liaw, C.-F., “A hybrid genetic algorithm for the open shop scheduling problem”, *European Journal of Operational Research*, 124, 28-42, 2000.
- [54]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.
- [55]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.
- [56]Mellor, P., “A review of job shop scheduling ”, *Operational Research Quarterly*, Vol. 17, No. 2, pp161-170, 1966.
- [57]Michalewicz, Z., *Genetic Algorithm + Data Structures = Evolution Programs*, Springer-Verlag Berlin Heidelberg, 1994.
- [58]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.
- [59]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.
- [60]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.
- [61]Min, L. and Cheng, W., “A genetic algorithm for minimizing the makespan in the case of scheduling identical machines,”*Artificial Intelligence in Engineering*, Vol. 13, pp. 399-403, 1999.
- [62]Mehta, S.V. and Uzsoy, R.M., “Predictable Scheduling of Job Shop Subjects to Breakdowns”, *IEEE Transactions on Robotics and Automation*, 14, pp.365-378, 1998.
- [63]Nawaz, M., “A Heuristic Algorithm for the ‘M’-Machine, ‘N’ Job Flow-Shop Sequencing Problem,” *Management Science*, 11(1), pp. 91-95, 1983.
- [64]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.
- [65]Nowicki E., and Czeslaw S., “The Flow Shop with Parallel Machines : A Tabu Search Approach,” *European Journal of Operational Research*, 106, pp. 226-253, 1998.
- [66]Norman, B.A., Bean, J.C., “A genetic algorithm methodology for complex scheduling

- problems”, *Naval Research Logistics* , 46 , 199-211, 1999.
- [67]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.
- [68]Pinedo, M., *Scheduling : theory, algorithms, and systems*, *Prentice Hall, Inc.* , pp118-141, 1995.
- [69]Piersma, N., and Van Dijk, W., “A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search”, *Mathematics and Computer Modeling*, Vol.24, No.9, pp11-19, 1996.
- [70]Park, B.J., Choi, H.R. and Kim, H.S., “A hybrid genetic algorithm for the job shop scheduling problems”, *Computers & Industrial Engineering*, 45, 597–613, 2003.
- [71]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.
- [72]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.
- [73]Rangsaritratamee, R., Ferrel Jr. W.G., Kurz, M.B., “Dynamic rescheduling that simultaneously considers efficiency and stability”, *Computers & Industrial Engineering*, 46, pp.1-15, 2004.
- [74]Sridhar, J., Rajendran, C., “Scheduling in A Cellular Manufacturing System: A Simulated Annealing Approach,” *International Journal of Production Research*, 31(12), pp. 2927-2945, 1993.
- [75]Santos, D.L., Hunsucker J.L., and Deal. D.E., “Flowmult: Permutation Sequences for Flow Shops with Multiple Processors,” *J. Inform. Optim. Sci.*, 16, pp. 351-366, 1995.
- [76]Suresh, V. and Chaudhuri, D., “Bicriteria scheduling problem for unrelated parallel machines,” *Computer and Industrial Engineering*, Vol. 30, No. 1, pp. 77-82, 1996.
- [77]Serafini P., “Scheduling jobs on several machines with the job splitting property,” *Operations Research*, Vol. 44, No. 4, pp. 617-628, 1996.
- [78]Srivastava, B., “An effective heuristic for minimizing makespan on unrelated parallel machines,” *Journal of the Operational Research Society*, Vol. 49, pp. 886-894, 1998.
- [79]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.
- [80]Shafaei, R. and Brunn, P., “Workshop scheduling using practical (inaccurate) data- Part2: An Investigation of the Robustness of the Scheduling Rules in a Dynamic and Stochastic Environment”, *International Journal of Production Research*, Vol. 37, No. 18, pp.1405-4117, 1999.
- [81]Sevaux, M., Sörensen, K., “A genetic algorithm for robust schedules”, Paper presented at

the 8th International Workshop on Project Management and Scheduling, Valencia, April 3–5, 2002.

- [82]Sörensen, K., “Tabu searching for robust solutions”, research paper, Faculty of Applied Economic Sciences UFSIA-RUCA, University of Antwerp, UA, 2002.
- [83]Tamimi, S. A. and Rajan, V. N., “Reduction of total weighted tardiness on uniform machines with sequence dependent setups,”6th Industrial Engineering Research Conference Proceedings, pp. 181-185, 1996.
- [84]Velagapudi, K.N., “Robust Schedule for Manufacturing Systems”, *Computers and Industrial Engineering*, Vol. 23, pp.133-136, 1992.
- [85]Vieira et al., “Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Method”, *Journal of Scheduling*, 6, pp.39-62, 2003.
- [86]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.
- [87]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.
- [88]Wu, S.D., Byeon, E.-S. and Storer, R.H., “A Graph-theoretic Decomposition of the Job Shop Scheduling Problem to Achieve Scheduling Robustness”, *Operations Research*, Vol. 47, No. 1, pp.113-124, 1999.