東　　海　　大　　學

工業工程與經營資訊學系

碩士論文

多瓶頸與瓶頸漂移於限制驅導式排程之研究

研 究 生：張碩淵

指導教授：張炳騰 教授

曾宗瑤 教授

中華民國一〇五年六月

A Research of Multi-Bottlenecks and Bottleneck Shifting on Drum-Buffer-Rope Job Shop Scheduling

By

Shuo-Yuan Chang

Advisors：Prof. Ping-Teng Chang

Prof. Tsueng-Yao Tseng

A Thesis

Submitted to the Institute of Industrial Engineering and Enterprise Information at Tunghai University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering and Enterprise Information

June 2016

Taichung , Taiwan

多瓶頸與瓶頸漂移於限制驅導式零工式之研究

學生：張碩淵　　指導教授：張炳騰教授

曾宗瑤教授

東海大學工業工程與經營資訊學系

摘　　要

實際的生產規劃中，瓶頸代表著所有工作站中產能負荷量最大的一站，在現場生產環境中常會有產能負荷與瓶頸站非常接近的工作站，由於產能負荷非常接近的關係，常會因生產環境的變動，使產能負荷接近的工作站產生轉變，並出現與最大產能負荷工作站同等負荷的工作站，或是替代原本瓶頸站成為最大負荷所在，造成瓶頸所在位置的漂移至新瓶頸位至，使該工作站儼然成為不可忽視一限制所在，此生產環境就同時具有「多瓶頸」與「瓶頸漂移」特性。

本研究以零工式生產排程規劃為基礎，提出三種訂單類型(頭尾型、中間型、相鄰型)與兩種瓶頸排程方法(前瓶頸為主，後瓶頸為輔、前瓶頸為輔，後瓶頸為主)，搭配基因演算法與限制驅導式，探討同時具有多瓶頸與瓶頸漂的環境，提出一排程規劃方法使瓶頸站與瓶頸站間的產能規劃得以最大化利用。經本研究系統證實，相較於以前瓶頸為主，後瓶頸為主之方法，在三種訂單類型中皆可求得較佳之排程結果。

關鍵字： 零工式生產、多瓶頸、瓶頸漂移、限制理論、限制驅導式

A Research of Multi-Bottlenecks and Bottleneck Shifting on Drun-Buffer-Rope Job Shop Scheduling

Student：Shuo-Yuan Chang

Advisors：Prof. Ping-Teng Chang
Prof. Tsueng-Yao Tseng

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

In real production planning, bottlenecks are one of the largest capacity loading workcenters. Usually, there are workcenters which may bear capacity loads that are very close to the bottleneck in real production environment. Due to the close capacity loads and often that the production environment may change, these workcenters may change along, and workcenters with the same largest capacity load may appear. Or it may replace the original bottleneck and becomes the new bottleneck. This results in the location of bottleneck shifting, and inevitably making the workcenters a new constraint that cannot be ignored. This production environment thus has both “multi-bottlenecks” and “bottleneck shifting” characteristics.

In this research, based on job shop scheduling planning, it proposes three types of orders and two methods of scheduling. Using Drum-Buffer-Rope with Genetic Algorithm, the research discusses a production environment that has both multi-bottlenecks and bottleneck shifting. It proposes a production scheduling method that makes the capacity utilization planning between bottlenecks which can be maximized.

Keywords：Job-Shop, Multi-Bottleneck, Bottleneck Shifting, Theory of Constraints (TOC), Drum-Buffer-Rope (DBR)

誌謝

大學四年加上研究所兩年的東海生活稍縱即逝，收穫卻也相當龐大，首先先要感謝的是我的父母，他們能我在求學期間無經濟壓力的後顧之憂，使我能全心專注於學術研究中，同時也感謝我的指導教授張炳騰老師，謝謝老師在研究所兩年對我的指導，當我遇上研究瓶頸時，適時的給予我方向與建議，也感謝曾宗堯老師、白炳豐老師、洪國禎老師、時序時老師抽空審閱我的論文，給予了我許多改善的建議，使我的論文能更趨於完善。

在研究的兩年間，感謝共同奮鬥努力的柏威、佳明、予容，使我在研究的過程中不會感到孤單，也感謝已經畢業的鄭佳佺學長，正值在職忙碌階段，卻時常抽空回來與我討論研究的內容，與常碰見的盲點，使我在做研究的過程中少走許多彎路，亦感謝碩一與碩零的學弟妹：志豪、庭愉、維綸、柏健、孟哲、可清、瑞陽的背後全力支援，使我可以心無旁騖地做研究，研究所兩年這一路走來，對於不時幫我的你們，致上萬分的感謝。

張碩淵 謹誌於

東海大學工研工程與經營資訊學系研究所

中華民國一０五年六月

目錄

表目錄

圖目錄

第一章 緒論

1.1 研究背景與動機

現今市場上的競爭愈加激烈，企業在競爭市場中占有一隅須仰賴其本身的競爭力以及應變能力，而排程規劃則是生產製造應變能力所在。透過對於排程規劃的嚴密控管，使所制定的排程規劃達到整體最佳績效。

一般從事生產製造的企業絕對脫離不了要設計生產排程，一個好的生產排程能讓生產快速上軌，降低產品交期與縮短流程時間。而「瓶頸」則是從事生產排程計畫時首要控制的問題，一般在生產規劃中都是以派工法則做為排程方法，不考慮瓶頸對於整體生產所造成的影響，因此 Goldratt, E. M.博士於 1980 年提出了限制理論，考慮了瓶頸對於生產規劃的影響，透過瓶頸的管理使生產排程更加穩定。

由於絕大多數對於瓶頸的管理生產規劃，都是找出生產規劃中所有作業裡影響最大的工作站，做為該生產規劃中唯一的限制所在，但作為限制所在的瓶頸工作站將不會是唯一的，而會呈現相同或是極為接近的存在與最大限制的瓶頸站並立，稱為「多瓶頸」生產環境，此時所產生出新限制瓶頸對於整體生產規劃的影響等同於原先的限制瓶頸，若是以新限制瓶頸做為之後排程規劃的唯一目標，排程規劃結果也會因舊有的限制瓶頸影響，使其無法達到最佳與排程規劃的穩定；同理推之，若是棄限制瓶頸，持續以就有限制瓶頸當作唯一目標，其結果亦然。

因此對於多瓶頸生產環境需考慮到一同進行生產規劃時對於排程之影響，Goldratt(1980)的限制理論中提到瓶頸對於整體生產是有影響，因此當並立之瓶頸站出現時，其之間的相互影響不可忽視，所以在對於並立瓶頸站的規劃亦考慮相互間的影響，而在排程中並立瓶頸站相互間必定有所犧牲，透過使目標限制的瓶頸站不會再是唯一的，讓生產規劃能有更周全的考量。

另外，實際生產環境中也會有許多原因造成生產規劃的不穩定與遲滯，可能會造成生產規劃中最大負荷瓶頸站的移轉或是不明確，並超越原本限制成為新的最大限制所在，使原本限制資源成為並立或是非限制資源，此一現象統稱為「瓶頸漂移」。

傳統上生產規劃對於瓶頸漂移的影響做法，將其以想放置於生產規劃

最後，透過增加排程上的作業時間，使排程甘特圖因其加入而整體作業時間向右移動，做為解決瓶頸漂移問題的手段，但是此一方法卻未考慮到瓶頸漂移後，新瓶頸站對於整體生產規劃的影響，或是成為並立之瓶頸，使整體排程績效不佳的結果，因此須針對此問題的一同放入生產規劃中討論，而非將其影響強加於排程結果中。

1.2 研究目的

根據本研究之研究背景與動機所述，整理歸納現有研究對於瓶頸問題如下：

1. 現行研究大多著重於只對單一瓶頸問題執行生產排程規劃，忽視並立瓶頸產生時的影響。
2. 並立瓶頸出現時，瓶頸間排程時所產生之影響，乃至於對整體排程規劃之影響。
3. 當生產環境中出現造成新並行瓶頸的原因時，傳統排程於完成後執行甘特圖中執行右移方法，未考慮新舊瓶頸間的相互影響。
4. 當生產環境中出現造成瓶頸漂移的原因，並新瓶頸負荷超越舊有瓶頸時，傳統排程將執行以新瓶頸之重排程，未考慮新舊瓶頸間的相互影響。
5. 現行運用 DBR 研究大多僅針對單一瓶頸解決問題，無法有效處理並立瓶頸所造成之瓶頸間之矛盾，乃至於整體生產規劃之影響。

本研究將探討零工式排程問題，並針對上述之瓶頸問題，提出一改善限制驅導式之模型方法，解決並立瓶頸與瓶頸漂移產生時問題，制定對於瓶頸站間於排程上出現之矛盾時，瓶頸間相互取捨之方法，並結合基因演算法使求得排程規劃結果最佳化。

1.3 研究範圍與限制

實際環境中會依照不同的生產型態與環境，產生數種不同複雜程度的排程規畫，本研究中將針對零工式(Job Shop)生產環境下，產生出多瓶頸與瓶頸漂移的排程問題進行探討，並根據本研究目的與工具的選擇，研究範圍與假設如下：

1. 針對零工式生產環境探討。
2. 探討多瓶頸與瓶頸漂移產生時，其瓶頸間衝突與生產規劃之影響。
3. 假設排程開始執行時，所有訂單接已備妥。
4. 假設皆無在製品產生。
5. 假設排程過程中皆無不良品產生
6. 不考慮人為因素影響。
7. 不考慮外包。
8. 排程過程中無迴流。

1.4 研究架構

本研究論文共分為五個章節做探討，第一章闡述本研究之背景、動機與目標，並給定研究之假設極限制等內容；第二章根據本研究之內容所提及之理論與方法做相關文獻整理並探討，包含限制理論、限制驅導式、多瓶頸、瓶頸漂移與基因演算法相關文獻；第三章則是依據前兩章所歸納整理，提出一解決多瓶頸與瓶頸漂移問題之排程求解方法；第四章則依照第三章之方法、步驟，並提出不同類型之訂單問題做系統驗證，針對驗證結果提出分析，第五章為本研究之結論，歸納並說明前四章所得之結果。本研究架構流程圖如下頁圖 1.1。

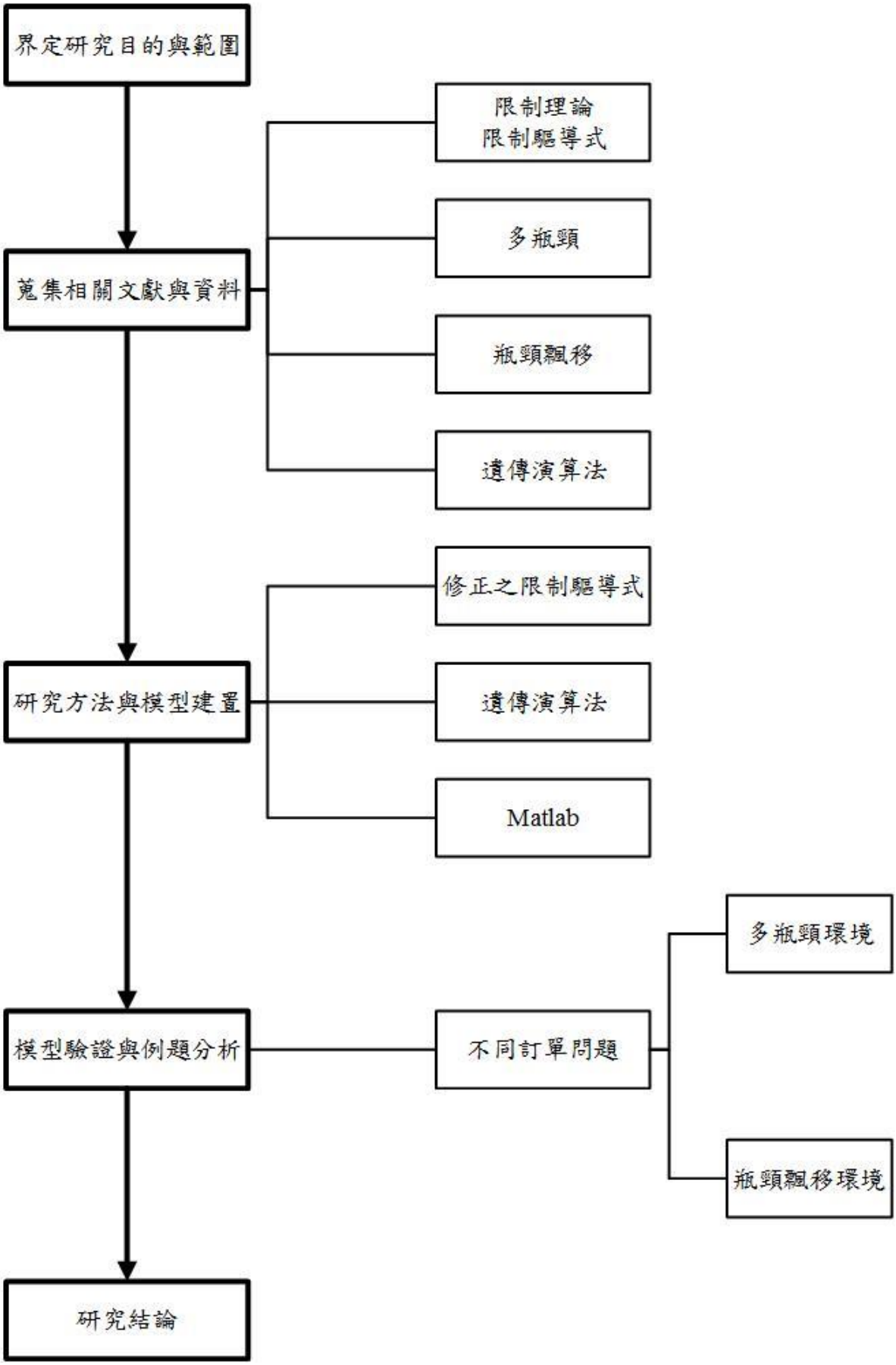
界定研究目的與範圍
蒐集相關文獻與資料
限制理論
限制驅導式
多瓶頸
瓶頸飄移
遺傳演算法
研究方法與模型建置
修正之限制驅導式
遺傳演算法
Matlab
模型驗證與例題分析
不同訂單問題
多瓶頸環境
瓶頸飄移環境
研究結論

圖 1.1 研究架構流程圖

第二章　文獻探討

2.1 排程問題描述

Pinedo 與 Chao (1999) 提出生產排程規劃是將有限資源分配的決策過程，常於生產製造中的採購、生產、加工、銷售中所使用，並將訂單或是生產規劃轉換成生產活動。Leung (2004) 的研究定義中，排程計劃是確立全部加工機器的工作順序和開始時間，因此排程問題是一最佳化問題須同時考慮限制資源與執行限制。擁有一個完善的排程規劃可使企業的資源利用最佳化，並且減少時間與金錢上的成本花費。

2.1.1 排程問題分類

Pinedo 與 Chao(1999)歸納提出排程問題會因生產現場的機器數目與配合的工單數量劃分，可以分成數種不同複雜程度的問題：

1. 單機（Single-machine）問題：每一張訂單都只會經過一個加工程序，並且只有一部加工機台。
2. 平行機台（Parallel-maghine）問題：每一張訂單都只會經過一個加工程序，並且有多部相同功能之加工機台。平行機台又可分成等效平行機台與非等效平機台。等效平行機台指的同站加工機台效能一樣；非等效平行機台是同站加工績效能不一。
3. 流程式生產（Flow Shop）問題：每一張訂單都有複數個加工程序，並且每一張訂單都有相同之加工順序。
4. 零工式生產（Job Shop）問題：每一張訂單都有複數個加工程序，而且每一張工單都有各自的加工順序。
5. 開放式生產（Open Shop）問題：每一張訂單都有複數個加工程序，但是每一張工單並無一定的加工順順序。

2.1.2 零工式排程問題

零工式生產為目前中實際生產方式中最普遍的，其特色在於相同功能之機器設備會共同組成一工作中心，訂單於不同工作中心流動，並根據其訂單特性與工作中心功能，執行不同的加工程序，最大特色在於每張訂單都有固定之加工流程，且所經過之途程不盡相同。零工式排程問題為本研究所探討目標，大多的排程問題都為最佳化問題，NP-hard 整理如下：

1. 由 n 張訂單與 m 台機器所組成。
2. 每張訂單接有負數個作業，所需之加工途程機台與順序部完全相同，加工順序需依照途程單上的順序依序加工。
3. 每張訂單皆有交期限制。
4. 訂單中每項作業階有各自加工時間。
5. 加工機台產能皆有限。

2.1.3 零工式生產排程求解方法

由於零工式生產於實際生產中為較普遍的一種方式，於近代也有不少學者針對此類問題進行研究，針對此類問題的解方法，本研究歸納出下列幾類：

1. 派工法則(Dispatching rule)：為最常見之解決方法，具有容易理解與求解快速之優點，但對於整體之排程績效較為不佳。常見之派工法則有最短作業時間法(Shortest Process Time, SPT)、先到先服務(First Come First Served, FCFS)、最常作業時間(Longest Processing Time, LPT)、最早到期日(Earliest Job Due Date, EDD)。
2. 最佳化法(Optimization method)：利用作業研究(Operations Research, OR)的方式是求解，可求得問題之最佳解，但會隨著求解問題的擴大，造成求解時間程指數成長拉長，也常因簡化問題使用過多限制與假設，於實務上不推薦使用，常使用之方法有線性規劃法(Linear Programming)、分枝界限法(Branch and Bound)、整數規劃法(Integer Programming)。
3. 啟發式演算法(Heuristic Algorithm)：因為本身具有分割ㄧ克服(Divide-and-Conquer)及重複改善(Iterative Improvement)的特點，可以在較短的時間內得到一個近似最佳解，於求解龐大排程問題時，可節省所需消耗

之時間與成本，因此實際環境中常使用啟發是演算法求解，常見的啟發式演算法有系統模擬法(Simulated Annealing)、塔布搜尋法(Tabu Search)、基因演算法(Genetic Algorithm)。

4. 人工智慧(Artificial Intelligence)：具有近似於人類的判斷能力與推理技巧，可決定出有效的搜尋，同時備有即時性的解題能力。

2.1.4 零工式生產排程績效指標

完成一完整生產排程規劃後，須有目標做為衡量標準用以判定此排程結果之優劣，其中 Baker (1984) 提出做為排程績效的目標大致可分為時間績效(Shop Time Performance)與交期滿足績效(Due Date Performance)兩類，而依照排程的績效可分類為常見下列數種：

1. 總完工時間(total completion time)：每張訂單最後完成之時間稱之為完工時間，所有訂單之完工時間總合稱為總完工時間。
2. 最大完工時間(make span)：生產排程完成後，於最一項後離開加工機器的作業之時間為最大完工時間。
3. 平均流程時間(mean flow time)：一張訂單從開始作業至完成，中間所經過之時間稱為流程時間，所有訂單流程時間的平均值為平均流程時間。
4. 平均差異時間(mean lateness)：訂單完工時間與交期之間差值的絕對值，而所有訂單之平均則為平均差異時間。

2.2 限制理論與限制驅導式

自限制理論提出至今，已廣泛的運用於不同領域中，對於組織管理與生產規劃提出全新的不同思維，針對其最弱之部分持續提供改善之方法，提升整體之績效。

根據 2.1.4 所提出數種衡量排程規劃之績效目標，透過限制理論找出系統中產能負荷最大之所在，並針對瓶頸資源嚴密恐控管，使瓶頸站之利用率達到最佳化，方可確保整體績效最佳，因此急需一方法，可針對瓶頸站做出處理，而限制驅導式(Drum-Buffer-Rope, DBR)概念正好與之契合，利用生產規劃中 Drum 的監控，與其他資源的全力配合下，使生產排程規劃能穩定。

2.2.1 限制理論

限制理論是 Goldratt 博士在 1980 年代所提出的一種管理思想方法，他認為任何生產形式或是組織都會有自己應達成的目的，會影響到其績效或是目的的就是限制，在從事生產的過程中就被稱為「瓶頸」。限制理論主要目的是在於找出生產系統時的瓶頸，並且管理瓶頸使生產穩定，讓系統的產出最佳化。

Goldratt 博士提出限制理論中的持續改善程序，稱之為五個專注步驟：

1. 指出系統的限制：生產系統中整體績效會受到限制資源影響，因此在改善前要先確認瓶頸資源發生處，並且謹慎監控確保可以持續改善。
2. 充分應用系統的限制：透過第一步找出瓶頸資源後，制定依計畫將瓶頸資源的利用最大化，以便使整體績效達到最佳。
3. 非限制資源站配合限制站：保護瓶頸資源站的可持續運作相當重要，因此須由非瓶頸站配合瓶頸站的生產計畫為優先考量。
4. 打破系統的限制：生產計畫設計者須提高瓶頸站的稼動率，藉以獲得更佳的效率，透過提高瓶頸站的稼動率，有機會使原先的瓶頸站消失，而有新的瓶頸站產生。
5. 尋找新的限制，回歸步驟一，別讓惰性成為限制，持續不斷地改善：限制理論所追求的是不斷的改善資源中的限制，藉以提高整體活動的績效，因此新的瓶頸產生，即回歸步驟一持續地改善整體活動。

2.2.2 限制驅導式

限制驅導式排程是以限制理論為基礎，所提出的生產規劃排程方法。在進行生產計畫時，優先以系統限制或是瓶頸站資源需求為導向，充分利用限制資源與非瓶頸資源的全力配合，分配在有限資源下排程技術。

「鼓」是使用限制驅導式的開端，同時鼓也是產系統中限制的所在，因此鼓也控制著生產計畫的產能限制;「緩衝」即在瓶頸資源站前設置庫存區，其目的主要是確保生產線不會因受限之產能出現停工之狀況;「繩」指將瓶頸站資源狀況傳遞給上游站，目的主要是在於生產能按照一定節奏，以及避免庫存的產生。限制驅導排程步驟：

1. 找出系統限制所在:限制驅導式排程主要將生產系統中資源區分為瓶頸資源與非瓶頸資源二種，因此在排成的過程中須先找出瓶頸資源。
2. 決定緩衝區之大小：在基於保護瓶頸資源站為前提下，在瓶頸站前設置一緩衝區，用來確保生產系統不會因受限制資源而發生停工事件。
3. 設計限制驅導節奏：先不考慮瓶頸站之產能，並以生產計畫排定一生產排程之理想甘特圖，圖內必定會有「廢墟」產生，代表著此一排程並不可行,因此須酌授進行廢墟之推平,使整體的生產排程計畫合理並可行。
4. 決定訂單開始時間：在找出系統限制與決定緩衝區，即應使用「繩」從緩衝區時程計算反推訂單開始之時間。

2.3 瓶頸問題

2.3.1 多瓶頸問題

廖建閎 (2005) 的研究發現多瓶頸與瓶頸飄移產生的原因歸類在市場端的產品組合發生變動，該研究將 MTO 生產模式為優先考量，當有剩餘產能產生時則是使用 MTS 生產將其填補使產線平穩，並特別指出 MTO 是直接導致多瓶頸與瓶頸漂移。

李偉榮 (1999) 的研究提出一套多瓶頸產能規劃演算法，並且運用兩階段產能規劃法，進行生產線的產能規劃活動。

1. 第一節段：考慮起使初步產能限制與各站產能上限，排入符合訂單。
2. 第二階段：第一階段排程結束後，考慮產能上線並規劃剩餘訂單，即使用產能堆疊方式排入產能規劃中，無法排入之訂單則推延至下次排程中在排入。

該研究指出透過該方法求得平均產出、使用率、標準差，和以往產能規劃比較，可有較佳績效。

2.3.2 瓶頸飄移

「TOC 限制理論」一書提到，造成瓶頸漂移現象原因為大批量批次化生產，同時瓶頸漂移也意味著該工廠沒有瓶頸，或解釋成瓶頸發生處為市場，由市場的需求來決定工場生產排程時的瓶頸站位置。所採行的解決方法是透過小批量生產，藉以縮小緩衝時間，非瓶頸人員也會因整備換線次數增加，以致於不會有太多閒置。

Lawrence 與 Buss (2009) 認為在生產系統中會有很多隨機性事件都會導致瓶頸站的漂移，如：訂單的開始時間、機器故障、工作站上人員的疏失，因此數量化並提出瓶頸漂移機率，使用每站變成瓶頸站的機率當作瓶頸漂移之機率，使管理者更易於管理。

沈妙妙等人 (2008)與徐學軍等人(2008)兩篇文獻共指出使用了限制理論來消除瓶頸漂移問題，沈妙妙等人(2008)指出藉由對於瓶頸站與非瓶頸站加工批量的彈性空來消除瓶頸漂移之問題；徐學軍等人(2008)則寫道每一站的加工批量皆為固定，藉由瓶頸站的加工時間加長來消除瓶頸漂移問題，另外更提出產生瓶頸漂移機率之數學模型來解決問題。並提出導致瓶頸漂

移之原因：

1. 待加工件抵達加工站之頻率：工件抵達加工站之頻率越頻繁，代表加工機器之利用率也就越高，某站之利用率的增加意味著會降低他站成為瓶頸之機率，而增加瓶頸漂移的機會。
2. 瓶頸站加工時間之長短：瓶頸站的加工時間若相較非瓶頸站時間短，則會增加瓶頸漂移機會。
3. 工作站規模大小：某工作站的加工機器數量越多，代表者在製品的規模也越大，機器利用率也就越高，瓶頸漂移的可能性也就增大。
4. 緩衝區的大小：緩衝區的容量越是大，代表相同加工時間下，機器的利用率也較越高，代表著更容易瓶頸漂移。

沈妙妙 (2011) 使用限制理論計算出各工作站產能的負荷量，並且強調須有完善之監控方法對瓶頸漂移有更好的控管，此外也提出產生瓶頸漂移之因素有「需求波動」與「設備故障」二種，搭配監控方法之模型來解決問題。Lawrence 與 Buss (2009)研究中提出造成瓶頸漂移之二種原因：

1. 需求波動：分類為生產系統之外在因素，市場需求波動引發的生產系統異動，主要原因是企業產品和生產決策所造成的，客戶訂單更動使產品組合的改變，意味著生產線上各站所負荷之產能與原先有差異，而產能的異動就代表著造成瓶頸漂移的可能性就大增。
2. 設備故障：分類為生產系統之內在因素，其內容包含設備故障，原物料之短缺，此時若是緩衝之保護不足，該站將會成為臨時瓶頸，造成瓶頸之漂移。

針對不同因素下的預防與應對：

1. 需求波動：
   (1) 若是波動不大，生產系統就相對穩定，根據限制理論的定義下，充分利用緩衝機制，加強緩衝區的緩衝保護，或是增加瓶頸站之前站的產能，確保瓶頸站有足夠產能應對突發狀況。
   (2) 若是波動較大，生產系統就相對不穩定，則要針對投料與排程計畫做更好之規劃，充分運用投料控制緩衝區大小，降低瓶頸漂移得發生，並保護瓶頸站資源的生產排程計畫順利進行。

2. 設備故障：

(1) 加強對加工設備之維修保養。

(2) 統計各站故障時的維修時間，設置合理的緩衝區與緩衝大小，保證在故障發生時有足夠的緩衝時間供排除故障，避免瓶頸漂移。

何文中與洪躍(2012)以及凌琳、劉明周、唐娟、趙志彪、葛茂根與蔣增強(2012)兩篇文獻提到，何文中與洪躍(2012)探討到絕大多數之瓶頸漂移皆為靜態模式，而此一篇文獻則是加入了時間因素並且給與量化，建構出動態之數學模型，求其有效控制瓶頸漂移問題之最佳解；凌琳等人(2012)則是結合了「限制理論」與「整體產出效能」兩方法來消除瓶頸漂移問題，限制理論提供解決瓶頸與平衡產線產能；整體產出效能提供生產線上的績效指標與瓶頸識別指標，最終提高生產線上之產能。

2.4 基因演算法

Holland 於 1975 年發表了基因演算法(Genetic algorithm, GA)此一搜尋機制，其概念源自於達爾文進化論中的「物競天擇，適者生存」，該演算法模擬自然界中的生物演化過程，隨著環境的改變不斷的自我改條是，於競爭過程中不斷重複著複製(Replacement)、交配(Crossover)與突變(Mutation)等方式，適應環境變化防止被淘汰。Michalewicz (1994) 歸納了基因演算法是由五個基本步驟所組成：

1. 以基因型態表是問題的特徵或解答：基因演算法對問題的表示方是依照不同的問題而有不同的表示，通常分別以一個或是一組基因表示，其中一基因為二元(binary)數字，如同一個體是由數個基因所組成，問題的解答是由不同的問題特徵所組成。
2. 創造任意數目的初始解答：於運作基因演算法開始之前，須先產生數組的初始解做為初始母體(initial population)。出使母體產生的方式則區分為隨機與特定兩類，Forgaty (1989) 指出出使母體產生之方法會影響演算法隻搜尋績效。
3. 評估工能的建立與適應函數(fitness function)的設計：若一個體的適應值越高，代表此一個體在環境中越容易從活下來，也同時越容易產生下一代。以適應函數作為結決問題時之目標函數，藉由適應函數判定

結果皆進預設目標之程度，若適應函數值越高代結果越接近目標，使基因演算法有更大的機會尋到最佳解答之個體。

4. 使用基因運算子(genetic operator)產生子代：複製(Replacement)、交配(Crossover)與突變(Mutation)為最常見之基因運算子。
   (1) 複製(Replacement)：決定個體是否可存活至下一代，透過「適者生存」的原則進行篩選，適應函數越高的個體有較高的機率被閃選存活至下一族群。
   (2) 交配(Crossover)：透過個體間的基因交換，產生一新的個體，增加新的搜尋空間。針對浮點實數編碼，Goldberg (1989) 以五種不同交配方式對於排序問題進行電腦模擬測試，以決定不同交配方是其績效之優劣。
   (3) 突變(Mutation)：藉由隨機改變各體內的基因的方式，產生一新的個體，增加新的搜尋空間。由於突變的發生為隨機性，使在求解過程中可以搜尋新的領域，避免限入局部最佳解(local optimal)之情況。Goldberg (1989) 以六種不同突變方式進行測試。
5. 參數設定：執行基因演算法的過程中，許多參數必須預先設定，如交配率、突變率、染色體長度與母族群數等，不同的參數設定對於基因演算的搜尋績效皆有影響。Forgaty (1989) 研究中曾對於不同的參數設定，對於基因演算的績效進行評估。

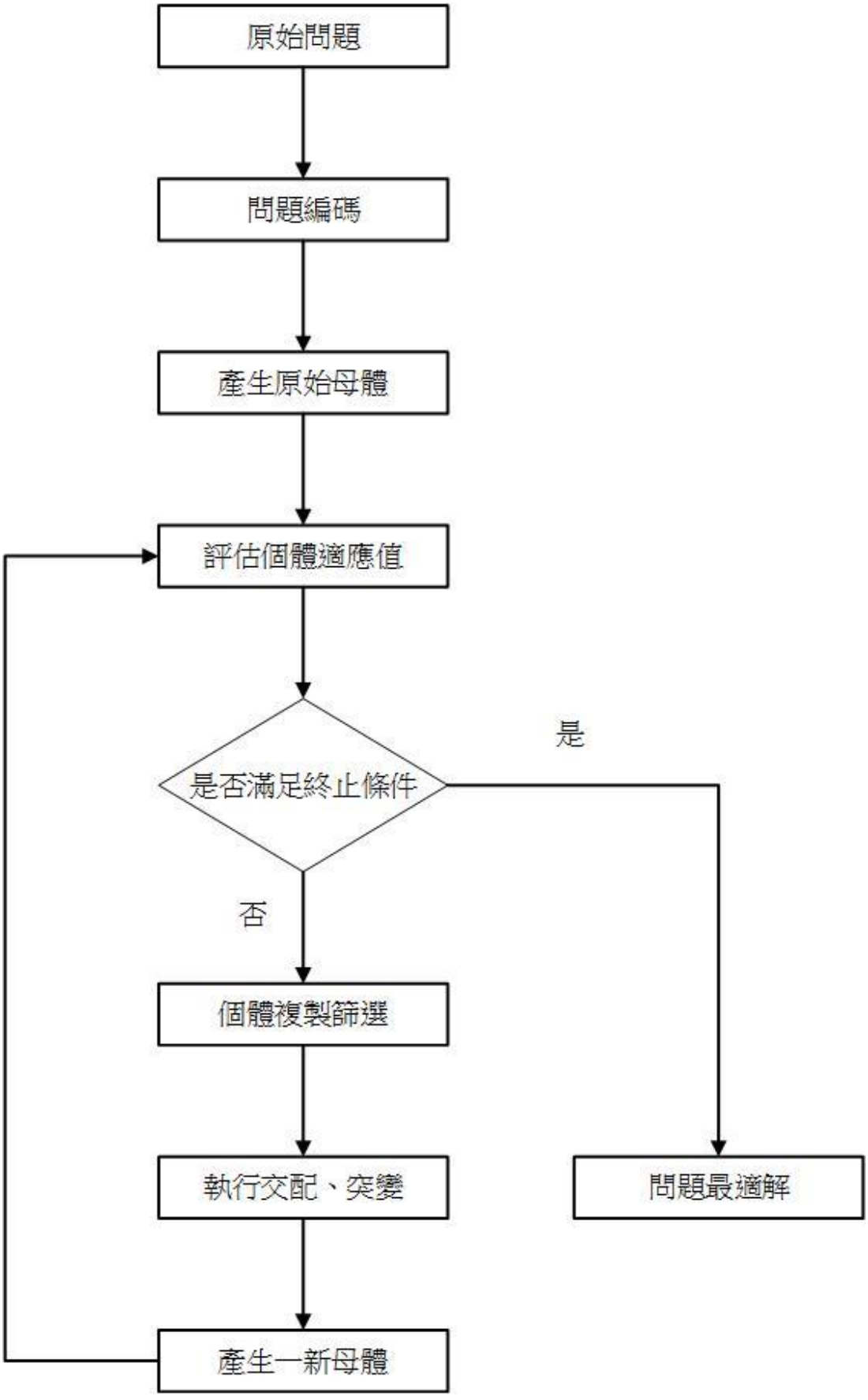

圖 2.1 基因演算法流程圖

2.4.1 編碼

編碼為基因演算法的第一步驟，編碼完成後即稱為染色體，染色體是由數個基因所組成，數個染色體則構成母體亦為一解集合。下為三種常見之編碼方式：

1. 二近位編碼

編碼方式藉由二進位之 0 與 1 所組成，圖 2.2 為一染色體，染色體由三個基因組成，每一個基因則由 2 個二位元來編碼。

0	1	1	1	0	1

圖 2.2 二位元編碼圖

2. 實數編碼

染色體內每一基因及對一實數變數，圖 2.3 為一染色體內含 3 個基因的實數編碼。

77.6	12.9	33.8

圖 2.3 實數編碼圖

3. 符號編碼

使用數字與符號的方式進行編碼，常用於求解排序問題中，以排程問題為例，因為訂單上的圖程順序有不可重複性，所以長以數字代表作業的圖程順序，染色體由左至右則為訂單的加工途程順序，圖 2.4 為長度為 6 的符號編碼圖，基因中 1 至 6 為加工途程順序。

1	5	2	4	3	6

圖 2.4 符號編碼圖

2.4.2 初始母體

基因演算法具有多點平行搜尋之特性，因此需產生一初始母體作為基因演算法之起始解，初始母體的大小亦會對於求解過程中的品質與效率有影響，初始母體態會造成基因演算法提早收斂，太大則會使求解時間過長，常用的產生出使母的方法有兩種：

1. 隨機產生：隨機產生可能導致系統求解時間過長，但不易使結果限入局部最佳解之中。
2. 啟發解產生：先行利用演算法則產生一初始母體，可增加系統的求解速度，但較易於限入局部最佳解中。

2.4.3 適應函數(fitness function)

一個體其適應值越高，代表此一個體在環境中越容易存活下來，並產生下一代個體。將較於適應函數做為解決問題的目標函數，藉由目標函數判斷所求解接近目標的程度，目標函數值越高則代表越所求解越接近目標，同時也代表越有機會使基因演算法搜尋到更好的目標。

2.4.4 複製(reproduction)

複製主要是針對母體篩選出合適的個體，過程中會依據是應值的大小決定個體是否會被複製並放置到交配池中，各體適應值較高者有較高機率被選中；反之，適應值越低之個體則越容易被淘汰。在複製的過程中，可以使適應值較佳之各體得以被保留下來，供後續的交配與突變的執行。常間的複製方式有以下兩種：

1. 競爭法(Tournament Selection)：自母體中挑選兩個或是兩個以上個體，比較其適應值大小，適應值大的個體則放入交配池中，重複比較與放的步驟至交配池內的個體數達到初始母體數量則終止
2. 輪盤法(Roulette Wheel Selection)：利用每一個體的適應值於母體中所佔比例，來設定各體於於輪盤中所戰之比例，適應值越高之個體於輪盤中所戰之比例則越高，複製過程中被篩選中的機率也越高，缺點在於，當有一個體其適應值遠高於其他染色體時，被重複篩選中的機率也就大增，於演化的途中所產生之個體差異也就相對較小，容易限區域最佳解中。

表 2.1 輪盤法執行步驟一

個體編號	1	2	3	4	5
適應值	27	39	26	34	51

表 2.2 輪盤法執行步驟二

適應值排序	1	2	3	5	4
個體編號	5	2	4	1	3
適應值	51	39	34	27	26

表 2.3 輪盤法執行步驟三

適應值排序	1	2	3	5	4
個體編號	5	2	4	1	3
適應值	51	39	34	27	26
個體積率	5/15	4/15	3/15	2/15	1/15

2.4.5 交配(Crossover)

基因演算法中，基因交配主要是為了讓個體相互交換有用的資訊，使個體有較較大的機會組合出更高適應值的個體，達到不斷演化的目的。

解決排序問題時，染色體是由許多不重複的數字做為基因所組成一字串，傳統中基因演算法所使用的運算子江蕙無法處理此類問題，傳統的運算子在搜尋上可能會使個體中產生數字相同的基因。為了不使排序問題執行交配時產生重複的情況，因此必須設計新的運算子，用以產生合理的子代。

針對浮點時數編碼與排序問題，Goldberg (1989) 研究中提出了五種常使用的交配方法，PMX(Partially Matched Crossover)、LOX(Linear Order Crossover)、SX(Simple Crossover)、RX(Random Crossover)、

CX(Cycle Crossover)，作為基因演算法運用的基準，並進行研究比較，且 Liaw (2000) 研究中，針對不同的交配運算子進行排成問題測試，根據其結果指出 LOX 於問題中可求得較佳之結果。

1. PMX(Partially Matched Crossover)：該交配運算子原先是針TSP(Traveling Salesman Problem)問題所發展的交配方法，PMX 特點在於著重保存各自員在母代染色體上的位置資訊，其運算步驟如下：

   (1) 隨機產生兩切點。

   個體 A：(1 2 3｜5 7 6｜9 8 4)

   個體 B：(9 5 4｜2 3 6｜8 7 1)

   (2) 將個體 A 與個體 B 在兩切點中的基因相互對調。

   個體 A：(1 2 3｜2 3 6｜9 8 4)

   個體 B：(9 5 4｜5 7 6｜8 7 1)

   (3) 將個體 A 在兩切點之外重複的基因與個體 B 於切點之外重複的基因對調。

   個體 A：(1 7 5｜2 3 6｜9 8 4)

   個體 B：(9 3 4｜5 7 6｜8 2 1)

2. LOX(Linear Order Crossover)：LOX 於 Goldberg (1989)研究中較為建議適用於排成問題的交配運算子，其運算步驟如下：

   (1) 隨機產生兩切點。

   個體 A：(1 2 3｜5 7 6｜9 8 4)

   個體 B：(9 5 4｜2 3 6｜8 7 1)

   (2) 將個體 A 切點中所有基因，於個體 B 中以*代替；個體 B 者亦然。

   個體 A：(1 * *｜5 7 *｜9 8 4)

   個體 B：(9 * 4｜2 3 *｜8 * 1)

   (3) 將*往切點中間移動，使兩切點中的基因皆為*。

   個體 A：(1 5 7｜* * *｜9 8 4)

   個體 B：(9 2 4｜* * *｜8 3 1)

(4) 將原本個體 A 與個體 B 切點中的基因互調。

個體 A：(1 5 7｜2 3 6｜9 8 4)

個體 B：(9 2 4｜5 7 6｜8 3 1)

3. SX(Simple Crossover)：該運算子常用於 TSP 與排成問題中，其運算步驟如下：

(1) 隨機產生一切點

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) 保留個體 A 切點左邊的基因，切點右邊的基因於個體 B 中順序填入；個體 B 者亦然。

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便，切點中的基因以隨機方式產生。

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) 於個體 A 中任選一基因，假設選到 9，其相對位置在個體 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) 於個體 A 中基因為 1，相對位置個體 B 中為 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) 於個體 A 中基因為 4，相對位置個體 B 中為 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) 於個體 A 中基因為 6，相對位置個體 B 中為 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) 重複上述步驟，直至標示回到 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 與個體 B 中，未被標示到的基因對調。

個體 A：(9 8 2 1 5 4 7 6 3)

個體 B：(1 2 3 4 7 6 5 8 9)

2.4.6 突變(Mutation)

達爾文的進化論一文中有提到，自然界除了倚靠交配來獲得較佳的子代來適應多變的自然環境，當母帶無法在位子代提供較佳的基因來適應環境時，則會有機率的會產生突變來為子代可以適應生存環境。Goldberg (1989)研究中，針對排序問題提出了六種常見的突變運算子，SM(Swap Mutation)、PBM(Position-based Mutation)、IM(Inverse Mutation)、TSM(Three Swap Mutation)、ASM(Adjustment Swap Mutation)、SHM(Shift Mutation)。

1. SM(Swap Mutation)：任選兩個基因，將其相互對調。

   (9 8 4 5 7 6 1 3 2)→(9 8 4 1 7 6 5 3 2)

2. PBM(Position-based Mutation)：任選兩個基因，假設為 5、1，將魚之後的 7、6 向前移動，再將 5 填入 6 原本的位置中。

   (9 8 4 5 7 6 1 3 2)→(9 8 4 7 6 5 1 3 2)

3. IM(Inverse Mutation)：任選兩切點，將切點內的基因反轉。

   (9 8 4｜5 7 6｜1 3 2)→(9 8 4｜6 7 5｜1 3 2)

4. TSM(Three Swap Mutation)：任選三個基因，使用隨機的方式交換。

   (9 8 4 5 7 6 1 3 2)→(9 7 4 5 3 6 1 8 2)

5. ASM(Adjustment Swap Mutation)：任選兩相鄰的基因，將其交換。

   (9 8 4 5 7 6 1 3 2)→(9 8 4 7 5 6 1 3 2)

6. SHM(Shift Mutation)：任選兩切點，將兩切點中的基因像後位移，最後一基因則移至第一位置。

   (9｜4 5 7 6 1｜3 2)→(9｜1 4 5 7 6｜3 2)

2.4.7 產生新子代

經過 2.4.4 到 2.4.6 的複製、交配與突變等步驟，舊的母體會產生出新的子代母體，新的子代母體會取代舊有的母體，常用的取代的方法如下：

1. 全部取代：不考慮產生出新子代母體的每一個體適應值，直接將新的子代母體取代掉舊有母體。
2. 菁英法則：從舊有母體中選取適應值較高的個體，再與其新產生的子代母體結合，取得另一新子代母體。

第三章 多瓶頸作業與瓶頸飄移之限制驅導零工式生產排程架構

本研究探討於零工式生產環境下，建構多瓶頸作業生產排程規劃與瓶頸漂移的求解。排程規劃考量多瓶頸與瓶頸漂移的總流程時間與交期等適應函數為評估績效，以限制驅導模式結合基因演算法做為排程演算法，對排程規劃求解。在排程開始前導入影響事件，判定多瓶頸作業的漂移情況，有影響則放入系統中 程為瓶頸站之一進行排程，沒有則於理響排程規劃後進行右移法調整影響的作業，求得合理排程解。

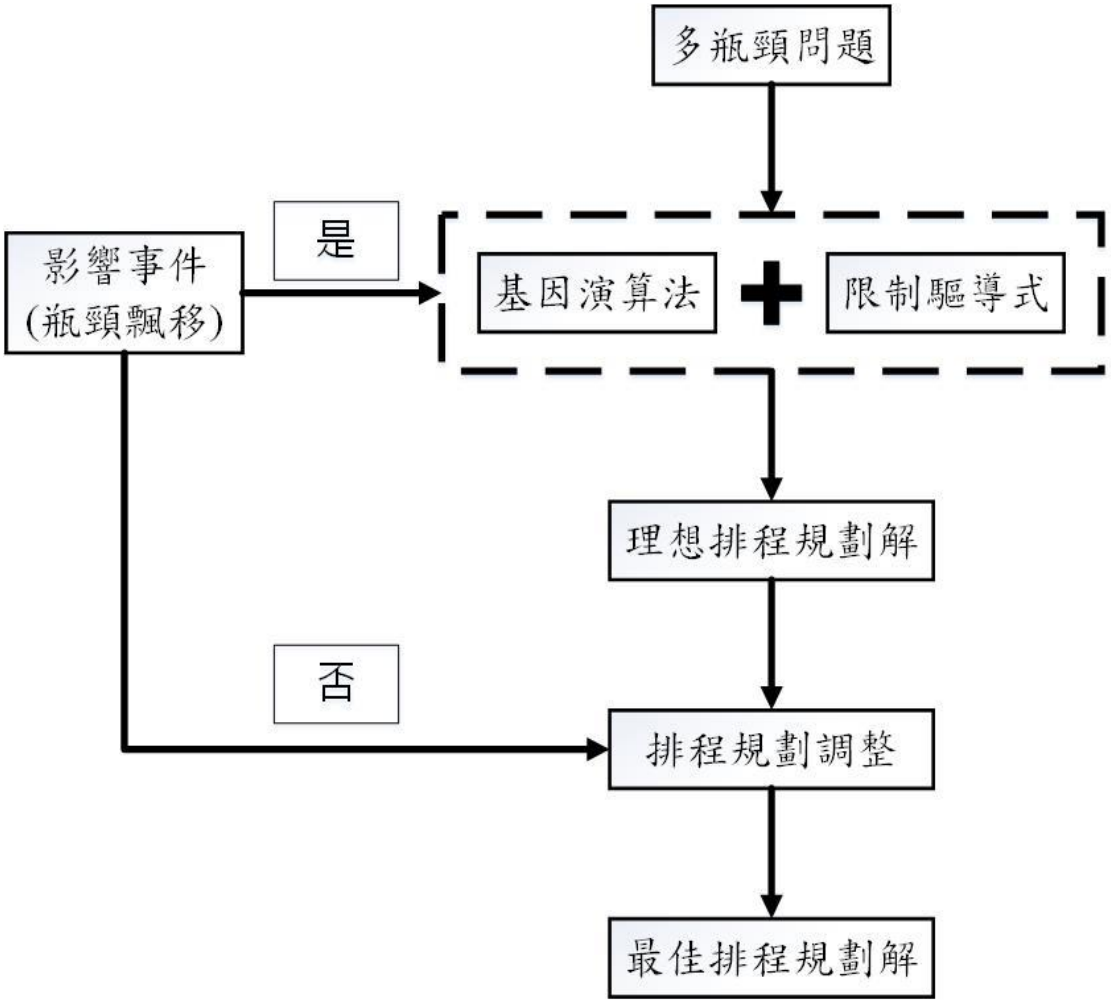

圖 3.1 多瓶頸作業與瓶頸飄移之限制驅導零工式生產排程架構

3.1 問題描述

大多數加工生產現場都是屬於零工式的生產環境，加工流程依照各別產品不同而不同，因此在生產排程上比流程式生產環境較為複雜。一般的生產排程系統中會將一個產能負荷率最高的工作站設定為瓶頸工作站，依照過往的經驗都會選最高負荷率的最為瓶頸限制所在，剩餘的則判定為非瓶頸站，若系統中產生兩個獲兩個以上產能負荷率非常接近的工作站時，將會忽略其他瓶頸對於整體生產系統的影響，造成生產規劃時效果不彰。本研究選擇將多個瓶頸限制納為生產排時的條件，並加入造成多瓶頸環境時同時發生瓶頸漂移的問題，探討對於個別流程時間與交期的對於整體績效影響。

以下為本章節中所使用之符號說明：

OB_i：訂單 i 之出貨緩衝時間。

$RB_{i,k}$：訂單 i 在機台 k 的瓶頸站間作業時間。

S_0：預計投料時間點，預設為 0。

S_i：第 i 張訂單之開工時間。

$SB1_i$：前瓶頸作業理想開工時間點。

$SB2_i$：後瓶頸作業理想開工時間點。

$SRP1_i$：第 i 訂單之前瓶頸作業理想投料時間點。

$SRP2_i$：第 i 訂單之後瓶頸作業理想投料時間點。

$P_{i,k}$：第 i 訂單在第 k 機台上的作業時間。

$b1_i$：前瓶頸作業時間。

$b2_i$：後瓶頸作業時間。

dt_i：訂單 i 之排程交期時間。

ft_i：訂單 i 之作業流程時間。

3.2 限制驅導排程運作

本研究所使用的 DBR 架構為 Sirikrai 與 Yenradee(2006)所提出的逐層回推機制，從瓶頸站的開工時間回推前一個作業的開工時間，逐層回推到投料點，避免傳統 DBR 在排程中，對於緩衝(Buffer)設置與投料繩(Rope)回推時造成緩衝長度或是緊密的缺點，導致對排程結果有不良影響，為了使瓶頸站不被中斷，本研究於瓶頸站前加入一段緩衝時間做為使排程平穩的機制。

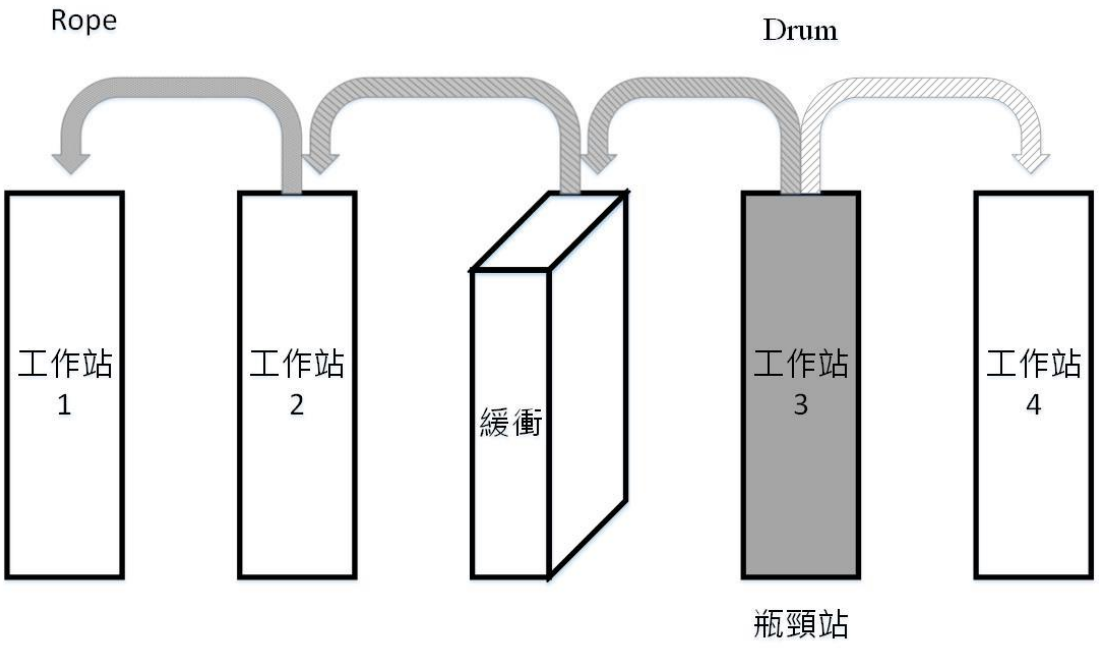

圖 3.2 本研究 DBR 架構圖

正式進入執行DBR流程，本研究採用的是Schragenheim與Ronen(1990)所提出的限制驅導排程技術的執行步驟做為依據：

1. 確認任系統限制
2. 決定系統緩衝(Buffer)大小
3. 限制驅導節奏(Drum)設計
4. 規劃投料時程(Rope)

下頁為各步驟於本研究中步驟細部說明：

步驟 1. 確認系統限制：

本研究中，以幾種問題造成多瓶頸站與瓶頸漂移原因成為限制資源，納入排程系統考量。原因分類如下：

正常：產品組合變動、機台例行維修

異常：緊急插單、機台偶發故障

根據上述原因加入訂單資訊，計算所有工作站的產能負荷，找出產能負荷最重的工作站以及與其負荷接近或相等的工作站，做為瓶頸工作站的瓶頸資源所在；或是因為上述幾種原因，產生於工作時程上高於現有瓶頸站之工作站，視之為瓶頸站的飄移並成為新的瓶頸所在位置，將此類瓶頸工作站稱之為受限資源(Capacity constraint Resource；CCR)，所有在瓶頸工作站進行作業的都稱為限制作業(constraint operations)，不是在瓶頸站作業的工作，則稱之為非限制作業(non-constraint operations)，若有一訂單中所有作業階未使用到瓶頸機台，則該訂單的全部作業就稱之為未佔用做業(free operations)。例：訂單 1、2 各有三個作業，表示為(作業代號/作業機台/加工時間)：

訂單 1：(1-1/M1/2)、(1-2/M2/2)、(1-3/M3/2)

訂單 2：(2-1/M1/1)、(2-2/M4/2)、(2-3/M3/1)

由以上訂單姿可知 M1 與 M3 機台的產能負荷加工時間皆為 3，M2 與 M4 的產能負荷都未超過前述機台，因此將 M1 與 M3 視為限制資源，而其餘則為非限制資源；若今有步驟 1 開頭所述造成瓶頸飄移之原因，導致某一機台的總作業時間超越原本限制資源的作業時間時，如：M4 因機台故障導致作業時間額外增加 2，使之作業總時間變更為 4，大於原先 M1、M3 機台的總作業時間，造成了瓶頸站的飄移，因此亦將 M4 視為限制資源。

步驟 2. 決定系統緩衝(Buffer)大小：

本研究於一訂單中若有瓶頸作業於其中，則在前一個非瓶頸作業後設置一緩衝，做為受限資源緩衝(Capacity constraint Resource Buff)。

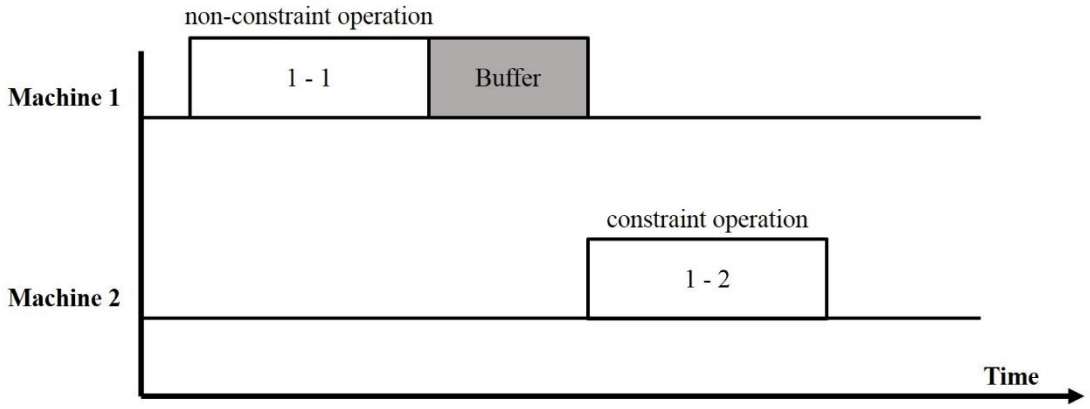

圖 3.3 緩衝設計示意圖

因為緩衝時間長短對於排程的結果影響相當大，在過去對於緩衝設置的研究中普遍使用經驗法則給定，但是當面對訂單結構發生變動時無法有效穩定排程規劃結果，因此本研究將緩衝時間設置為一決策變數，將其編碼並透過基因演算法求其最佳緩衝時間長度。對於緩衝時間的長短制定將會影響排程系統的穩定度，設置過長的緩衝時間，可以使排程系統的彈性更佳、穩定度也有提升，但會造成作業流程時間拉長、交期也會有所延後；反之，則會造成緩衝時間過短、排程系統也較無彈性，對於維持排程結果的穩定度也會降低。

因為排程是一種對於有限資源做分配的方法，所以排程中所有作業不是每一個都可以緊密連接，所以瓶頸作業的開工時間會與前一項非瓶頸作業完工時間有差異，若差異值大於緩衝時間，則以差異值作為總緩衝時間長度，反之則以緩衝值作為總緩衝時間長度。如下頁圖 3.4。

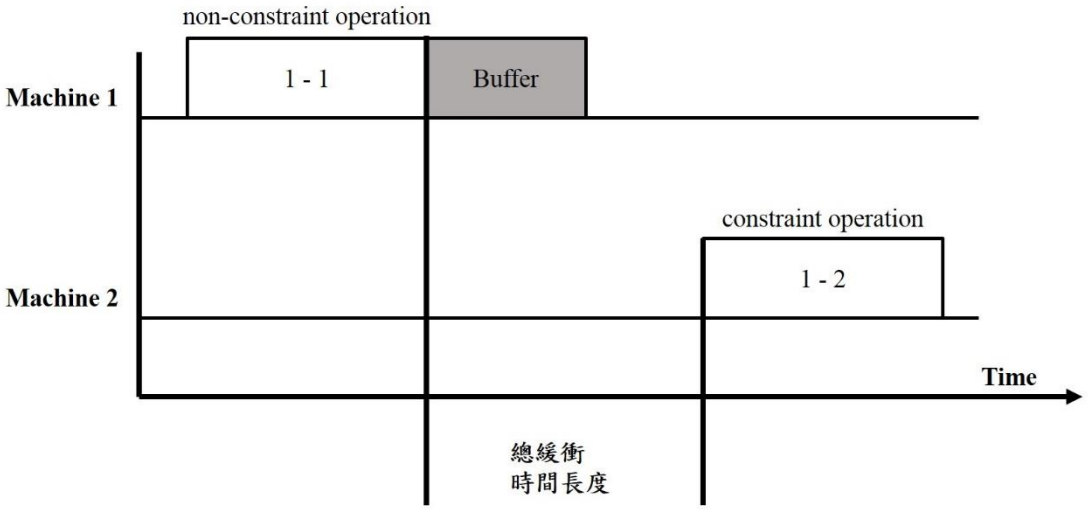

圖 3.4 總緩衝時間示意圖

步驟 3. 限制驅導節奏(Drum)設計：

在步驟 1 確定受限制作業(constraint operation)，傳統上以 Goldratt(1986)、Wu & Yeh(2006)提出的制定限制驅導節奏(Drum)方法，先將假設所有的限制作業工作站為無限產能，以規劃的作業時間為起點加上非瓶頸作業之時間總和，推算出所有訂單之瓶頸站的開工時間，制定各訂單瓶頸站理想生產時程獲得廢墟圖，再將重疊之作業時間予以推瓶，即可得到合理的瓶頸生產台程規劃，此步驟可以求得單一瓶頸站生產排程規劃的生產節奏(Drum)。本研究在步驟 1 確認初受限制作業(constraint operation)，將瓶頸站依訂單上的作業順序區分為前瓶頸站與後瓶頸站，再依傳統上制定限制驅導節奏(Drum)的方法與廢墟推平步驟，分別求得前後瓶頸站的合理生產排程節奏(Drum)。例：訂單 1、2 各有三個作業，表示為(作業代號/作業機台/加工時間)：

訂單 1：(1-1/M1/2)、(1-2/M2/2)、(1-3/M3/2)

訂單 2：(2-1/M1/1)、(2-2/M4/3)、(2-3/M3/1)

假設訂單緩衝時間長度皆為 1，其中機器 1 與機器 3 的產能負荷最高且相近，為多瓶頸機台，假設生產規畫優先順序為 1-1 優於 2-1 且 2-3 優於 1-3，因此須先確定個瓶頸站之想開工時間，在作業堆疊中若有廢墟產生則於每個瓶頸站中各自推平，可分別求得前後瓶頸站的合理開工時間，前後瓶頸站間產生之作業時間上矛盾，則於步驟 4 中進行調整以及根據前後瓶頸站的選定進行節奏重新規劃與投料時間之計算。

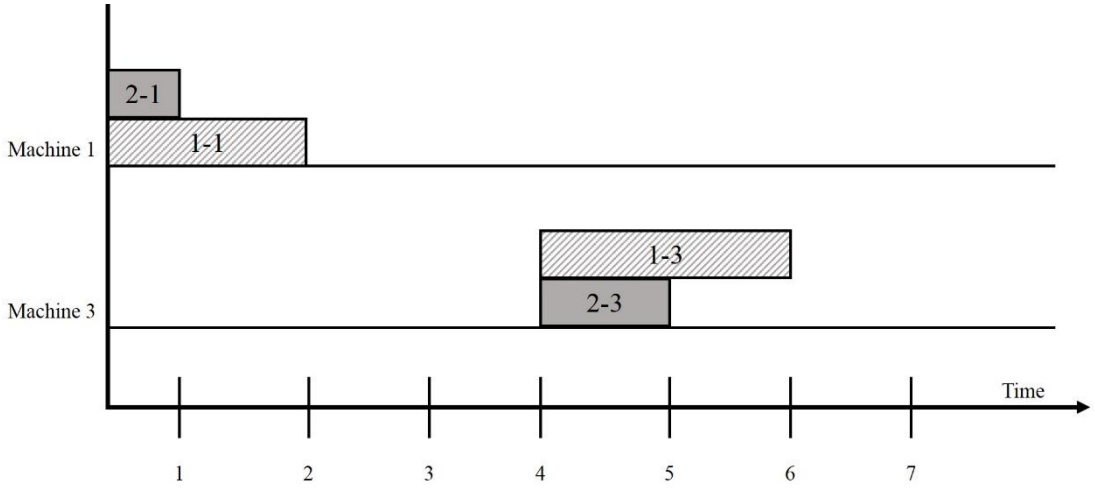

圖 3.5 多瓶頸限制驅導式廢墟圖

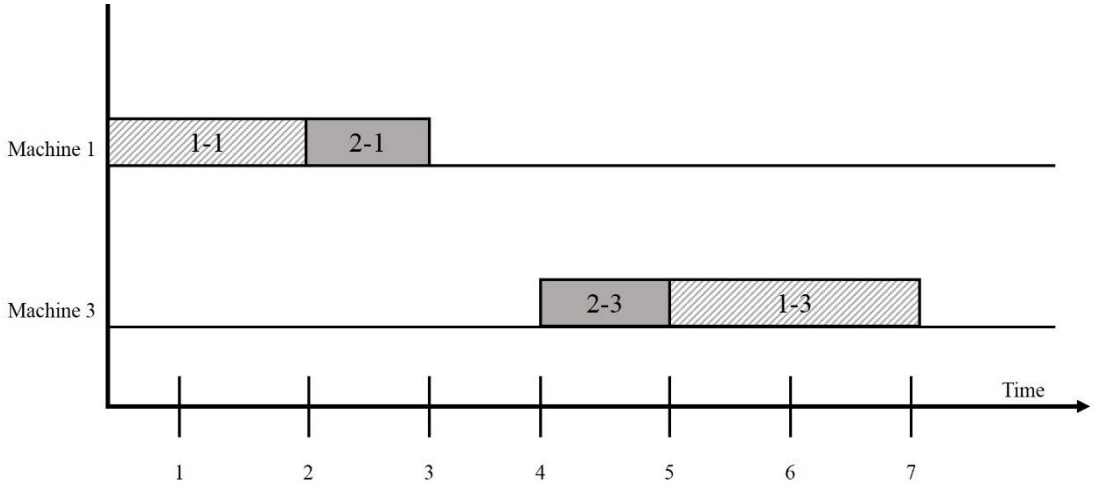

圖 3.6 多瓶頸限制驅導式各別瓶頸廢墟推平圖

步驟 4. 規劃投料時程(Rope)：

本研究根據 Sirkrai & Yenradee (2006)提出的逐層逆推方式，配合本研究所加入的瓶頸站緩衝機制，作為規畫投料時程的規劃修正。於瓶頸工作站開工時間之前放入一緩衝時間，再依據條件的不同，選擇是以前瓶頸為逆推點；或是以後瓶頸為逆推點，逐層逆推前項作業的開工時間、往後回推後續作業之開工時間，以及重新制定非主要條件之瓶頸站的限制驅導節奏，依序往回推求得每一訂單之投料時間。

以步驟 3 範例為例，機器 1 與機器 3 分別為瓶頸站，將機器 1 視為前瓶頸站之節奏做為主要條件限制驅導節奏，因為生產規劃優序為 1-1 優於 2-1，且為每一訂單之第一項作業，向後回推訂單開工會遇上後瓶頸工作站，在步驟 3 中已有先行之理想開工時間，將其排入並執行後瓶頸工作站之限制驅導節奏合理化，後瓶頸工站之後之作業將以此修正後瓶頸節奏為限制驅導節奏，並以此為新限制驅導節奏做後續作業向後回推開工時間，直至得出出貨時間。

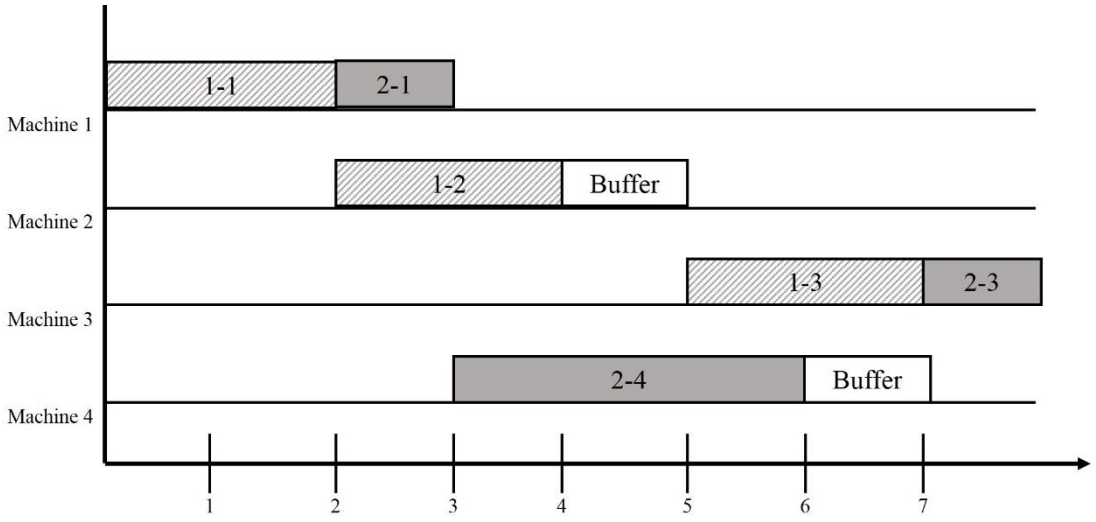

圖 3.7 前瓶頸為主投料時程甘特圖

以步驟 3 範例為例，機器 1 與機器 3 分別為瓶頸站，將機器 3 視為後瓶頸站之節奏做為主要條件限制驅導節奏，因為生產規劃優序為 2-3 優於 1-3，向前回推訂單開工會遇上後瓶頸工作站，在步驟 3 中已有先行之理想開工時間，將其排入並執行前瓶頸工作站之限制驅導節奏合理化，前瓶頸工站之前之作業將以此修正後瓶頸節奏為限制驅導節奏，並以此為新限制

驅導節奏做回推前項作業直至計算出投料時間；而為主瓶頸限制驅導節奏後瓶頸工作機台，維持原本節奏向後回推出貨時間。

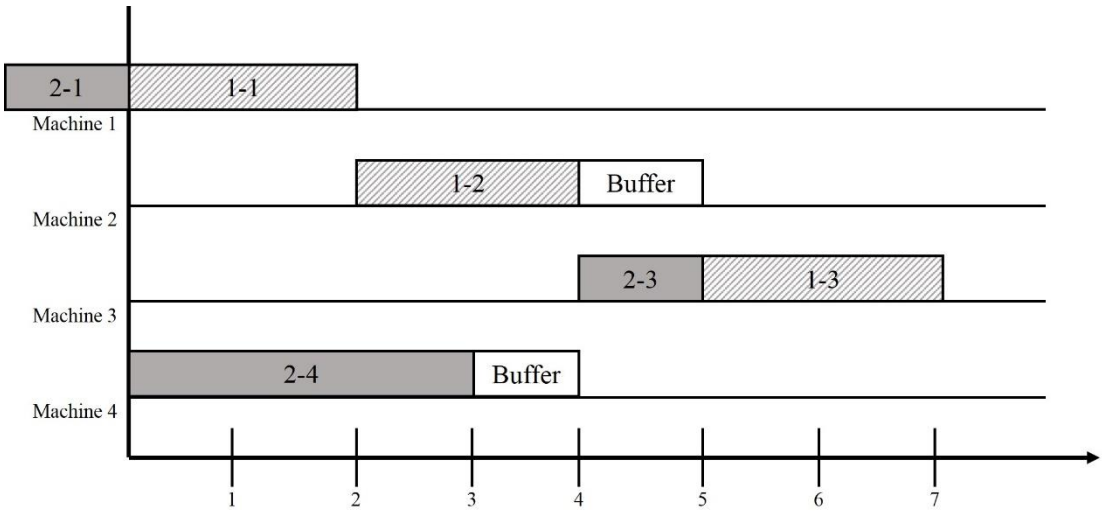

圖 3.8 後瓶頸為主投料時程甘特圖

由於本例題的作業起始開工時間為時間點 0 之位置，而如圖 3.8 所示，作業 2-1 的起始加工時間已超過規劃之起始時間，所以需進行投料時間上的修正，本研究所使用之修正方法為，將所有已排定之作業整體向右推移，直至無任何作業機台有不合理之開工時間。

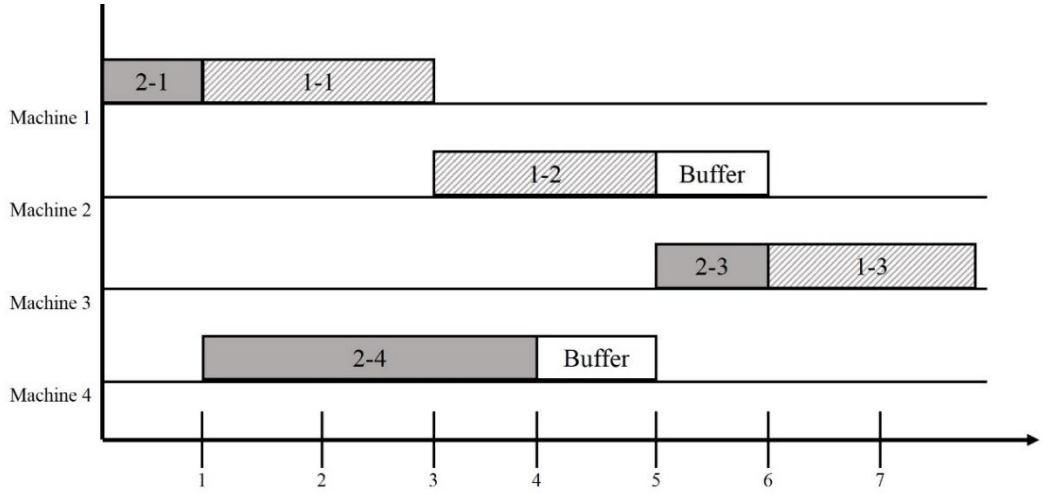

圖 3.9 後瓶頸為主投料時程修正甘特圖

3.3 限制驅導式結合基因演算法的運作架構

本小節將說明本研究中對於限制驅導式與基因演算法結合的設計與運算過程。限制驅導視為本研究生產排程規劃的主要求解方法，由於傳統上的對於限制驅導式有諸多限制，如：緩衝時間的設計、推平廢墟於作業順序上的方法皆無依訂規則，因此本研究將以基因演算法作為補足限制驅導式上的限制要求。本研究中的緩衝時間將以編碼的方式呈現，排程作業上推平廢墟的順序則以亂數編碼方式表示，依據解碼的步驟將染色體轉換成為排程之結果，並計算目標函數的適應值，評估此次排程結果的優劣，求得一排程規畫之最佳解。

1. 作業優序基因編碼

作業優序基因採用實數編碼，每一個基因代表著一項作業的排程順序，基因的排列先依照訂單序號，在根據訂單內的作業順序做排序。例：第一個基因填入訂單一的第一項作業，第二個基因填入訂單一的第二項作業，以此類推至填入第一個訂單的最後一項作業；下一個基因則從第二張訂單的第一項作業填入，直至最後一張訂單的最後一項作業，此為一條完整的作業優序染色體編碼過程。

染色體	訂單一			訂單二			…	訂單 n		
	作業 1	作業 2	作業 3	作業 1	作業 2		…			

圖 3.10 作業優序基因編碼圖

2. 緩衝時間基因編碼

緩衝時間基因則是使用 2 原編碼，每 6 個基因碼組成前後瓶頸站之緩衝時間，而每 3 個基因碼代表著一張訂單一個瓶頸工作站的緩衝時間。例：一條染色體的前三個基因為 101，代表訂單一的前瓶頸站緩衝時間為$1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 = 5$。代表訂單一的前瓶頸作業的加工時間修正為原加工時間與緩衝時間的總和。

圖 3.11 緩衝時間基因編碼圖

染色體	訂單一(前)			訂單一(後)			…	訂單 n		
	2^0	2^1	2^2	2^0	2^1	2^2	…			

3. 初始母體設計

本研究對於初始母體設計採用隨機亂數的方式產生，假設全部的訂單合計共有 N 項作業，作業優序基因的初始母體就以隨機亂填入 1～N 個不重複的數字填入。

	作業 1	作業 2	…	作業 m
訂單一				
訂單二				
…				
訂單 n				

圖 3.12 作業優序基因母體示意圖

但是以此種方式產生母體會有缺陷產生，如：某一訂單產生出的基因為[1 , 5 , 15 , 13 , 25]，以加工途程來看第三作業與第四作業的加工順序並不合理，因此須於以修正，將第三作業與第四作業的加順序作對調，使該訂單的作業順序合理化。本研究以此限制作為判定母體訂單作業順序合理性的條件。

緩衝時間基因各訂單瓶頸站的緩衝時間，以隨機 0 或 1 的方式填入，再以 2 進位進行解碼，得出瓶頸站緩衝時間。

訂單一(前)			訂單一(後)			…	訂單 n		

圖 3.13 緩衝時間基因母體示意圖

4. 目標函數適應值設計

本研究主要探討排程後總流程與訂單的交期時間兩部分，目標函數適應值計算方式如下：

$\min f(x) = \sum ft_i + \sum dt_i$，

$SB1_i = S_0 + \sum P_{i,k}$，(計算前瓶頸作業理想開工時間)

$SB2_i = S_0 + \sum P_{i,k} + \sum RB_{i,k}$，(計算後瓶頸作業理想開工時間)

$OB_i = \sum P_{i,k}$，(計算每張訂單出貨緩衝時間)

$SRP1_i = SB1_i - \mathrm{b1}_i - \sum P_{ik}$，(計算以前瓶頸為主，向前回推時每一項作業開工時間)

$dt_i = SB1_i + \mathrm{b1}_i + \sum RB_{i,k} + \mathrm{b2}_i + OB_i$，(計算以前瓶頸為主完工時間點)

$SRP2_i = SB2_i - \mathrm{b2}_i - \sum RB_{ik} - b1 - \sum P_{i,k}$，(計算以後瓶頸為主，向前回推時每一項作業開工時間)

$dt_i = SB2_i + \mathrm{b2}_i + OB_i$，計算以後瓶頸作業為主完工時間點)

$ft_i = dt_i - S_i$，(計算每張訂單流程時間)

5. 複製、挑選機制

在基因演算法中對於父代的複製、挑選，常見的方法有兩種：輪盤法 (roulette wheel selection)、競爭法 (tournament selection)。

輪盤法依每個物種(字串)的適應函數值分割輪盤上的位置，適應函數值越大則在輪盤上佔有的面積也越大，每個物種在輪盤上所佔有的面積比例也就代表其被挑選至交配池的機率，然後隨機地選取輪盤上的一點，其所對應的物種即被選中送至交配池中。假設交配池共有 N 個物種，f_N 代表第 N 個物種的適應函數值。那麼，理論上此物種應該會有 個被複製至交配池中，其中 代表所有物種的平均適應函數值。

競爭法則是在每一代的演化過程中，首先隨機地選取兩個或更多個物種(字串)，具有最大適應函數值的物種即被選中送至交配池中。

由於競爭式選擇所需的計算量較少、可以藉由一次選取物種個數的多寡來控制競爭者的速度，也可避免因輪盤法在族群演化的途中將最佳解被汰換掉，造成演算過程中適應值陷入局部最佳解的情況，因此本研究是採用競爭法做為複製、挑選的機制。

6. 交配運算子

由於本研究以訂單排程的優先順序與加工途程的工作站做為依據設計染色體，因此傳統上的交配運算無法應用於本研究的排序性問題。本研究依據 Liaw (2000)研究指出 LOX 運算子於工廠排序性問題有較佳的結果，故本研究對於作業優序基因交配運算子將採用此方法，而緩衝時間基因則採用一般的兩點交配運算子。下為 LOX 運算子步驟

隨機產生兩切點

母體 A：(9 8 4｜5 6 7｜1 3 2)

母體 B：(8 7 1｜2 3 6｜9 5 4)

將母體 A 切點中與 B 相同之基因，在母體 B 以*代替；B 者亦然

母體 A：(9 8 4｜5 6 *｜1 * *)

母體 B：(8 * 1｜2 3 *｜9 * 4)

將*往切點中間移動，使切點中所以基因為*

母體 A：(9 8 4｜* * *｜5 7 1)

母體 B：(8 1 2｜* * *｜3 9 4)

將原本母體 A、B 切點中的基因對調

子代 A：(9 8 4｜2 3 6｜5 7 1)

子代 B：(8 1 2｜5 6 7｜3 9 4)

7. 突變運算子

過往研究中大多研究都以染色體為單位考慮突變機率，而本研究則以基因為單位考量，對於每一個基因都給予不同的突變機率，界以來判定是否要執行突變，藉以增加基因演算法的搜尋空間，增加搜尋的效率；由於本研究的排程問題以訂單排程的優先順序與加工途程的工作站做為依據設

計染色體，因此本研究將 SM 運算子加以修正做為作業優序突變運算子，例：作業優序基因中判定為要突變，再搜尋剩餘基因中是否有與突變後相等基因，有則將期待換為上未突變前的基因，無則部進行代換；緩衝時間基因則仍依照本研究規則，判定每一基因是否突變，是則將原本 0 的基因代換成 1，或是 1 代換成 0。

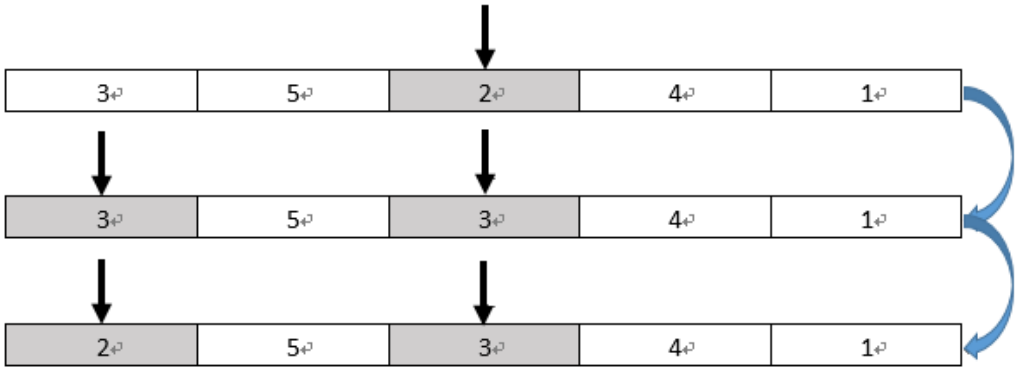

圖 3.14 作業優序突變示意圖

3.4 限制驅導式結合基因演算法的運作步驟

本小結說明限制驅導式結合基因演算法之解碼步驟，將一染色體轉換為排程規劃之結果，即一組染色體代表一個排程規劃之解，做為計算適應函數值，判斷每一染色體的優劣。

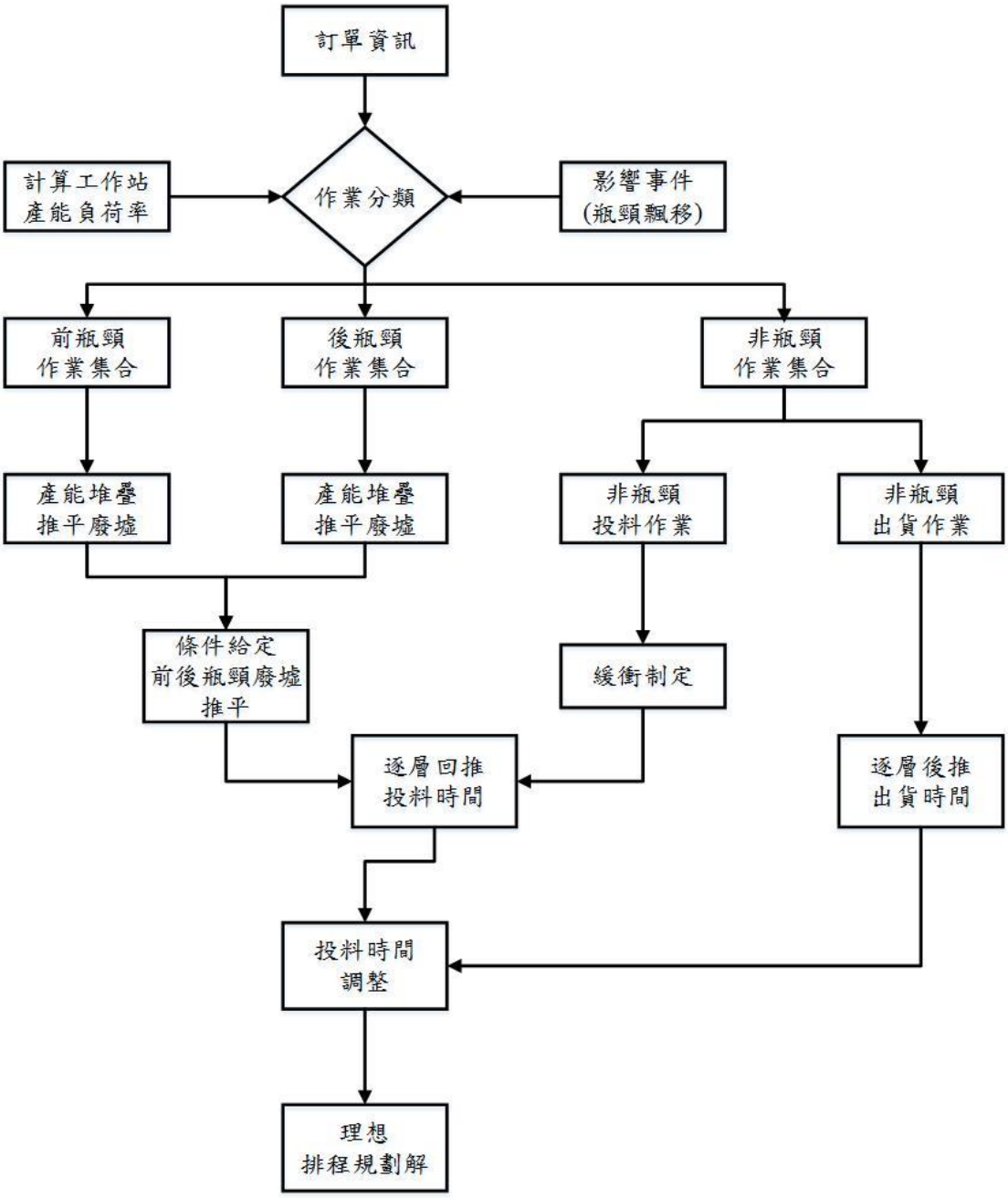

圖 3.15 解碼流程圖

下為結合限制驅導式與基因演算法之解碼步驟：

1. 先將基因編碼分類為三種集合：前瓶頸作業基因碼集合、後瓶頸作業基因碼集合、非瓶頸作業基因碼集合。
2. 對瓶頸作業基因碼組合執行排序，按照排序的先後，作為瓶頸作業解碼的順序。
3. 依據 TOC 原則，計算出各訂單投料時間與瓶頸作業開工時間。
4. 將各訂單作業切割為兩部分：前瓶頸作業、後瓶頸作業。
5. 執行前瓶頸作業排成。
6. 將理想開工時間作為要排前瓶頸作業之開工時間。
7. 判斷前瓶頸作業依序排入時，是否有廢墟產生，是則執行推平廢墟，並直至步驟 8，否則至步驟 10。
8. 重新計算各訂單之投料時間與瓶頸作業開工時間。
9. 重複步驟 7~步驟 8，直至無廢墟產生。
10. 判斷前瓶頸作業中非瓶頸作業之集合。
11. 各訂單中前瓶頸作業之前第一個作業基因代碼並且排序，作為非前瓶頸作業的作業代碼以及順序。
12. 各訂單中前瓶頸作業之前第二個作業基因代碼並且排序，作為非前瓶頸作業的作業代碼以及順序。根據步驟 11~步驟 12 逐步回推，求得非前瓶頸作業解碼順序。
13. 欲排作業的訂單中，欲排作業的下一途程作業的開工時間，作為欲排業的理想完工時間，排入該作業加工機器上。
14. 判斷排入的作業是否有閒置時間，是則選擇最晚的閒置時間作為欲排作業的完工時間，否則進入步驟 15。
15. 判斷作業機台排定作業的最早開工時間，作為欲排作業的完工時間，推算欲排作業的開工時間。
16. 根據步驟 13~步驟 15，求得所有非前瓶頸作業排定。
17. 前瓶頸作業排程完成。
18. 執行後瓶頸作業排成。
19. 將理想開工時間作為要排後瓶頸作業之開工時間。

20. 判斷後瓶頸作業依序排入時，是否有廢墟產生，是則執行推平廢墟，並直至步驟 21，否則至步驟 23。
21. 重新計算各訂單之投料時間與瓶頸作業開工時間。
22. 重複步驟 20~步驟 21，直至無廢墟產生。
23. 判斷後瓶頸作業中瓶頸站間非瓶頸作業之集合。
24. 各訂單中後瓶頸作業之前第一個作業基因代碼並且排序，作為瓶頸站間非後瓶頸作業的作業代碼以及順序。
25. 各訂單中後瓶頸作業之前第二個作業基因代碼並且排序，作為瓶頸站間非後瓶頸作業的作業代碼以及順序。根據步驟 24~步驟 25 逐步回推，求得瓶頸站間非後瓶頸作業解碼順序。
26. 欲排作業的訂單中，欲排作業的下一途程作業的開工時間，作為欲排業的理想完工時間，排入該作業加工機器上。
27. 判斷排入的作業是否有閒置時間，是則選擇最晚的閒置時間作為欲排作業的完工時間，否則進入步驟 18。
28. 判斷作業機台排定作業的最早開工時間，作為欲排作業的完工時間，推算欲排作業的開工時間。
29. 根據步驟 26~步驟 28，求得所有非前瓶頸作業排定。
30. 後瓶頸作業排程完成。
31. 判斷訂單內前後瓶頸作業間是否有廢墟產生，是則執行廢墟推平。
32. 重新各訂單之投料時間與瓶頸作業開工時間。
33. 重複步驟 31~步驟 32，直至無廢墟產生。
34. 計算各訂單之合理完工時間。
35. 前後瓶頸作業排成完成。
36. 判斷是否有投料時間小於 0，是則將最早投料時間與投料時間小於 0 之差值，將全部投料時間和瓶頸站作業時間加上此差值作為修正方法。

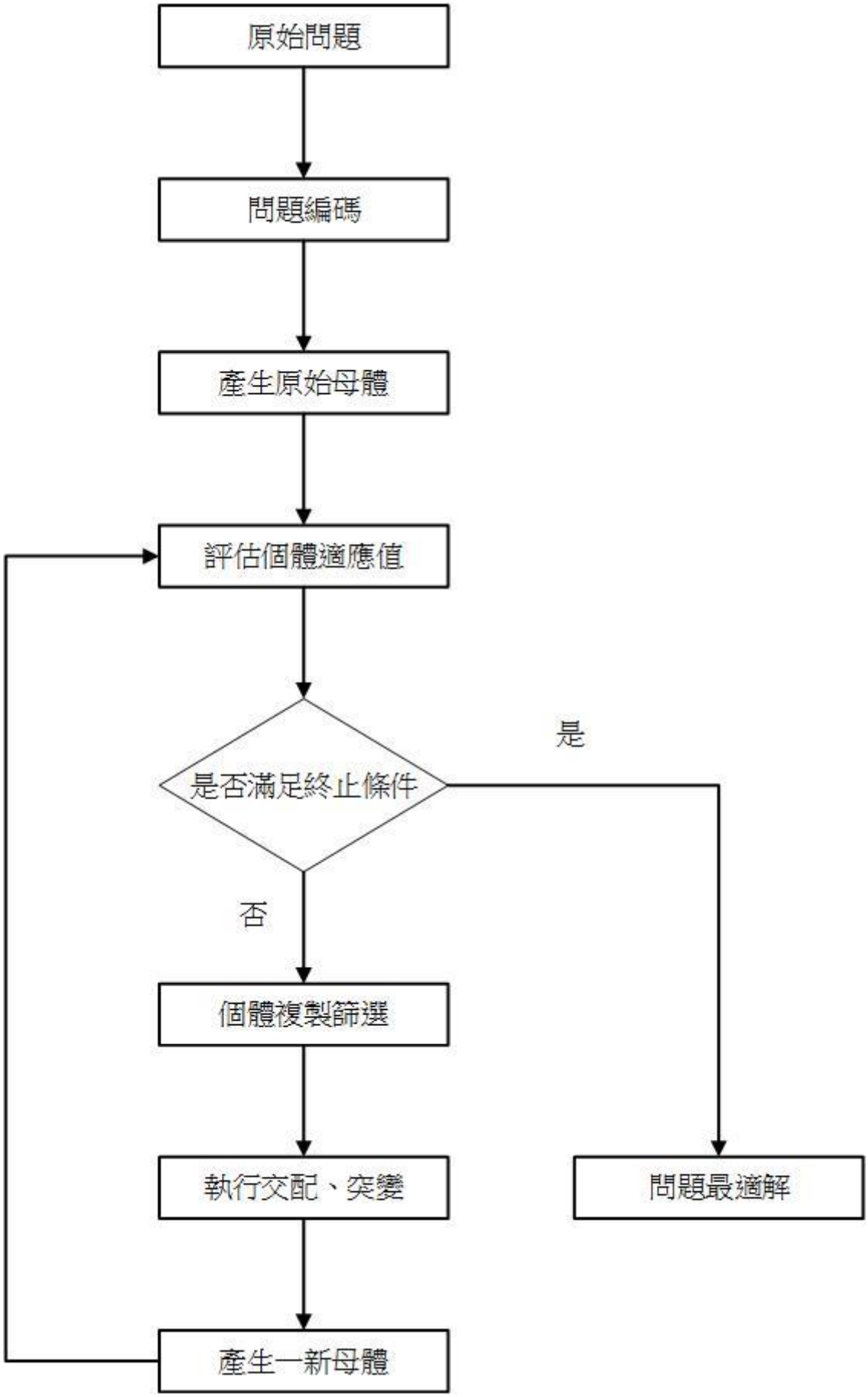

圖 3.16 基因演算法運作流程圖

第四章 系統驗證

本章節將以零工式生產訂單排程問題為例，並依照第三章所建構之方法與系統來驗證。本研究將訂單類型依照多瓶頸環境分為三種不同類型並進行系統之實例驗證：

1. 頭尾型：每一張訂單作業產生多瓶頸位置於第一作業與最後作業上。
2. 中間型：每一張訂單作業產生多瓶頸位置於訂單作業中，非第一或最後作業上，且瓶頸之間互不相鄰。
3. 相鄰型：每一張訂單作業產生多瓶頸位置於訂單作業中，非第一或最後作業上，且瓶頸之間相鄰。

於上述三種瓶頸類型訂單驗證過程中加入瓶頸飄移條件原因，使問題擴增為同時具有多瓶頸與瓶頸飄移問題，並更為貼近現實之訂單作業結構，藉由基因演算法進行排程最佳化，並根據排程最佳化之結果進行分析與歸納。

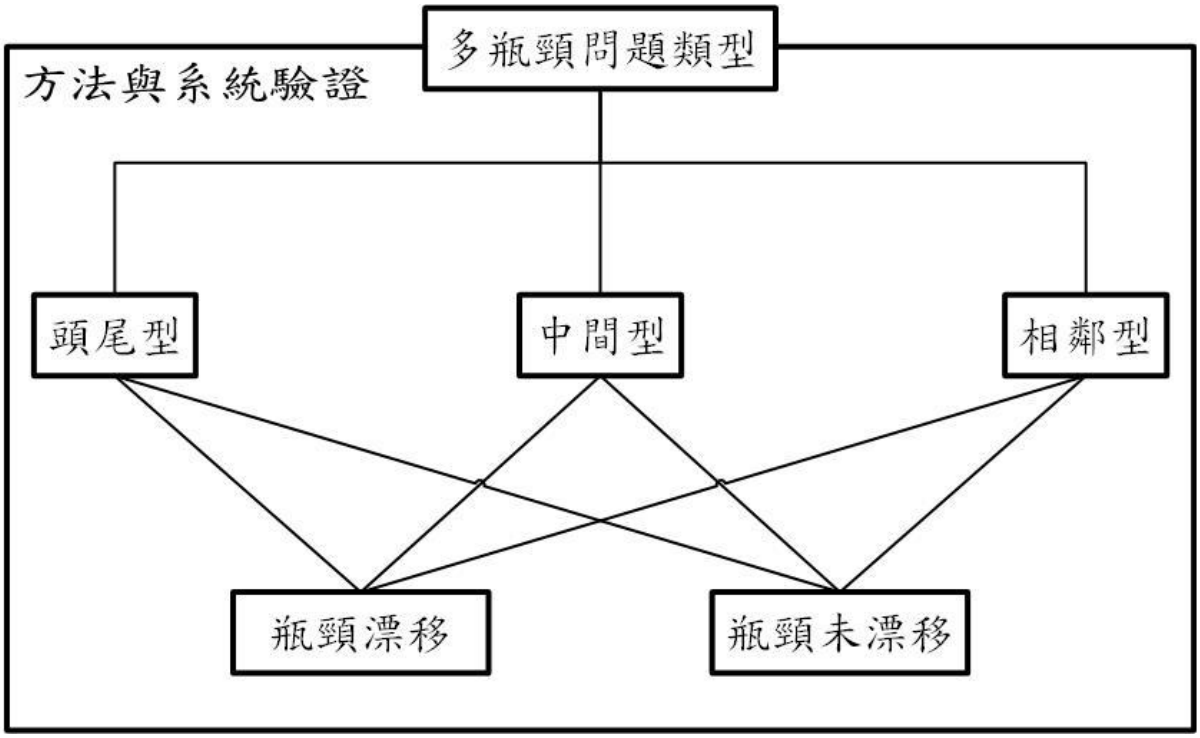

圖 4.1 實例問題架構圖

4.1 Job Shop 驗證例題一

驗證例題一中，排程環境為：現場工作環境有 16 個不同種類的作業站，各站皆僅有一台機器；訂單數量為 5 張，每一張訂單接代表不同種類產品，且每張訂單需經過之作業數量與訂單交期不盡相同。驗證例題一之訂單資料如表 4.1：

表 4.1 例題驗證一之訂單資料表

訂單	加工機台	整備時間	加工時間	交期
訂單 1	8	0.8	24	255
	4	1.8	6	
	7	1.4	16	
	1	0.9	17	
	6	2.8	17	
	3	2	23	
	11	2	23	
	12	1.8	6	
	15	1.4	16	
	9	0.9	17	
	14	2.8	17	
	16	0.8	24	
訂單 2	8	0.5	12	195
	4	0.8	4	
	1	0.4	10	
	7	0.9	8	
	5	1.3	17	
	2	2	13	
	10	2	13	

訂單	加工機台	整備時間	加工時間	交期
訂單 2	12	0.8	4	195
	9	0.4	10	
	15	0.9	8	
	13	1.3	17	
	16	0.5	2	
訂單 3	8	0.7	36	429
	2	2.2	49	
	4	1.2	12	
	3	0.4	36	
	7	0.4	21	
	11	0.4	36	
	10	2.2	49	
	12	1.2	12	
	15	0.4	21	
	16	0.7	36	
訂單 4	5	0.6	15	378
	6	0.8	24	
	4	0.7	23	
	2	0.6	20	
	1	0.4	20	
	3	1	17	
	13	0.6	15	
	14	0.8	24	
	12	0.7	23	
	10	0.6	20	
	9	0.4	20	

訂單	加工機台	整備時間	加工時間	交期
訂單 4	11	1	17	378
訂單 5	8	2.1	19	291
	4	1.7	28	
	1	1.1	19	
	5	0.8	27	
	9	1.1	19	
	12	1.7	28	
	13	0.8	27	
	16	2.1	19	

表 4.2 例題一各工作站之能負荷表

作業機台	1	2	3	4	5	6	7	8
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1
作業機台	9	10	11	12	13	14	15	16
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1

根據表 4.1 的資料，計算各工作站之產能負荷如表 4.2 所示，由表 4.2 可知產能負荷最高的分別為工作站 8 與工作站 16，此訂單資料為多瓶頸訂單作業資料，因此將將前述兩站同設為瓶頸工作站，於之後系統驗證時將分別以此做為主要條件執行基因演算法排程最佳化。排程基因演算法參數設定如下表：

表 4.3 例題一基因演算法參數值表

參數	參數值
母體大小	20
迭帶次數	150
交配方式	LOX
上半部基因突變綠	0.03
下半部基因突變綠	0.03

4.1.1 以前瓶頸為主之多瓶頸排程

本小節是以前瓶頸作為主要之排程瓶頸節奏，而後瓶頸則作為限制驅導式節奏向後推時之調整對象，經調整在之後的節奏往後推將以調整後之前瓶頸作後續動作，但因本例題是頭尾型，前後頸代表每張訂單之最後一項作業，因此不需再作後續之往後推，而是調整後之前瓶頸投料時間則為系統投料時間且最佳適應函數為

$\min f(x) = \sum ft_i + \sum dt_i$ =1235.4+1134.2=2369.6。

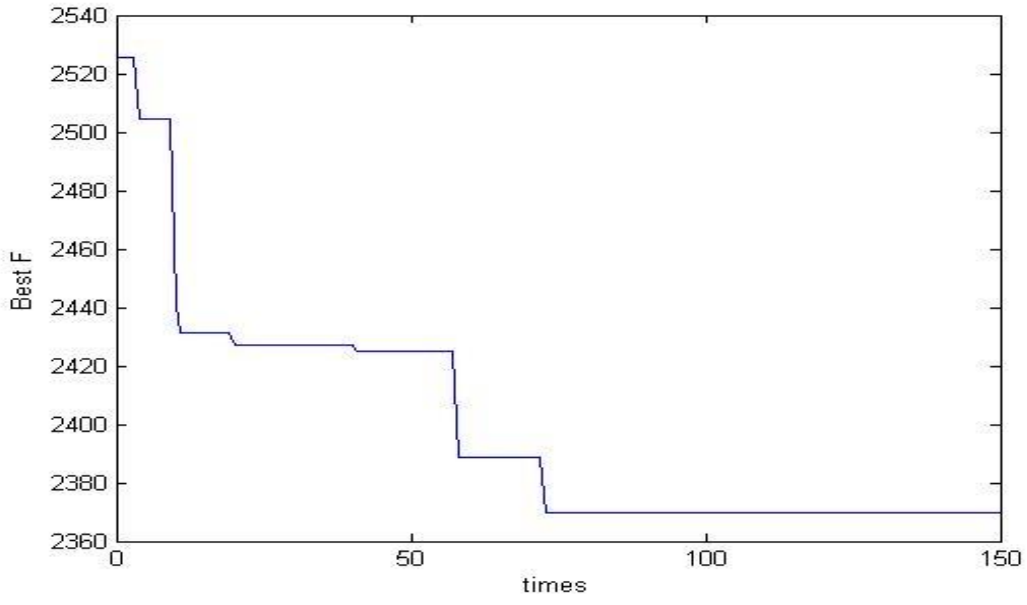

圖 4.2 例題一前瓶頸為主之適應函數趨勢圖

表 4.4 例題一前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	21.1	21.1	175.8	246.5	255
訂單 2	82.6	82.6	114.8	271.4	195
訂單 3	45.9	45.9	244.4	363.7	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	0	0	155.2	197.4	291

4.1.2 以後瓶頸為主之多瓶頸排程

本小節以後瓶頸作為主要之排程瓶頸節奏，而前瓶頸則作為限制驅導式節奏回推時之調整對象，經調整在之後之節奏回推將以調整後之前瓶頸作後續動作，但因本例題是頭尾型，前瓶頸代表每張訂單之第一項作業，因此不需再作後續之回推，而是調整後之前瓶頸投料時間則為系統投料時間且最佳適應函數為$\min f(x) = \sum ft_i + \sum dt_i$ =702+1208=1910.0。

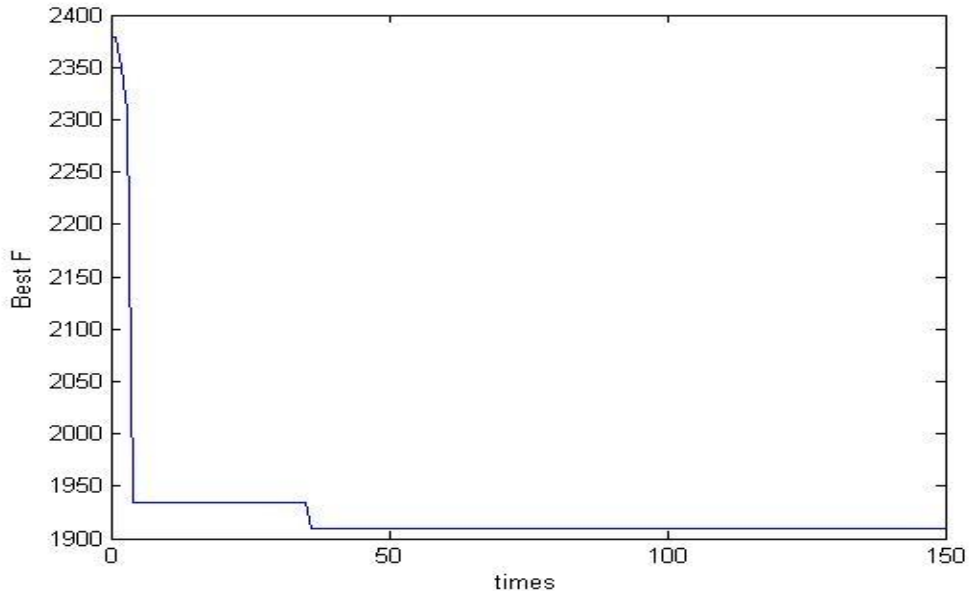

圖 4.3 例題一後瓶頸為主之適應函數趨勢圖

表 4.5 例題一前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	21.1	21.1	234.1	258.9	255
訂單 2	45.9	45.9	258.9	271.4	195
訂單 3	58.4	58.4	234.1	281.1	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	0	0	176.3	197.4	291

4.2 Job Shop 驗證例題二

驗證例題二中，排程環境為：現場工作環境有 16 個不同種類的作業站，各站皆僅有一台機器；訂單數量為 5 張，每一張訂單接代表不同種類產品，且每張訂單需經過之作業數量與訂單交期不盡相同。驗證例題二之訂單資料如表 4.6：

表 4.6 例題驗證二之訂單資料表

訂單	加工機台	整備時間	加工時間	交期
訂單 1	8	0.8	24	255
	4	1.8	6	
	7	1.4	16	
	1	0.9	17	
	6	2.8	17	
	3	2	23	
	11	2	23	
	12	1.8	6	

訂單	加工機台	整備時間	加工時間	交期
訂單 1	15	1.4	16	255
	9	0.9	17	
	14	2.8	17	
	16	0.8	24	
訂單 2	8	0.5	12	195
	4	0.8	4	
	1	0.4	10	
	7	0.9	8	
	5	1.3	17	
	2	2	13	
	10	2	13	
	12	0.8	4	
	9	0.4	10	
	15	0.9	8	
	13	1.3	17	
	16	0.5	2	
訂單 3	8	0.7	36	429
	2	2.2	49	
	4	1.2	12	
	3	0.4	36	
	7	0.4	21	
	11	0.4	36	
	10	2.2	49	
	12	1.2	12	
	15	0.4	21	
	16	0.7	36	

訂單	加工機台	整備時間	加工時間	交期
訂單 4	5	0.6	15	378
	6	0.8	24	
	4	0.7	23	
	2	0.6	20	
	1	0.4	20	
	3	1	17	
	13	0.6	15	
	14	0.8	24	
	12	0.7	23	
	10	0.6	20	
	9	0.4	20	
	11	1	17	
訂單 5	8	2.1	19	291
	4	1.7	28	
	1	1.1	19	
	5	0.8	27	
	9	1.1	19	
	12	1.7	28	
	13	0.8	27	
	16	2.1	19	

表 4.7 例題二各工作站之能負荷表

作業機台	1	2	3	4	5	6	7	8
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1
作業機台	9	10	11	12	13	14	15	16
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1

根據表 4.1 的資料，計算各工作站之產能負荷如表 4.2 所示，由表 4.2 可知產能負荷最高的分別為工作站 8 與工作站 16，此訂單資料為多瓶頸訂單作業資料，因此將將前述兩站同設為瓶頸工作站，於之後系統驗證時將分別以此做為主要條件執行基因演算法排程最佳化。排程基因演算法參數設定如下表 4.8：

表 4.8 例題二基因演算法參數值表

參數	參數值
母體大小	20
迭帶次數	200
交配方式	LOX
上半部基因突變綠	0.03
下半部基因突變綠	0.03

4.2.1 以前瓶頸為主之多瓶頸排程

本小節是以前瓶頸作為主要之排程瓶頸節奏，而後瓶頸則作為限制驅導式節奏向後推時之調整對象，經調整在之後的節奏往後推將以調整後之前瓶頸作後續動作，但因本例題是中間型，前後瓶頸工作站皆不盡為第一或是最後一項作業，因此在執行向後推時，後瓶頸工作站將會因為前瓶頸之因素，調整自身之生產節奏，產生一新後瓶頸生產節奏，後續至出貨時間之節奏則以此為準，而前瓶頸向前回推之生產節奏則為原本節奏不做調整，最佳適應函數為 2049.2。

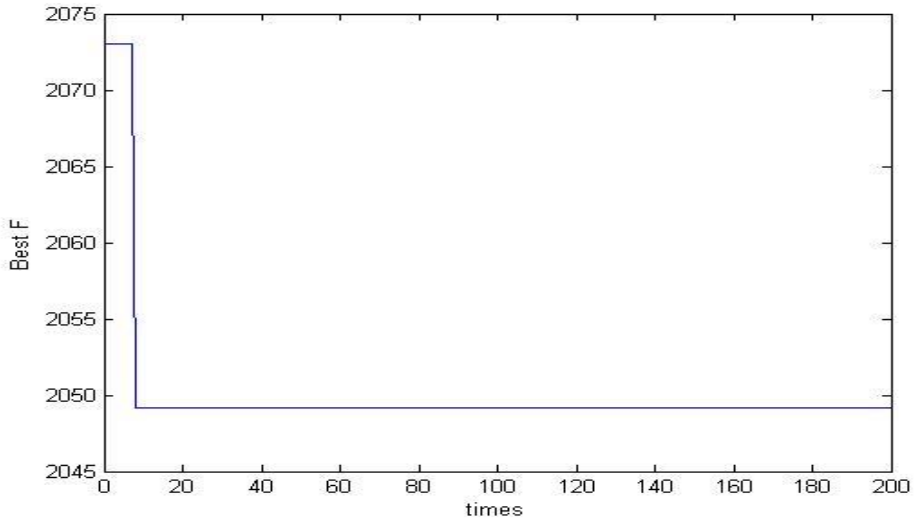

圖 4.4 例題二前瓶頸為主之適應函數趨勢圖

表 4.9 例題二前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	38.2	89.9	128.1	268.6	255
訂單 2	72.9	74.9	147.8	217.7	195
訂單 3	0	102.8	264.7	309.4	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	0	51.8	153.5	202.4	291

4.2.2 以後瓶頸為主之多瓶頸排程

本小節是以後瓶頸作為主要之排程瓶頸節奏，而後前瓶頸則作為限制驅導式節奏向前回推時之調整對象，經調整在之後的節奏往前回推將以調整後之前瓶頸作後續動作，但因本例題是中間型，前後瓶頸工作站皆不盡為第一或是最後一項作業，因此在執行向前推時，前瓶頸工作站將會因為後瓶頸之因素，調整自身之生產節奏，產生一新前瓶頸生產節奏，後續至投料時間之節奏則以此為準，而後瓶頸往後推之生產節奏則為原本節奏不做調整，最佳適應函數為 1740.0。

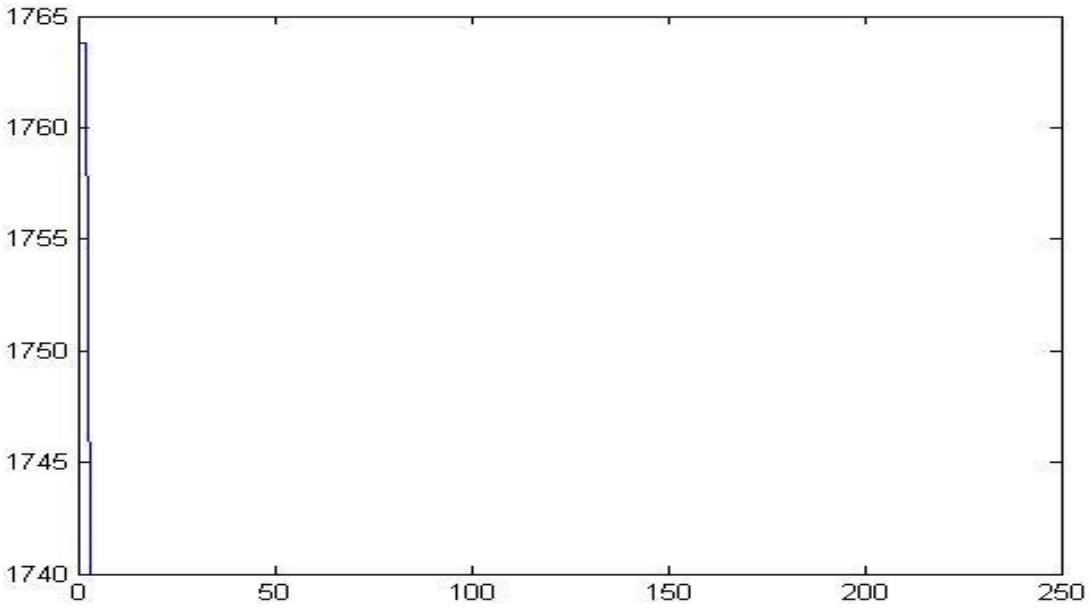

圖 4.5 例題二後瓶頸為主之適應函數趨勢圖

表 4.10 例題二前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	2	89.9	205.6	230.4	255
訂單 2	0	0	74.9	144.8	195
訂單 3	2	102.8	264.7	322.8	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	2	51.8	153.5	202.4	291

4.3 Job Shop 驗證例題三

驗證例題三中，排程環境為：現場工作環境有 16 個不同種類的作業站，各站皆僅有一台機器；訂單數量為 5 張，每一張訂單接代表不同種類產品，且每張訂單需經過之作業數量與訂單交期不盡相同。驗證例題二之訂單資料如表 4.6：

表 4.11 例題驗證三之訂單資料表

訂單	加工機台	整備時間	加工時間	交期
訂單 1	8	0.8	24	255
	4	1.8	6	
	7	1.4	16	
	1	0.9	17	
	6	2.8	17	
	3	2	23	
	11	2	23	
	12	1.8	6	

訂單	加工機台	整備時間	加工時間	交期
訂單 1	15	1.4	16	255
	9	0.9	17	
	14	2.8	17	
	16	0.8	24	
訂單 2	8	0.5	12	195
	4	0.8	4	
	1	0.4	10	
	7	0.9	8	
	5	1.3	17	
	2	2	13	
	10	2	13	
訂單 2	12	0.8	4	195
	9	0.4	10	
	15	0.9	8	
	13	1.3	17	
	16	0.5	2	
訂單 3	8	0.7	36	429
	2	2.2	49	
	4	1.2	12	
	3	0.4	36	
	7	0.4	21	
	11	0.4	36	
	10	2.2	49	
	12	1.2	12	
	15	0.4	21	
	16	0.7	36	

訂單	加工機台	整備時間	加工時間	交期
訂單 4	5	0.6	15	378
	6	0.8	24	
	4	0.7	23	
	2	0.6	20	
	1	0.4	20	
	3	1	17	
	13	0.6	15	
	14	0.8	24	
	12	0.7	23	
	10	0.6	20	
	9	0.4	20	
	11	1	17	
訂單 5	8	2.1	19	291
	4	1.7	28	
	1	1.1	19	
	5	0.8	27	
	9	1.1	19	
	12	1.7	28	
	13	0.8	27	
	16	2.1	19	

表 4.12 例題三各工作站之能負荷表

作業機台	1	2	3	4	5	6	7	8
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1
作業機台	9	10	11	12	13	14	15	16
產能負荷	65.8	86.8	79.4	79.2	61.7	44.6	47.7	95.1

根據表 4.1 的資料，計算各工作站之產能負荷如表 4.2 所示，由表 4.2 可知產能負荷最高的分別為工作站 8 與工作站 16，此訂單資料為多瓶頸訂單作業資料，因此將將前述兩站同設為瓶頸工作站，於之後系統驗證時將分別以此做為主要條件執行基因演算法排程最佳化。排程基因演算法參數設定如下表 4.13：

表 4.13 例題三基因演算法參數值表

參數	參數值
母體大小	20
迭帶次數	200
交配方式	LOX
上半部基因突變綠	0.03
下半部基因突變綠	0.03

4.3.1 以前瓶頸為主之多瓶頸排程

本小節是以前瓶頸作為主要之排程瓶頸節奏，而後瓶頸則作為限制驅導式節奏向後推時之調整對象，經調整在之後的節奏往後推將以調整後之前瓶頸作後續動作，但因本例題是鄰近型，前後瓶頸工作站皆不盡為第一或是最後一項作業，因此在執行向後推時，後瓶頸工作站將會因為前瓶頸之因素，調整自身之生產節奏，產生一新後瓶頸生產節奏，後續至出貨時間之節奏則以此為準，而前瓶頸向前回推之生產節奏則為原本節奏不做調

整，最佳適應函數為 2247.4。

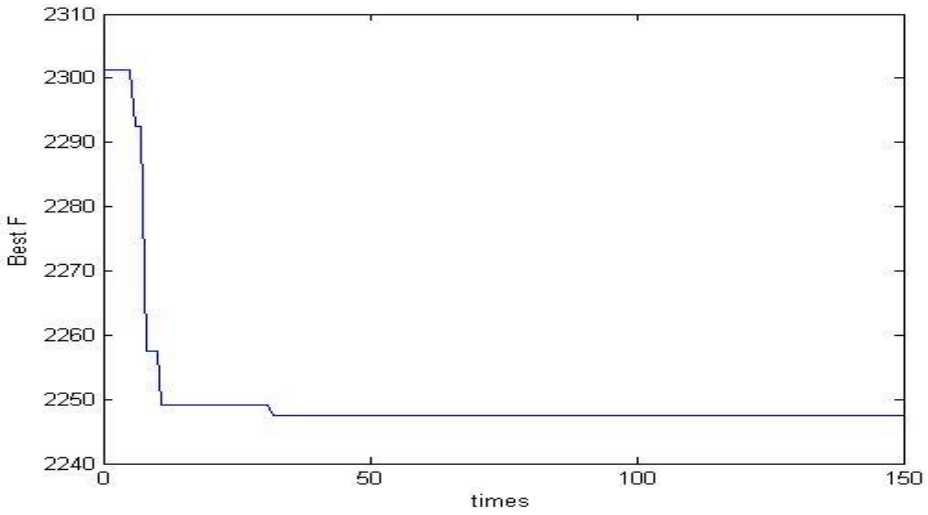

圖 4.6 例題三前瓶頸為主之適應函數趨勢圖

表 4.14 例題三前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	23.9	89.9	180.7	293.4	255
訂單 2	2	59.4	121.8	228.4	195
訂單 3	0	124.2	160.9	298.3	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	0	79.6	100.7	199.4	291

4.3.2 以後瓶頸為主之多瓶頸排程

本小節是以後瓶頸作為主要之排程瓶頸節奏，而後前瓶頸則作為限制驅導式節奏向前回推時之調整對象，經調整在之後的節奏往前回推將以調整後之前瓶頸作後續動作，但因本例題是中間型，前後瓶頸工作站皆不盡為第一或是最後一項作業，因此在執行向前推時，前瓶頸工作站將會因為後瓶頸之因素，調整自身之生產節奏，產生一新前瓶頸生產節奏，後續至

投料時間之節奏則以此為準，而後瓶頸往後推之生產節奏則為原本節奏不做調整，最佳適應函數為 2150.4。

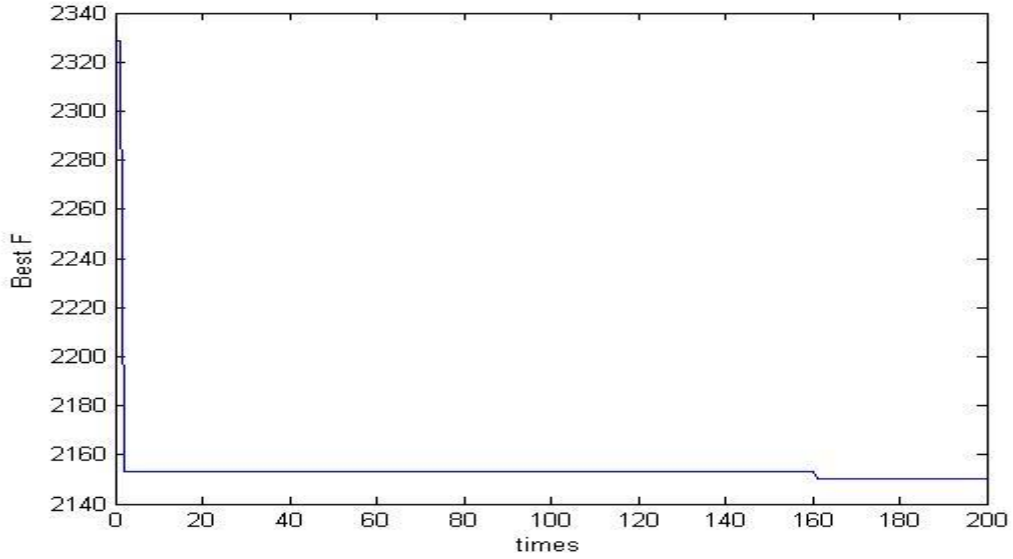

圖 4.7 例題三後瓶頸為主之適應函數趨勢圖

表 4.15 例題三前瓶頸為主之理想排程結果

訂單	投料時間	前瓶頸投料時間	後瓶頸投料時間	出貨時間	交期
訂單 1	38.4	89.9	171	230.4	255
訂單 2	12.5	71.9	158.5	144.8	195
訂單 3	0	124.2	264.7	322.8	429
訂單 4	0	N/A	N/A	246.2	378
訂單 5	0	19.6	153.5	202.4	291

4.4 本章小結

根據 4.1 至 4.3 節經由系統驗證，以三種不同訂單類型的問題，並在加入現實影響條件使所有訂單資料同時具有多瓶頸與瓶頸漂移之特性，再分別以前瓶頸與後瓶頸為主之排程進行求解，證實結果在相同的訂單環境下，可得知以後瓶頸為主之排程規劃皆優於以前瓶頸排程為主之排程規劃。如表 4.16 所示：

表 4.16 不同訂單類型與排程條件之適應值表

	前瓶頸為主排程	後瓶頸為主排程	差異值(前-後)
頭尾型	2369.6	1910.0	459.6
中間型	2049.2	1740.0	309.2
相鄰型	2247.4	2150.4	97

在三種訂單問題中，分別以不同之生產規劃條件進行排程，從排程延傳演算法中可以發現訂單中瓶頸作業之間的作業間距越趨接近，根據其條件求得之最佳適應值差異度則越小，其原因在於執行生產規劃中，後一項作業的開通時間永遠受制於前一項作業，因此當瓶頸之間間距越大時，對於排程之結果也越大，反之則越小；同理亦可得當以前瓶頸為主要排程節奏，後瓶頸做為調整之節奏時，會對於越靠近作業尾端之作業影響甚鉅，雖然於投料時間時可以求得較佳之結果，但是對於流程時間與最終交期之結果仍是不佳，而以後瓶頸作為主要排程節奏，前瓶頸為調整之節奏時，對於越靠近作業尾端之作業則是越穩定，雖然投料時間會略為不穩定，但是可以在流程時間與交期可取得較優之結果。

第五章 結論

相較於以往生產規劃僅考量單一瓶頸站所做排程，本研究加入了多瓶頸與瓶頸漂移的環境同時探討，針對零工式生產環境中常見常三種訂單型態，以基因演算法結合限制驅導式，進行兩種不同瓶頸站排程方法的比較，求得一較佳之瓶頸站排程方法，根據系統驗證之結果，本演就歸納以下幾點結論：

1. 相較於以前瓶頸為主，使用以後瓶頸為主之方法結果較優越。原因在於執行生產規劃中，不單單只有後一項作業的開通時間永遠受制於前一項作業，且與該站本身的排程順序，造成後續排程規劃的時間推延震盪增大，因相比以前瓶頸為主，以後瓶頸為主的排程方法所造成的排程影響較低。
2. 在三種不同訂單類型下，當瓶頸站間兩者作業間距越近，則本研究所提出之排程方法差異則越小。原因在於瓶頸間作業間距越遠時，兩瓶頸站間於限制驅導式排程上，造成相互間的矛盾與衝突影響則越大，反之，越靠近時，相互間所造成之穎則越小，更甚者，當兩者相鄰時，可當作單一瓶頸站執行限制驅導式排程

參考文獻

1. 何文中與洪躍(2012)。基於 TOC 理論的生產系統瓶頸識別分析與優化。**機械製造，50**，575。
2. 李偉榮(1999)。多瓶頸生產線產能規劃研究。(碩士論文，中原大學，1999)。**台灣碩博士論文知識加值系統**。
3. 沈妙妙、陳雷雷與魯建廈(2008)。生產瓶頸轉移性影響因素的經濟計量研究。**上海電機學學院報，11**，2。
4. 沈妙妙與陳雷雷(2011)。基於約束理論的生產系統瓶頸轉移問題的研究。**上海電機學院院報，14**，1。
5. 凌琳、劉明周、唐娟、趙志彪、葛茂根與蔣增強(2012)。製造車間生產物流瓶頸指數研究。**農業機械學報**，43，5。
6. 徐學軍、丁雯與余愿(2008)。基於約束理論的 MTO 型企業生產批量建模研究。**工業技術經濟，27**，11。
7. 陳清洲(2000)。多瓶頸生產線動態產能規劃研究。(碩士論文，中原大學，2000)。**台灣碩博士論文知識加值系統**。
8. 廖建閔(2005)。考慮瓶頸漂移及多瓶頸情境之 MTO 與 MTS 混線生產環境下生產控制之研究。(碩士論文，清華大學，2005)。**台灣碩博士論文知識加值系統**。
9. 賴崇瑋、姚銘忠與曾宗瑤(2001)。使用遺傳演算法求解二機開放工場具工作連接性限制問題。**東海學報，42**，79。
10. Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management science*, *34*(3), 391-401.
11. Aguilar-Lasserre, A., Posada-Gómez, R., Alor-Hernández, G., Cortés-Robles, G., Moras-Sánchez, C., Azzaro-Pantel, C., & Pibouleau, L. (2009). Multiobjective Multiproduct Batch Plant Design Under Uncertainty: Application to Protein Production. *Computer Aided Chemical Engineering*, *27*, 1101-1106.
12. Braune, R., Zapfel, G., Affenzeller, M. (2012). An exact approach for single machine subproblems in shifting bottleneck producedures for job shops with total weighted tardiness objective. *European Journal of Operational Research, 218*, 76-85.
13. Chang, P. T., & Lo, Y. T. (2001). Modelling of job-shop scheduling with multiple quantitative and qualitative objectives and a GA/TS mixture approach.*International Journal of Computer Integrated Manufacturing*, *14*(4), 367-384.
14. Cheng, H. C., Chiang, T. C., & Fu, L. C. (2011). A two-stage hybrid memetic algorithm for multiobjective job shop scheduling. *Expert systems with applications*, *38*(9), 10983-10998.
15. Della Croce, F., Tadei, R., & Volta, G. (1995). A genetic algorithm for the job shop problem. *Computers & Operations Research*, *22*(1), 15-24.

16. Della Croce, F., Tadei, R., & Volta, G. (1995). A genetic algorithm for the job shop problem. *Computers & Operations Research*, *22*(1), 15-24.
17. ElMaraghy, H., Patel, V., & Abdallah, I. B. (1999). A genetic algorithm based approach for scheduling of dual-resource constrainded manufacturing systems.*CIRP Annals-Manufacturing Technology*, *48*(1), 369-372.
18. ElMaraghy, H., Patel, V., & Abdallah, I. B. (2000). Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *Journal of Manufacturing Systems*, *19*(3), 186-201.
19. Falkenauer, E., & Bouffouix, S. (1991, April). A genetic algorithm for job shop. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (pp. 824-829). IEEE.
20. Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning* (Vol. 412). Reading Menlo Park: Addison-wesley.
21. Goldberg, D. E. (1994). Genetic and evolutionary algorithms come of age.*Communications of the ACM*, *37*(3), 113-120.
22. Gorski, J., Klamroth, K., & Ruzika, S. (2012). Generalized multiple objective bottleneck problems. *Operations Research Letters*, *40*(4), 276-281.
23. Ivens, P., & Lambrecht, M. (1996). Extending the shifting bottleneck procedure to real-life applications. *European Journal of Operational Research*, *90*(2), 252-268.
24. Kerem, B. (2011). A hybird shifting bottleneck-tabu search heuristic for the job shop total weighted tardiness problem. *Computer & Operations Research*, *38*, 967-983.
25. Lar, M., & Rene, D. (2005). A sidtributed shifting bottleneck heuristic for complex job shops. *Computer & Industrial Engineering*, *49*, 363-380.
26. Liaw, C. F. (2000). A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research*, *124*(1), 28-42.
27. Mönch, L., Schabacker, R., Pabst, D., & Fowler, J. W. (2007). Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *European Journal of Operational Research*, *177*(3), 2100-2118.
28. Pan, J. C. H., & Huang, H. C. (2009). A hybrid genetic algorithm for no-wait job shop scheduling problems. *Expert Systems with Applications*, *36*(3), 5800-5806.
29. Scholz-Reiter, B., Hildebrandt, T., & Tan, Y. (2013). Effective and efficient scheduling of dynamic job shops—Combining the shifting bottleneck procedure with variable neighbourhood search. *CIRP Annals-Manufacturing Technology*,*62*(1), 423-426.
30. Vilcot, G., & Billaut, J. C. (2008). A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *European Journal of Operational Research*, *190*(2), 398-411.
31. Wenqi, H., & Aihua, Y. (2004). An improved shifting bottleneck procedure for the job shop scheduling problem. *Computers & Operations Research*, *31*(12), 2093-2110.

32. Xu, J., Xu, X., & Xie, S. Q. (2011). Recent developments in Dual Resource Constrained (DRC) system research. *European Journal of Operational Research*, *215*(2), 309-318.
33. Zhenqiang, B., Weiye, W., Peng, W., & Pan, Q. (2012). Research on Production Scheduling System with Bottleneck Based on Multi-agent. *Physics Procedia*, *24*, 1903-1909.