

國立臺灣大學工學院工業工程學研究所

碩士論文

Graduate Institute of Industrial Engineering

College of Engineering

National Taiwan University

Master Thesis



以限制規劃求解彈性零工生產與搬運排程問題

Constraint Programming Models For Flexible Job And
Material Delivery Scheduling Problem

劉昶沅

Yun-Yuan Liu

指導教授：楊烽正 博士

Advisor: Feng-Cheng Yang, Ph.D.

中華民國 109 年 7 月

July 2020


致謝



這兩年的碩士生涯，我最感謝的人就是我的指導教授楊烽正老師。老師在學術上嚴謹的態度是我學習的典範，他對於教學的熱誠也使我獲益良多。以前的我很害怕寫程式，而且程式基礎底子也不太好。此當時進到研究室其實心中充滿恐懼與不安，每週的程式作業也壓的我喘不過氣。但是當我遇到困難去詢問老師時，老師總是不厭其煩地回答我的問題，並點出我的盲點所在，讓我逐漸在寫程式的過程中找到成就感。在撰寫論文的這段期間，老師訓練了我們對於科技論文寫作的技巧，任何段落都須要有明顯的破引申結，且不該有任何的贅句讓讀者混淆。真的非常感謝老師對我的教導，未來我會一直謹記著研究所所學的知識和老師的教誨。

除了指導教授外，我也非常感謝我的家人。謝謝父母一直支持我讀研究所的決定，提供我生活上所需讓我能無後顧之憂專心地做研究；也謝謝弟妹常常分享日常有趣的瑣事讓我的生活中有更多的趣味。謝謝靖民和詠軒，每當我對於程式或是研究上有困難時，總是會認真地幫我想解決方法。最後感謝男友的陪伴，謝謝你這八年來一直陪在我身邊，在我忙於寫論文的日子體諒我且鼓勵著我。謝謝這一路上幫助過我的人，讓我順利完成論文，在往後的日子裡我會更加地努力。

摘要



彈性零工生產與搬運排程問題源於使用無人搬運車(以下簡稱 AGV)搬運工件的彈性製造系統。此類型問題複雜度高，難以建立數學規劃求解模型，以往大多使用啟發式演算法求解。本研究針對此問題研擬限制規劃求解模型，期能求得問題的最佳解。已知各產品的加工作業順序已、加工作業的候選機台和對應的加工時間、各設施間的取貨點和卸貨點之間的距離、以及 AGV 運行的速度。求解目標是排定產品加工作業執行的順序，選定加工機台並派遣 AGV 搬運工件，排出能使產品完工時間最小的排程。本研究除了限制規劃求解模型的研擬外，也使用 IBM ILOG CPLEX 的 CP Optimizer 建構模型。為了驗證本模型的求解效能，以(李佳陽, 2019)文獻中使用的中小型 Y 群標竿和大型 L 群標竿問題為測試範例。求解結果與 GA⁺ 啟發式演算法比較。測試結果顯示本求解模型能求得比 GA⁺ 法更佳的解，降低產品完工時間並提升機台和 AGV 的稼動率。此外，因實務上產品訂單可能來自多個客戶，本研究另研擬目標函式為最小化產品總完工時間的求解模型，測試不同情境下兩種模型的求解結果分析，供使用者根據排程目標選用。

關鍵字：彈性零工生產與搬運排程問題、彈性製造系統、限制規劃、IBM ILOG CPLEX CP Optimizer、無人搬運車

Abstract



The Flexible Job and Material Delivery Scheduling Problem is derived from the flexible manufacturing system that transports the parts of products with Automated guided vehicle (hereinafter referred to as AGV). This type of problem is highly complicated, hence it's difficult to establish the mathematical programming model. In the past, heuristic algorithms were mostly used to solve this problem. However, a constraint programming model developed in this paper in order to find the best solution of this problem. The processed sequence of each operation from each product, the candidate machines for each operation and the corresponding processing time, the distance between the pickup port and the delivery port of each facility, and the moving speed of AGVs are known. The goal is to schedule the processing order of each operation, select a candidate machine and dispatch an AGV to deliver the part that can minimize the largest completion time of the products. Aside from constructing the constraint programming model, our research also implemented the model with CP Optimizer of IBM ILOG CPLEX. With an aim to verify the effectiveness of the model, the small and medium scale Y group problems and the large-scale L group problems used in the academic literature (李佳陽, 2019) are tested as benchmarks. The solution is compared with the GA^+ heuristic algorithm, and the result shows that this model can obtain a greater solution that reduces the largest completion time and improves the utilization rate of machines and AGVs. Since the production order may come from multiple customers in practice, this paper also developed another constraint programming model using the objective function of minimizing the total completion time of the products. Our research analyses the result of two models in different scenarios for users to choose according to their scheduling target.

Keywords: Flexible Job and Material Delivery Scheduling Problem, Flexible Manufacturing System, Constraint Programming, IBM ILOG CPLEX CP Optimizer, AGV

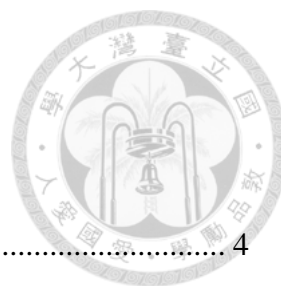
目錄



致謝.....	i
摘要.....	ii
Abstract.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第 1 章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究方法.....	3
第 2 章 文獻探討.....	5
2.1 典型生產排程問題.....	5
2.1.1. 零工式生產排程問題.....	5
2.1.2. 彈性零工式生產排程問題.....	5
2.2 考量搬運的彈性製造系統.....	6
2.3 彈性零工生產與搬運排程問題.....	8
2.4 限制條件滿足問題及限制條件優化問題.....	9
2.5 限制規劃.....	10
2.6 IBM ILOG® CP Optimizer.....	11
2.6.1. 區段變數(Interval Variable)及其限制設定.....	12
2.6.2. 序列變數(Sequence Variable)及其限制設定.....	16
第 3 章 彈性零工生產與搬運排程問題及限制規劃求解模型.....	21
3.1 彈性零工生產與搬運排程問題.....	21



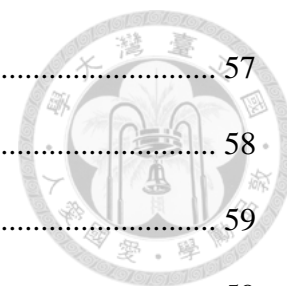
3.1.1.	問題描述與假設	21
3.1.2.	彈性零工生產與搬運排程問題的數學模式	22
3.1.3.	產品、機台、和 AGV 的狀態轉換	26
3.2	時段與時段變數	31
3.3	限制規劃求解模型	33
3.3.1.	時段變數和序列變數	33
3.3.2.	限制條件	34
3.3.3.	目標函式	47
3.4	小節	49
第 4 章	彈性零工生產與搬運排程問題的求解測試及應用	52
4.1	標竿問題	52
4.1.1.	標竿問題格式	52
4.1.2.	Y 群問題	53
4.1.3.	L 群問題	55
4.2	求解系統	56
4.3	範例測試與效能分析	61
4.4	小結	72
第 5 章	結論與未來研究建議	74
5.1	結論	74
5.2	未來研究建議	75
參考文獻	76
附錄一	78
附錄二	80
附錄三	87



圖目錄

圖 1-1 研究流程.....	4
圖 2-1 擇一匹配限制示意圖.....	15
圖 2-2 時段變數 a 和時段變數 b 的排列順位與結束時間順序無關.....	17
圖 2-3 序列變數 π 中時段變數可能的順序排列.....	18
圖 2-4 型別轉換間隔時間矩陣與時段變數轉換間隔時間示意圖.....	20
圖 3-1 機台直線型佈置的彈性製造系統.....	24
圖 3-2 機台 U 型佈置的彈性製造系統.....	24
圖 3-3 AGV 無等待派遣模式.....	27
圖 3-4 AGV 無閒置派遣模式.....	28
圖 3-5 產品等待 AGV 示意圖.....	28
圖 3-6 產品的狀態轉換示意圖.....	29
圖 3-7 機台的狀態轉換示意圖.....	29
圖 3-8 AGV 兩種排程模式的狀態轉換示意圖.....	30
圖 3-9 加工作業 oij 、空車取貨作業 aij 、和負載運送作業 bij 的時段及狀態排程關係.....	32
圖 3-10 產品 i 的加工作業和負載運送作業的時段變數排程關聯關係.....	35
圖 3-11 加工作業和負載運送作業的候選時段變數設定範例以及擇一匹配示意圖.....	41
圖 3-13AGV v 的排程結果模擬.....	45
圖 3-14 根據 AGV 負載運送作業推算空車取貨作業時間和等待時間.....	47
圖 3-15 C_{max} 相同但 C_{total} 不同的兩種排程結果.....	48
圖 4-1 標竿問題檔案內容範例.....	53

圖 4-2 求解系統人機介面	57
圖 4-3 標竿匯入問題的資訊	58
圖 4-4 問題標題和求解結果	59
圖 4-5 加工作業詳細排程結果	59
圖 4-6 機台和 AGV 的統計資料	60
圖 4-7 排程結果甘特圖	61
圖 4-8 Y3 - 4 - 3排程結果甘特圖	64
圖 4-9 Y9 - 5 - 4排程結果甘特圖	65
圖 4-10 Y9 - 5 - 5排程結果甘特圖	65
圖 4-11 L8 - 8 - 5排程結果甘特圖	68
圖 4-12 L15 - 8 - 6排程結果甘特圖	68
圖 4-13 L15 - 10 - 6排程結果甘特圖	69
圖 4-14 CP 模型使用 Minimize C_{max} 目標函式求解 L15 - 8 - 6 的排程結果甘特圖	72
圖 4-15 CP 模型使用 Minimize C_{total} 目標函式求解 L15 - 8 - 6 的排程結果甘特圖	72



表目錄



表 3-1 三種產品和三台機台的範例問題.....	37
表 3-2 AGV 在三台機台和倉儲中心間的搬運時間.....	38
表 4-1 Y3 - 4 - 3問題各加工作業的候選機台和對應的加工時間.....	54
表 4-2 Y3 - 4 - 3問題 AGV 在機台和倉儲中心間移動的時間.....	54
表 4-3 L 群問題的 18 個問題.....	55
表 4-4 Y 群問題的變數和限制式數量.....	62
表 4-5 不同求解法求解 Y 群問題的結果.....	63
表 4-6 L 群問題的變數和限制式數量.....	66
表 4-7 不同求解法求解 L 群問題的結果.....	67
表 4-8 L8 - 8 - 5、L15 - 10 - 6、和L15 - 8 - 6問題的變數和限制式數量.....	70
表 4-9 L8 - 8 - 5、L15 - 10 - 6、和L15 - 8 - 6問題的限制規劃法及GA +法求解結果.....	70

第1章 緒論




現今產品客製化的需求日益增加，且產品的生命週期縮短，因此過往大量單一的製造模式開始轉為少量多樣的智慧化生產。傳統製造業或是高科技產業為了快速適應市場的變化和顧客的多樣化需求，積極導入高度自動化的彈性製造系統，以提升機台和設備的使用效率，期望在最短的時間內完成訂單生產並交貨給客戶，達到電腦製造整合以及工廠自動化。若能有效地安排系統中產品加工作業、加工機台、以及搬運設備的排程，降低產品的完工時間，即能在最短的時間內製造多樣產品，提高生產效率以及增加競爭力。

過往已有許多文獻探討單一資源的彈性生產排程問題，而本研究探討的是彈性零工與生產搬運問題，即考量物料搬運的彈性生產排程問題。求解目標是排定產品加工作業執行的順序，選定加工機台並安排搬運設備，排出能使產品完工時間最小的排程。本章說明本研究的背景、目的、及研究方法和流程。首先說明本研究探討的問題相關背景和期望達到目的，再簡述本研究的進行步驟及流程。

1.1 研究背景與動機

彈性製造系統是高度自動化且具有製造彈性的生產系統，可適應頻繁變化的生產型態，能在短時間內生產多樣產品，讓工廠能快速回應客戶少量多樣的需​​求。但彈性製造系統的複雜度很高，因此也衍生出許多問題，例如：各加工作業應在哪台加工機台上執行、產品由哪個搬運設備運送、如何設定加工作業在機台上執行的順序、如何擺放機台位置等等。系統中產品、搬運設備、和加工機台的排程會相互影響。若沒有妥善安排，會造成產品的完工時間增加，無法如期交貨給客戶，也可能排程不當讓機台和設備有過多的閒置時間，降低稼動率以及生產效率。

已有文獻針對須安排搬運作業和選定加工機台的彈性零工式生產問題，定義彈性零工與生產搬運問題。但開發的啟發式演算法遇大型問題無法求出全域最佳解。此外，因加入搬運作業，排程問題的複雜度高，難以建立一般的數學規劃求解模型。本研究另闢蹊徑嘗試研擬限制規劃求解模型求解此問題，期望能有效求




得問題的全域最佳解。此問題範疇來自彈性製造單元(Flexible Manufacturing Cell)。在一廠區範圍內有一個用於儲存胚件、再製品、或完工件的倉儲中心，有多台執行加工作業的機台，多台搬運工件的設備，和搬運設備運行的軌道。問題假設倉儲中心和機台皆有足夠大的暫存區空間，可供搬運設備停駐或是工件存放。產品、機台、和搬運設備的數量已知，各產品的加工作業順序已知，加工作業有多台候選機台且對應的加工時間已知。機台同一時間僅能執行一個加工作業，且搬運設備同一時間也僅能搬運一種產品。搬運設備皆為等效機台，然加工機台不是。

產品抵達機台時，需等待機台空閒時才能執行加工作業；產品在機台上加工會耗用加工時間；產品加工完成後，暫時存放在機台暫存區等待搬運設備抵達機台；產品由搬運設備運送會耗用搬運時間。加工作業的執行和工件的搬運作業都是使用製造資源一段時間，因此本研究擬以時間區段(interval)表示這些作業的排程。本研究針對此問題研擬限制規劃的求解模型，須綜合考量各種作業間的限制建立各種限制條件，期望完成的限制規劃求解模型能求得最佳的排程。

1.2 研究目的

本研究就文獻探討的「彈性零工與生產搬運問題」，首次嘗試研擬限制規劃求解模型求解此問題。最佳的排程是安排的加工作業在各機台上的執行順序、選擇的加工機台、和排定的搬運作業，能最小化產品的最大完工時間。除了模型的研擬外，將實作求解系統求解現有考量搬運設備運行時間的彈性零工生產標竿問題。本研究目的如下：

- (1) 依據問題內容，包含產品的加工作業、加工作業的候選加工機台及對應的加工時間、搬運設備的運行時間等，建立彈性零工與生產搬運問題的限制規劃求解模型。
- (2) 研習限制規劃的求解手法和建模技巧，依現有求解引擎特性，建立求解本問題的限制規劃模型及開發求解系統。

- 
- (3) 以文獻中現有的範例問題為測試對象，進行不同的標竿測試。比較文獻使用的啟發式演算法和本研究的限制規劃求解法，驗證本研究研擬的模型求解的正確性和效能。
 - (4) 討論此彈性零工與生產搬運問題的限制規劃求解模型的優劣和求解效能，提出結論及後續建議。

1.3 研究方法

本研究以彈性零工生產與排程問題為核心議題。整體流程是在確定研究議題後，進行相關的文獻探討。針對本研究的方法和背景研擬限制規劃求解模型。以(李佳陽, 2019)文獻中的 Y 群問題和 L 群問題為標竿問題，使用 IBM ILOG® CPLEX 內的 CP Optimizer 建模套件建構模型，並以 C#程式語言開發此問題的優化求解系統進行範例測試。將實驗結果進行分析與比較，最後提出結論與未來發展方向。本研究流程如圖 1-1 所示。

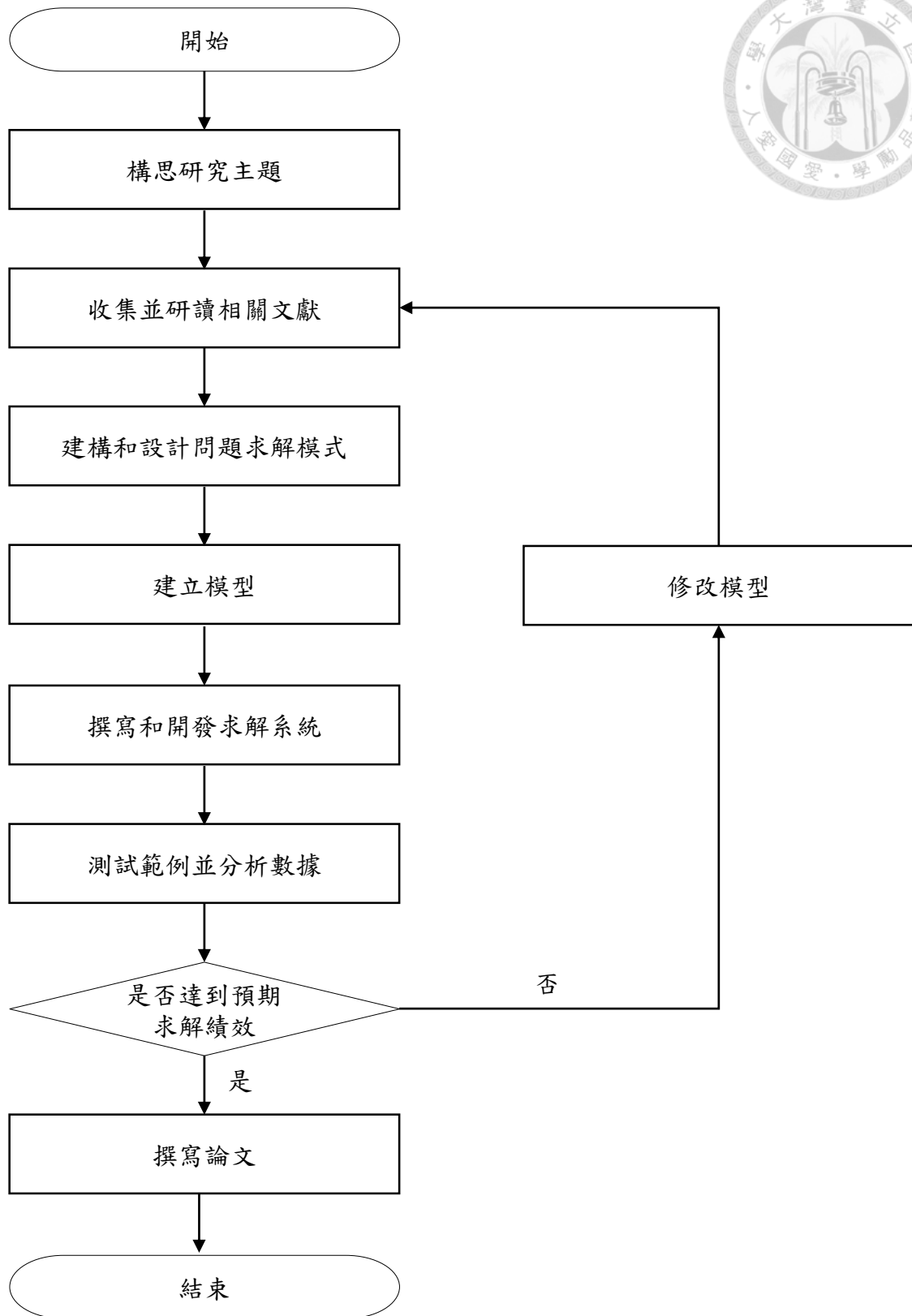


圖 1-1 研究流程

第2章 文獻探討



本章首先介紹典型的零工生產排程問題和彈性零工生產排程問題。接著探討過往考量搬運的生產排程問題相關文獻。再介紹本研究探討的彈性零工生產與搬運排程問題，分析各種求解法的研究現況和成果。此外，生產排程問題屬於限制條件滿足問題，而限制規劃是用來求解限制條件滿足問題的一種技術。因此本章也會介紹限制條件滿足問題，並說明限制規劃的定義、發展、與應用範圍。最後介紹本研究使用的限制規劃求解模型建模工具：IBM ILOG® 的 CP Optimizer。內容會注重在本研究建模使用到的變數及限制式的用法。

2.1 典型生產排程問題

2.1.1. 零工式生產排程問題

在零工式生產排程問題(Job Shop Scheduling Problem, JSP)中，產品會經過一系列的加工作業，每個加工作業都會在一台固定的機台上執行，且加工時間固定已知。每個產品的加工作業執行順序都不相同。求解 JSP 時，須排定加工作業在機台上執行的順序。求解目標是最小化產品的最大完工時間。

(Ku & Beck, 2016)文獻中探討四種不同的數學規劃求解模型求解 JSP 的效能，分別是(Manne, 1960)和(Liao & You, 1992)的分隔式數學模型(disjunctive model)、(Kondili, Pantelides, & Sargent, 1993)的以時間為基的數學模型、以及(Wagner, 1959)的以執行順序為基的數學模型。結果顯示數學規劃能在短時間內求出小型 JSP 的全域最佳解；但求解中大型 JSP 時，求解時間極長，且幾乎無法求解產品或機台數超過 15 的問題。

2.1.2. 彈性零工式生產排程問題

彈性零工式生產排程問題(Flexible Job Shop Scheduling Problem, FJSP)的機台具有多重加工能力，意即機台能執行多種加工作業，且耗用的加工時間不同。因此在 FJSP 中，任一加工作業會有多台可選擇的候選機台。求解 FJSP 時，求須從



候選機台選擇一台執行。除了依序安排各產品的加工作業執行順序外，還須選定一機台並排定加工作業的開始和結束時間。求解目標仍是最小化產品的最大完工時間。

相較於 JSP，FJSP 多了加工作業選擇機台的決策變數，複雜度大幅提升，因此過往少有文獻針對 FJSP 研擬的數學規劃模型，且求解效能不佳。Ozguven, Ozbakir, and Yavuz (2010)提出以加工作業在機台上的前後順序為決策變數的整數數學規劃模型。經測試該數學模型只能求解小型 FJSP。Demir and Kürşat İşleyen (2013)也指出隨著加工作業和機台數量的增加，CPU 求解的時間呈現指數成長。

Brandimarte (1993)提出以禁忌搜尋法(Tabu Search)為主的二階段搜尋法來求解 FSJP。此求解法將 FJSP 視為加工作業的排序和機台的選擇兩個階層的子問題，並使用禁忌搜尋法分別求解子問題。Kacem, Hammadi, and Pierre (2002)提到傳統上認為分派加工作業和調度機台是兩個不同層級的決策，若要同時考量兩個決策會提高複雜度，甚至產生新的問題。為了更有效率地求解 FJSP，Kacem et al. (2002)提出以局部搜尋法(Approach by Localization, AL)結合基因演算法(Genetic Algorithm, GA)的求解法，以同時處理加工作業和機台的決策。(Brandimarte, 1993)和(Kacem et al., 2002)兩篇文獻所提供的標竿問題，則是目前學界探討 FJSP 時常用來比較不同求解法的標竿問題。

2.2 考量搬運的彈性製造系統

第 2.1 節說明探討的 FJSP 較 JSP 加入機台選擇的彈性，已較接近工廠生產狀況。但實務上產品在機台間加工須由搬運資源搬運，因此也有些文獻開始探討考量搬運資源的彈性製造系統。

Sawik (1996)認為若在排程時未考慮工件搬運的時間，很容易造成系統出現瓶頸機台，也會使機台的閒置時間提升而降低機台的稼動率。Bilge and Ulusoy (1995)探討彈性製造系統中機台和搬運設備排程間的關聯關係，並建立該問題的非線性混整數規劃數學模型。模型中機台和搬運設備的相關限制條件會透過時窗

相互限制，並採用迭代式求解法。每個代次先使用啟發式演算法安排機台的排程，再使用移動時窗排定 AGV 的排程，產生一個可行解。

Xie and Allen (2015)將考量物料搬運的排程問題分為靜態和動態兩種問題。靜態排程問題是系統預先排定機台加工產品的排程。動態排程問題是工件抵達機台時，系統才會開始進行動態排程模擬。他們針對機台的選定和搬運設定的調度使用不同的派遣法則，並建立考量物料搬運的動態排程模擬系統。Xiao, Wu, Zeng, and Zhai (2020)結合無方向性的導引路徑網路(unidirectional guide-path network, UDN)的設計問題和 AGV 的派工問題，並提出雙母體合作演化的基因演算法(dual-population collaborative evolutionary genetic algorithm, CEGA)平行求解此二問題。

過往機械界注重在搬運設備的調度，而工業工程領域注重機台的排程(Ham, 2020)。然而，此兩者會互相影響，不可視為兩個獨立的問題。因此須同時考量搬運設備和機台的排程。Ham (2020)探討半導體產業的生產中，生產和物料搬運的同步調度，針對使用 2 台機器人搬運工件的 JSP 提出了限制規劃求解法。文獻中搬運設備的數量固定，且 JSP 中加工作業的加工機台固定。本研究之後會提出以限制規劃法求解加入搬運問題的 FJSP，同時考量機台選擇、加工作業執行順序、以及搬運設備的調度，較接近實務且複雜度更高。

Lin and Gen (2009)探討如何在使用 AGV 搬運物料的彈性製造系統中，派遣 AGV 使系統的排程最佳化。同時也將此複雜問題視為網路優化問題，並以網路架構圖表示。他們提出基於隨機密鑰的基因演算法(Random Key-based Genetic Algorithm, RKGA)求解此問題。作法是將 RKGA 法優化成基於隨機密鑰的混合粒子群演算法(Random Key-based Partical Swarm Optimization Algorithm, RKPSO)進行系統中 AGV 的調度，宣稱此求解法獲得不錯的成效。RKPSO 的交配和突變法能避免求解時過早收斂，維持解的多樣性(diversity)。文獻中測試的範例是 FJSP 結合 AGV 搬運的標竿問題。本研究之後會以(Liang, Lin, Gen, & Chien, 2012)文獻的測試結果做為參考的對象，比較限制規劃求解法與他們的方法的求解效能。



2.3 彈性零工生產與搬運排程問題

為了有效地整合機台與搬運設備，李佳陽 (2019)詳細定義了彈性零工生產與搬運排程問題(Flexible Job and Material Delivery Scheduling Problem, FJMDS)模式，說明問題和假設，內容包括產品、機台、和 AGV 的狀態轉移以及排程解的架構。問題的求解目標是找到最佳的排程解使產品的最大完工時間最小化。以 (Liang et al., 2012)文獻使用的標竿問題以及他自行擴充定義的標竿問題做為此研究的測試範例。他研擬該問題的隨機解建構法、貪婪解建構法和遺傳演算求解法，並使用 C#程式語言實作求解系統，比較各方法的求解效率。

彈性零工生產與搬運排程問題是由彈性零工式生產排程問題擴充，結合工件搬運資源的派工問題。搬運作業由每項產品初始工件於倉儲中心開始，將工件送達下一加工作業選取的機台，直到完成該項產品所有加工作業後送回倉儲中心。此問題求解時，須選定各加工作業的機台和運送工件的 AGV，也須排定其時間排程。一個可行解會由三個陣列組成，分別是加工作業順序陣列、AGV 選擇陣列和候選機台選擇陣列。

他研擬的隨機解建構法是在合理的範圍內，隨機產生可行解內加工作業、AGV、和加工機台陣列的數值。貪婪解建構法將可行的加工作業、AGV、和候選機台索引的組合視為每個解建構步驟的「分配」。每個解建構步驟中都有一組有限的「可用分配組合」可以使用，並在此解建構步驟選擇最佳的「分配」來構建可行解，逐步完成整個排程。最佳的分配是每個建構步驟中，被評估擁有最小貪婪值的分配，而貪婪值是依循該分配的最大產品可用時間。

除了隨機解建構法和貪婪解建構法，他還提出了以遺傳演算求解法(Genetic Algorithm, GA)為主的啟發式演算法(Meta-Heuristic Algorithm)。其染色體編碼採用與可行解相同的格式，每條染色體分為三段。第一段的加工作業順序陣列是由整數排列編碼(permutation)，針對此陣列使用萬用型交配與突變演算法(canonical)；第二段的 AGV 選擇陣列和第三段的候選機台選擇陣列是整數編碼，採用 NTU CALAB 開發的交配與突變法。染色體的適應值(fitness value)設定為產品最大完工時間的倒數。

母體的初使化是亂數產生加上適當比例的貪婪解，以提高解的多樣性和收斂速度。每個解建構步驟皆有一定數量的相同貪婪值，經測試後得知最佳集合產生的數量龐大。因此，此研究從「最佳分配集合」隨機挑選一個分配將貪婪解加入遺傳解進行各步驟的建構。但隨機挑選的品質差異大，所以區分為「一般貪婪解」(GA⁺)和「菁英貪婪解」(GA_x^{*})二種模式。GA⁺是在執行隨機解一次後產生的貪婪解。GA_x^{*}是產生x個一般貪婪解後從中挑選最佳作為菁英貪婪解。


該研究的標竿問題分為兩類。第一類是(Liang et al., 2012)文獻中使用的標竿問題，稱為「Y 群問題」，此類型的問題共有 3 個。第二類是由(Brandimarte, 1993)和(Kacem et al., 2002)文獻中的標竿問題加上他根據常見的工廠機台佈置(U 型和直線型)自行設計的搬運時間，成為新的標竿問題，也稱為「L 群問題」，此類型的問題共有 18 個。

他使用開發的系統求解 Y 群問和 L 群問題，並執行五種測試和兩個不同情境的分析。求解結果顯示GA⁺能求得最小的最大完工時間，平均結果也較GA和GA_x^{*}好，且GA⁺很適合求解大型的標竿問題。

啟發式演算法雖能在短時間求解大型標竿問題，但無法驗證結果不一定是全域最佳解，只能透過增加迭代(iteration)次數驗證是否還有更好的解。參數的設定以及編碼的方式會影響求解的品質，也可能陷入區域最佳解。限制規劃模型能求出小型標竿問題的全域最佳解，且不須使用者調整參數設定，同一個模型能適用所有問題模式。因此本研究研擬此問題的限制規劃模型進行求解，以驗證和比較GA⁺的求解結果。

2.4 限制條件滿足問題及限制條件優化問題

限制條件滿足問題(Constraint Satisfaction Problem, CSP)由(X, D, C)三個部分構成(Tsang, 1995)。其中集合 $X = \{x_1, x_2, \dots, x_n\}$ ，是n個決策變數。每個決策變數 x_i 可能的值已知，這些值組成 x_i 的定義域 $D_i = \{d_{i1}, d_{i2}, \dots, d_{ik_i}\}$ ， k_i 是 D_i 的大小。令 $D = \{D_1, D_2, \dots, D_n\}$ ，是決策變數的定義域集合。集合 $C = \{c_1, c_2, \dots, c_m\}$ 內有m個限制式。CSP 中的限制式可以限制單一變數的值，也可以限定多個變數間的關



係。相較於典型的作業研究領域大多在探討最佳化問題，目標是求出一個最佳解，CSP 的目標是求出一個或是多個可行解。CSP 的可行解是從各個 D_i 選定一個值設定 x_i 的值，使能滿足所有限制式。如果無法從 D 找出一組滿足 C 內所有限制式的解，則此問題稱為「無法滿足」的 CSP。許多作業研究領域的問題都可以 CSP 的形式表示，例如：排程問題、排班問題、以及派遣問題等。在定義域集合 D 內各變數值數量明確下，問題 CSP 是 NP-complete 的問題。問題難度高時沒有一個泛用的演算法能保證求出 CSP 的可行解。

使用限制規劃(Constraint Programming, CP)技術在限制條件中找尋可行解，再給定目標函式，則可以從可行解中找到最佳解，此類型問題即是限制條件優化問題(Constraint Optimization Problem, COP)。一般 CP 求解 COP 的方法是將目標函式設定成一個變數，每當找到一個可行解時，更新限制式，限制下一次找到的解目標函式的值要更好。重複這個步驟直到漸趨嚴苛的目標函式值限制無法滿足，也就是無法找到目標值更好的可行解。最後一個找到的可行解就是最接近目標的最佳解。

2.5 限制規劃

根據 Brailsford, Potts, and Smith (1999)的定義，限制規劃是運用電腦程式發展的模式化語言，讓使用者能輕易地描述限制滿足問題，並透過經規劃的電腦演算程序有效率地求解。演算法早期是使用像 PROLOG 這種傳統的邏輯式程式語言實作；但是這種邏輯式的求解程序(logical solution procedure)無法有效地處理限制式與限制式之間的關係。Jaffar and Lassez (1987)針對這個問題，改良傳統邏輯式語言程式，設計出限制式邏輯規則程式(constraint logic programming, CLP)語言。CHIP (Constraint Handling in PROLOG) 就是由 PROLOG 修改後的限制式邏輯程式語言，讓使用者能以邏輯處理的方式設定限制式及其間的關聯關係(Dincbas et al., 1988)。

隨著電腦科技的發展，限制規劃被用來求解更多實務上的問題。因此也有許多以物件導向程式語言開發的限制規劃函式庫陸續出現，例如：ILOG

Solver(C++)以及 ILOG Jsolver(JAVA)等著名的函式庫。這些函式庫使用物件導向語言實作限制規劃的演算法，以物件為出發點，有益於之後程式碼的維護。

在人工智慧領域的發展下，限制規劃使用電腦科技來求解限制條件滿足問題和組合優化問題(Combinational Optimization Problem)。限制規劃也可視為一種解的空間搜尋技術，透過系統化的搜尋技術找尋滿足所有限制的可行解。常見的搜尋演算法有區域搜尋法 (Local Search)、回溯法 (Backtracking Search)、由 Haralick and Elliott (1980)設計的往前查詢法(Forward Checking, FC)以及由 Sabin and Freuder (1997)首創的分支一致法(Maintaining Arc Consistency, MAC)。

相較於數學規劃(mathematical programming)，限制規劃沒有標準的模型建構語法，隨著不同的函式庫會有不同的表示方式。由於限制規劃使用的限制式邏輯程式語言是屬於宣告式語言(Declarative language)，能使用易建置的陳述式來表示難以線性化的限制條件，且限制式在模型中的前後順序不會影響求解結果。因此對於複雜的 CSP，建構和修改限制規劃模型會比傳統數學規劃模型容易。

限制規劃應用的領域廣泛，但是尚未被廣泛運用在作業研究的領域中。Simonis (2008)指出作業研究中的排程、資源分配、運輸、和人員排班等問題是最適合使用限制規劃求解的問題。原因是這些問題較易依照問題需求的變化增加或修改模型內的限制式，而不須要改變整個模型的架構。

2.6 IBM ILOG® CP Optimizer

IBM ILOG® CPLEX 內的 CP Optimizer 建模及求解引擎，提供限制條件滿足問題和限制優化問題等問題建模需要的變數、函式、運算式、和限制式(Laborie, Rogerie, Shaw, & Vilím, 2018)。使用這些元件建立的 CP 模型能由使用限制規劃技術建置的求解引擎求解。模型的建構能以 ILOG 的優化程式設計語言 (Optimization Programming Language, OPL) 建立，CPLEX 也提供了 Concert Technology，讓使用者能用 C#、C++、和 JAVA 等程式語言建立 CP Optimizer 的 CP 模型並叫用求解引擎函式求解問題(Laborie, 2009)。以下介紹 CP Optimizer 提供的決策變數、限制式、以及運算子，並說明個別的用途。



2.6.1. 區段變數(Interval Variable)及其限制設定

數學定義裡的區段(interval)是一個由上、下兩個邊界點(boundary point)定義的所有實數集合。譬如，區段 $[a, b] = \{z \in \mathbb{R} \mid a \leq z \leq b\}$ ；也就是包含 a 、 b 、及兩者間所有實數的集合。區段可以用來表示一段時間、一個範圍、和一段距離等無法僅用單一實數代表的變數。CP Optimizer 引用數學的區段，提供區段變數(Interval Variable)作為決策變數，以有效模式化問題。

CP Optimizer 提供許多以函式形式表示的運算子(operator)，一般的運算子作用在運算元(operand)上輸入運算子會得到運算值。形成的運算式(expression)，可用以限定變數屬性或建立變數間的限制式。以區段變數為例，區段變數係以物件形式存在 CP 的模型中，因此區段變數會有自己的運算子，顯示自身的屬性值，或限定自身屬性產生限制式。以下將會介紹本研究研擬的 CP 模型作用在區段變數的運算子。

彈性零工生產與搬運排程問題中的作業在執行時，會使用機台或是搬運設備資源加工及運送，問題涉及的區段都屬於時間區段。區段用於本問題的排程解是使用資源的一段時間，以下以「時段變數」簡稱排程問題中代表某作業的時間區段變數。

時段變數在排程問題中可用以表示任務的執行、資源的派遣、物料的移轉、工件的加工、甚或在佇列裡等待等。每個時段變數都由開始時間、結束時間、作業時間、和一個此時段變數是否是可選(optional)的布林值所組成。時段變數除了以上的屬性外，也可設定時段變數的型別(type)。時段變數可以根據特性歸類其型別，像是產品的編號、資源的種類、或作業的狀態等。型別的編號只能以整數表示。例如：某加工作業在機台 2 上執行，則該作業的時段變數的型別設定為 2。當任二時段變數相鄰且各有不同型別時，可以區別該相鄰的關聯類型以設定其他限制條件。時段變數的型別與序列變數的限制式相關，其中序列變數會在本章後續定義，因此行別的用處會在本章序列變數的部分詳細介紹。

時段變數的起始時間、結束時間、和持續時間

令 x 是時段變數， x 的起始時間和結束時間分別可用運算子 $StartOf(\cdot)$ 和 $EndOf(\cdot)$ 取得求解後 x 的排程。 x 的大小可用運算子 $LengthOf(\cdot)$ 取得。因此

$$LengthOf(x) = EndOf(x) - StartOf(x) \quad (2.1)$$

選擇性

在 CP Optimizer 的模型中，時段變數預設會存在於最終的排程解。設定時段變數 x 是可選時須使用限定運算子 $SetOptional(\cdot)$ 。因此限定 x 是可選擇的變數是限制式

$$SetOptional(x) \quad (2.2)$$

因此，若時段變數 x 被設定是可選的，則時段變數 x 可能存在(present)或是不出現(absent)在排程解中。CP 求解引擎會決定可選的時段變數是否出現在求解結果。若可選的時段變數未被選取，不會出現在排程解中。令 s 是時段變數 x 的起始時間， e 是 x 的結束時間，若 x 是可選的，則 x 的定義域可用

$$Domain(x) = \{\perp\} \cup \{[s, e] | s, e \in \mathbb{Z}, s \leq e\} \quad (2.3)$$

來表示。式中 \perp 符號表示 x 沒被選取，否則會是由起使的時間 s 到結束時間 e 的一個時段變數。

時段變數是否出現在排程解

CP 的一般運算子 $PresenceOf(\cdot)$ 可用以檢出某可選的時段變數是否出現，取得時段變數是否出現的邏輯值，即布林值 $True$ 或 $False$ ，供其他運算式或限制式使用。因此

$$PresenceOf(x) = \begin{cases} True, & \text{if } x \text{ 出現在排程解中。} \\ False, & \text{otherwise} \end{cases} \quad (2.4)$$

擇一匹配限制條件

時段變數的可選性有助於求解具有多重選擇的加工作業的排程問題。在彈性零工式生產排程問題(Flexible Job Shop Scheduling Problem, FJSP)中，加工作業可在各個不同的機台上執行，選擇不同機台加工產生的排程解都不相同。因此這些候選的時段變數都是可選的，加工作業的時段變數會從中擇一匹配，即意味從候選機台選擇一個機台加工。

設 $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ 內有 n 個選擇性的時段變數 x_i 所組成。若要讓 CP 求解引擎從 \mathbf{X} 選出一個時段變數出現在排程解，且其餘不出現，則須設定一個時段變數 y ，再以 CP 的限定運算子 $Alternative(\cdot, \cdot)$ 建立一個擇一匹配的限制式，因此限制式

$$Alternative(y, \mathbf{X}) \quad (2.5)$$

限定當 y 出現在排程解中，即 $PresenceOf(y) = True$ 時，系統會從 \mathbf{X} 中選擇一個時段變數作為匹配，也可稱 y 是 \mathbf{X} 的主要時段變數，而 \mathbf{X} 中的時段變數就是 y 的候選變數。 y 的起始時間、結束時間、以及作業時間由引擎從 \mathbf{X} 中選取其一設定。假設選擇 x_k 匹配 y ，則 y 的作業時間、起始時間、和結束時間都會設成與 x_k 相同。也就是 $StartOf(y) = StartOf(x_k)$ 且 $EndOf(y) = EndOf(x_k)$ ，因此兩者的時段常相等， $LengthOf(y) = LengthOf(x_k)$ 。相反地，若 y 是可選的但沒被選取，即 $PresenceOf(x) = False$ ，則 \mathbf{X} 內所有變數都不會出現在排程解。

假設如圖 2-1 所示， y 是主要時段變數， x_1 、 x_2 、和 x_3 都是選擇性的時段變數，是 y 的候選變數，供 y 擇一匹配。圖中以虛線描繪的時段變數是選擇性的時段變數。灰色的時段變數代表存在於排程解。假設 CP 求解時選擇 x_2 作為 y 的匹配， y 的起始時間、結束時間、以及作業時間都與 x_2 相同。

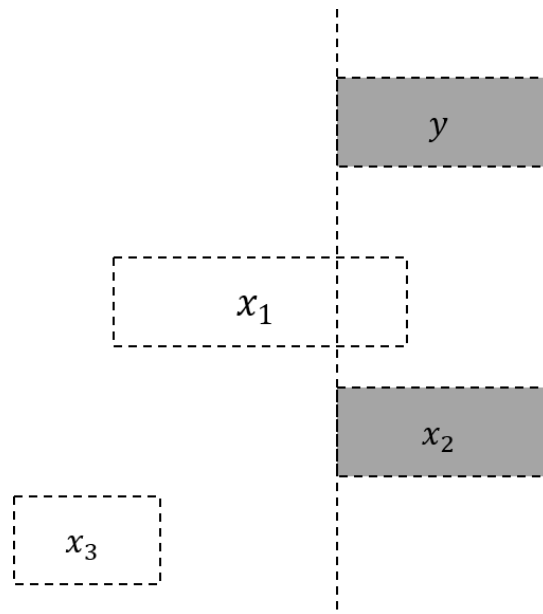


圖 2-1 擇一匹配限制示意圖

時段變數前後順序限制


在 FJSP 中作業有前後順序限制，CP Optimizer 提供各種限定運算子能規範時段變數的順序關係，二元運算子 $EndBeforeStart(·,·)$ 能限定兩變數的前後順序，而二元運算子 $EndBeforeStart(·,·,t)$ 能限制兩變數的前後順序外並限定二者的間隔時間值。舉例來說，假設某加工作業完成後，須經過至少 t 時間單位，另一加工作業才可開始加工。令 a 和 b 代表兩加工作業的時段變數，則限制式

$$EndBeforeStart(a, b, t) \quad (2.6)$$

會將時段變數 a 排在時段變數 b 之前，且中間相隔至少 t 時間單位。也就是 $EndOf(a) + t < StartOf(b)$ 。限制式 $EndBeforeStart(a, b)$ 僅限定 $EndOf(a) \leq StartOf(b)$ ，也就是時段變數 a 須排在時段變數 b 之前，因此結果兩者可能相連在一起。

邏輯運算子

除了與時段變數相關的運算子外，CP Optimizer 也提供各種邏輯運算子讓模型建構者混用以建立複雜的運算式和限制式。例如：二元運算子 $AND(·,·)$ 、 $OR(·,·)$ 、 $IfThen(·,·)$ 及一元運算子 $NOT(·)$ 等。以下將以運算子 $PresenceOf(·)$ 建立



的運算式做為邏輯運算子的運算元，示範複雜限制式的建立。邏輯運算子 $AND(PresenceOf(x), PresenceOf(y))$ 執行 AND 邏輯運算。若 $PresenceOf(x) = True$ 且 $PresenceOf(y) = True$ ，則運算值為 $True$ 。也就是 CP 求解結果時段變數 x 和時段變數 y 都必須出現在排程解。類似地，邏輯運算子 $OR(PresenceOf(x), PresenceOf(y))$ 執行 OR 邏輯運算。當 $PresenceOf(x) = False$ 且 $PresenceOf(y) = False$ 時，其值為 $False$ 。之間只須有一個條件式以上成立即可。同理一元邏輯運算子 $NOT(PresenceOf(x))$ 執行 NOT 邏輯運算。當 $NOT(PresenceOf(x))$ 值為 $False$ 時 $PresenceOf(x) = True$ 。相對地，運算子 $IfThen$ 用來建立若 P 則 Q 的限制式。因此 $IfThen(PresenceOf(x), PresenceOf(y))$ 限定若 $PresenceOf(x) = True$ ，則 $PresenceOf(y) = True$ 。也就是說，若 x 出現在排程解，則 y 也必須出現在排程解。

2.6.2. 序列變數(Sequence Variable)及其限制設定

CP Optimizer 的序列變數(Sequence variable)是一種特別的決策變數。序列變數轄有一組需排定順序的時段變數，求解時由系統安排這些時段變數的順位。序列變數可用以定義某一個資源的使用排程，即安排使用該資源的時段變數的先後順位。求解時序列變數須賦予每個時段變數一個排列順位。

設序列變數 π 的管轄時段變數集合 $\{x_1, x_2, \dots, x_n\}$ 內有 n 個時段變數 x_i 。若 π 內轄的時段變數 x_i 是選擇性的，CP 求解引擎只會排列最終存在的時段變數，未出現在排程解的選擇性時段變數不會有排列的順位。使用者可在序列變數上限制轄內時段變數的特定順位關係。以排程問題為例，機台是提供加工服務的資源，因此每台機台都可以建立一個對應的序列變數。在同一機台上執行的不同產品的加工作業時段變數則由該機台序列變數管轄。求解時排定各加工作業的加工順序。

選擇性的時段變數也屬於其使用資源的序列變數，CP 求解引擎會決定序列變數中的哪些時段變數會存在於排程解，並設定存在於排程解的時段變數的執行順序。序列變數內轄的時段變數會排定順位，但未規範時段變數是否會交疊在一

起。序列變數的排序結果只表示時段變數執行的順位，不代表時段變數結束時間的絕對順序。以圖 2-2 為例，在同一序列變數中，時段變數 a 的順位在時段變數 b 之前，但時段變數 a 的結束時間在時段變數 b 的結束時間之後，也就是 $EndOf(b) < EndOf(a)$ 。此情形意味序列變數代表的資源同時間內可以提供多的服務時段。例如，一間多座椅的休息室，可提供多人使用一段時間。

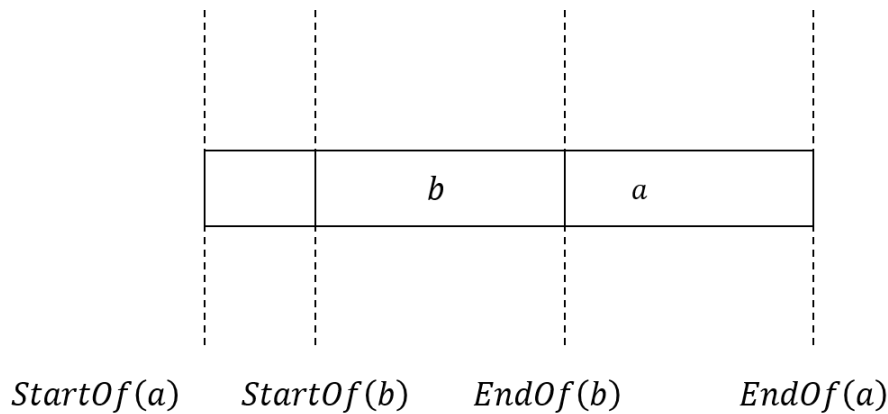


圖 2-2 時段變數 a 和時段變數 b 的排列順位與結束時間順序無關

假設序列變數 π 管轄 x 、 y 、 z 、 p 、和 q 等五個選擇性的時段變數所組成。若 p 和 q 未出現在求解結果，則 $(x \rightarrow y \rightarrow z)$ 、 $(x \rightarrow z \rightarrow y)$ 、 $(y \rightarrow x \rightarrow z)$ 、 $(y \rightarrow z \rightarrow x)$ 、 $(z \rightarrow x \rightarrow y)$ 、 $(z \rightarrow y \rightarrow x)$ 六種順序都是可能的排程解。圖 2-3 展示系統求解時序列變數 π 內的時段變數的順位設定，圖(a)-(e)是六種可能的順位設定。序列變數中的時段變數可能會交疊，因此圖中以圓形描繪的時段變數，只表示時段變數的排列順位。圖例因時段變數的排程結果不一定是分開的。CP 求解結果 x 、 y 、和 z 會存在於排程解。 p 和 q 未出現在排程解，以中間有兩條交叉線的圓形表示。若求解結果當五個時段變數都不存在時，表示該資源都沒有執行任何作業，因此序列變數 π 不會排出任何時段變數的任何順位。

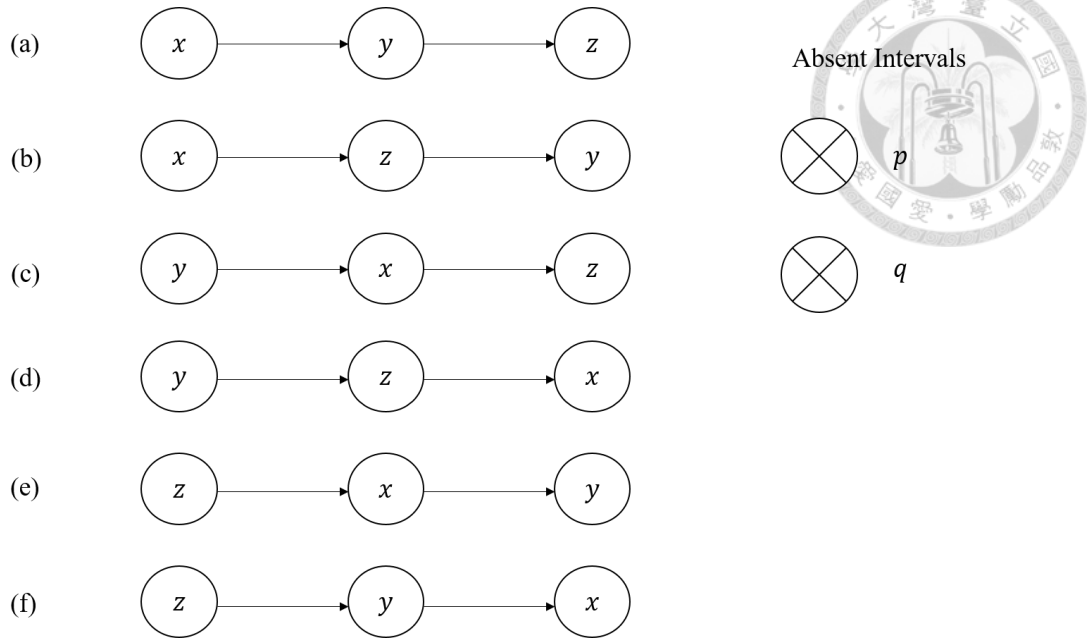


圖 2-3 序列變數 π 中時段變數可能的順序排列

CP Optimizer 提供許多限定運算子讓使用者限定序列變數中的時段變數的順位。設序列變數 π 轄有時段變數 x 和時段變數 y 。當時段變數 x 存在於排程解時，二元限定運算子 $first(\cdot, \cdot)$ 用以建立某時段變數在其序列變數的排頭順位限制式。因此限制式

$$first(\pi, x) \tag{2.7}$$

限定時段變數 x 排在序列變數 π 的排頭。三元限定運算子 $before(\cdot, \cdot, \cdot)$ 限定序列變數內轄的兩變數先後關係。因此若時段變數 x 和時段變數 y 都存在於排程解，限制式

$$before(\pi, x, y) \tag{2.8}$$

會設定在序列變數 π 中，時段變數 x 須排定在時段變數 y 之前，但二者之間可以有其他的時段變數。若要設定時段變數 y 緊鄰排在時段變數 x 後，可使用三元限定運算子 $prev(\cdot, \cdot, \cdot)$ 建立限制式

$$prev(\pi, x, y) \tag{2.9}$$

序列變數 π 內轄的時段變數內定允許交疊，即資源可同時間多份共享。但當資源在同一時間只能提供一個服務時，譬如一個加工作業完成後，機台才能加工



下個加工作業，必須限制它們不得交疊。因此可用一元限定運算子 $NoOverlap(\cdot)$ 建立限制式

$$NoOverlap(\pi)。(2.10)$$

CP Optimizer 會將 π 內時段變數排程結果相互分隔，出現的時段變數不會有交疊的情況。此限制式也稱為無重疊限制。

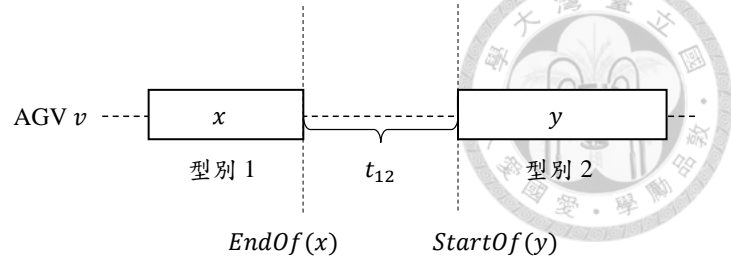
前面提及每個時段變數可設定一個整數的型別，用以代表其特定的屬性或標籤。在序列變數中，兩相鄰時段變數各有型別設定時，二型別的先後關聯可以代表特定的轉換關係。型別可用以限制時段變數間的時間。設時段變數有 n 種型別。令時段變數型別轉換的型別轉換間隔時間矩陣是 $\mathbf{M} = [t_{kk'}]_{n \times n}$ ，其中 $t_{kk'}$ 是型別 k 轉換到型別 k' 的型別轉換間隔時間。

限制式

$$NoOverlap(\pi, \mathbf{M}) (2.11)$$

可將序列變數 π 內的時段變數相互隔開，且時段變數與時段變數的時間會根據型別轉換，有最小的間隔時間。設在序列變數 π 中有時段變數 x 和時段變數 y ，且 x 的順位在 y 之前。再設 x 的型別是 1， y 的時段變數是 2。 $NoOverlap(\pi, \mathbf{M})$ 會設定時段變數 x 和時段變數 y 的排程至少須間隔 t_{12} 個時間單位，表示 $EndOf(x) + t_{12} < EndOf(y)$ 。如圖 2-4 所示，(a)是型別轉換間隔時間矩陣，(b)是時段變數 x 和時段變數 y 轉換間隔時間示意圖。此限制式可用在考量運輸時間的 FJSP 中，工件經過一段運輸時間才會抵達加工機台。可將時間矩陣設為工件在機台間的運輸時間，而加工作業的時段變數型別就是機台的編號。CP 求解引擎會排定加工作業的排程，並加入運輸時間在時段變數之間。

型別	1	2
1	t_{11}	t_{12}
2	t_{21}	t_{22}



(a) 型別轉換間隔時間矩陣

(b) 時段變數 x 和時段變數 y 的轉換間隔時間

圖 2-4 型別轉換間隔時間矩陣與時段變數轉換間隔時間示意圖

第3章 彈性零工生產與搬運排程問題及限制規劃求解模型

本章說明彈性零工生產與搬運排程問題，展示研擬的限制規劃求解模型。首先說明彈性製造系統內的產品、機台、和 AGV 間的關聯，描述它們排程的問題以及狀態轉移流程，並解釋排程內的時段和限制規劃模型的時段變數間的關聯。本章重點則展示本問題使用 CP Optimizer 的時段變數和序列變數所建立的限制規劃模型，依序解釋各限制式所代表的意義，並依照問題求解目標設定模型的目標函式。

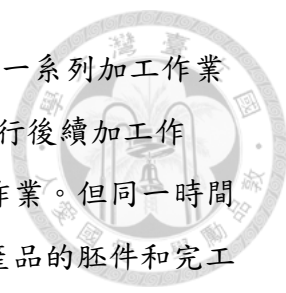
3.1 彈性零工生產與搬運排程問題

本節首先詳細定義彈性零工生產與搬運排程問題，介紹產品從胚件經一系列加工變成完工件的完整流程，過程須選擇加工機台和 AGV 搬運。接著介紹機台布置的種類、系統內物件狀態的轉移、和物件的數學模式，並說明 AGV 空車取貨和負載運送二種搬運作業與加工作業間的排程關係。

3.1.1. 問題描述與假設

典型 JSP 裡，每種產品的一序列加工作業都已設有特定的機台加工，加工時間也是固定的。FJSP 多了選擇機台的決策變數。因機台具多種加工功能，可執行不同的加工作業；不同的機台對同種加工作業的加工效能也不同，意即加工時間不同。

彈性零工生產與搬運排程問題源自於使用無人搬運車(以下簡稱 AGV)搬運工件的彈性製造系統。系統運作時由單元電腦(cell computer)整合機台和搬運設備，使用數值控制(Numerical Control)加工產品、安排機台加工的產品種類、和派遣搬運設備等，實現工廠自動化以及電腦整合製造生產模式。本研究探討如何有效地安排彈性製造系統內的機台和搬運設備的工作排程，期能在最短的時間內完成產品生產。



本彈性製造系統的產品是由胚件依製程規劃(process plan)經一系列加工作業(operation)加工完成。過程須在完成前面的加工作業後，才能進行後續加工作業。彈性製造系統的機台有高度的製造彈性，能執行多種加工作業。但同一時間內只能加工一個工件。除了機台外，系統內有倉儲設備，儲存產品的胚件和完工件，也可能存放在製品(work in process, WIP)。此外，產品的任一加工作業已知有多個效能不同的候選機台可執行，且各有不同的加工時間。產品的胚件一開始會存放在倉儲中心，由派遣的 AGV 前來搬運到第一個加工作業選取的機台。抵達機台的工件，由機台採先到先服務(First Come First Serve, FCFS)的方式當機台有空時即刻開始加工。工件在機台上加工完成後，再由派遣的 AGV 搬運到該工件下個加工作業所選擇的機台。如此依序執行直到產品的所有加工作業都完成後，由 AGV 將完工件運送回倉儲，完成生產流程。

本問題的本質可視為是有考量 AGV 派工和搬運時間的彈性零工式生產排程問題。求解時須派遣 AGV 執行各道次搬運作業，並從每個加工作業的候選機台中選擇加工機台，安排它們的加工順序，設定加工作業的開工和結束時間。問題求解即是在安排產品的加工作業、AGV 的搬運、和機台加工的排程。排程解不得違反各產品加工作業的先後限制。求解目標是求得一個排程有最短的產品最大完工時間。

3.1.2. 彈性零工生產與搬運排程問題的數學模式

本問題的彈性製造系統製造 n 種產品，產品集合是 $J = \{1, 2, \dots, n\}$ 。設產品 i 有 r_i 個序列加工作業， o_{ij} 是產品 i 的第 j 個加工作業； $j = 1, 2, \dots, r_i$ 。令產品 i 的加工作業集合是 $O_i = \{o_{i1}, o_{i2}, \dots, o_{ir_i}\}$ ， $i \in J$ 。產品的加工作業有前後順序限制，每個加工作業都要依序加工。因此，加工作業 o_{ij} 須在加工作業 $o_{i,j+1}$ 之前進行加工。例如： o_{21} 要在 o_{22} 之前完成加工，即產品 2 的第 1 個加工作業要在產品 2 的第 2 個加工作業之前完成。

設此系統內有 m 台機台，機台集合是 $M = \{1, 2, \dots, m\}$ 。機台負責執行產品的加工，但不能同時加工多個工件。機台對每個加工作業有不同的加工能力，所以每個加工作業在不同機台上加工所需的加工時間也不同。每個加工作業都有一組

已知的候選機台可以選擇，並已知對應的加工時間。令加工作業 o_{ij} 有 θ_{ij} 個候選機台可執行， w_{ijk} 是 o_{ij} 的第 k 台候選機台， $w_{ijk} \in M$ ， $k = 1, 2, \dots, \theta_{ij}$ 。設 o_{ij} 的加工候選機台集合是 $W_{ij} = \{w_{ij1}, w_{ij2}, \dots, w_{ij\theta_{ij}}\}$ ， $i \in J$ ， $j = 1, 2, \dots, r_i$ 。另設 \bar{p}_{ijk} 是 o_{ij} 在第 k 台候選機台上執行的時間，也就是機台 w_{ijk} 執行 o_{ij} 的加工時間。令 o_{ij} 在各個候選機台上對應的加工時間集合是 $\bar{P}_{ij} = \{\bar{p}_{ij1}, \bar{p}_{ij2}, \dots, \bar{p}_{ij\theta_{ij}}\}$ ， $\bar{p}_{ijk} > 0$ ， $i \in J$ ， $j = 1, 2, \dots, r_i$ ， $k = 1, 2, \dots, \theta_{ij}$ 。

系統內設有 g 台 AGV，AGV 集合是 $G = \{1, 2, \dots, g\}$ 。AGV 負責搬運工件到其加工作業所選的機台進行加工，或搬運完工件回倉儲存放。本問題的 AGV 是單一負載型(unit load)，每次只能運送單一工件。假設每台 AGV 的行走速度相同且固定。

系統內的倉儲中心和 m 台機台的位置已依據設施佈置和 AGV 路徑設計設定，位置固定不變。常見的機台佈置型態有直線型、S 型、U 型、灣型(bay)、以及星型(star)。圖 3-1 和圖 3-2 分別展示由倉儲中心、機台 1、 \dots 、機台 6 所組成的直線型佈置和 U 型佈置範例。倉儲中心和每台機台都設有兩個 AGV 停駐點，分別是 AGV 卸貨用的 D 點 (Delivery Point)和取貨用的 P 點 (Pickup Point)。AGV 在 D 點卸下工件，在 P 點上載工件。因此 AGV 運行時會行走在串接 P 點與 D 點或是 D 點與 P 點的軌道上。AGV 以固定的車速，沿著設計路徑在軌道上單向運行。當 AGV 從機台或倉儲中心 D 點出發到目的機台或倉儲中心 P 點取貨時車上沒有工件。AGV 到達 P 點上載工件後，沿軌道負載運送到目的機台 D 點，卸下工件。AGV 會在各機台和倉儲中心的站點間移動，因此將倉儲中心視為索引值為 0 的一個設施，再令包含倉儲中心及機台的設施集合是 $M = \{0, 1, 2, \dots, m\} = M \cup \{0\}$ 。

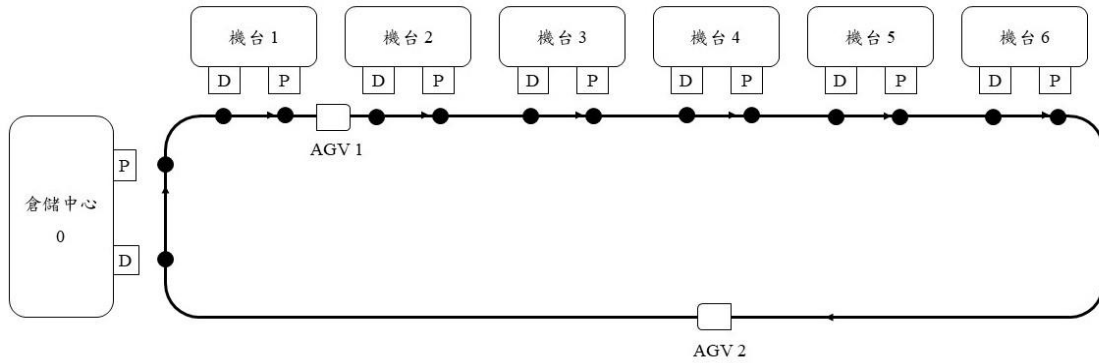


圖 3-1 機台直線型佈置的彈性製造系統

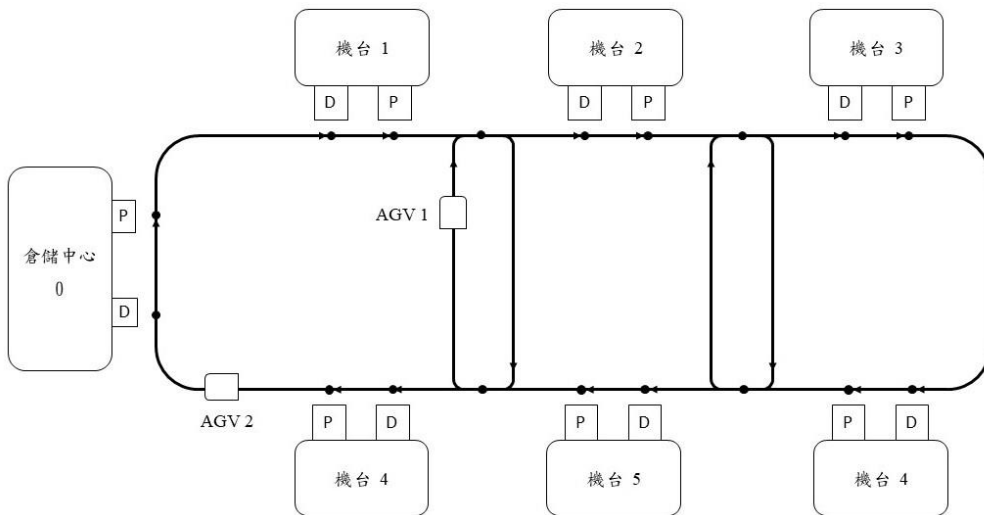



圖 3-2 機台 U 型佈置的彈性製造系統

已知 AGV 以固定的速度在軌道上運行，由設施 φ 運行到設施 φ' 間的時間可以由兩設施間的最短距離除以速度求得。因 AGV 在停駐點間有兩種搬運情境，運行時間分成兩種。第一種是從設施 φ 的 P 點負載運送到設施 φ' 的 D 點。令 P 點到 D 點的從至時間矩陣(from-to time matrix)是 $T = [t_{\varphi\varphi'}]_{(m+1)\times(m+1)}$, $\forall \varphi, \varphi' \in M$, $t_{\varphi\varphi'}$ 是 AGV 從設施 φ 的 P 點到設施 φ' 的 D 點的搬運時間。第二種是從設施 φ 的 D 點空車移動到設施 φ' 的 P 點。令 D 點到 P 點的從至時間矩陣是 $T' =$



$[t'_{\varphi\varphi'}]_{(m+1)\times(m+1)}$, $\forall\varphi, \varphi' \in M$, $t'_{\varphi\varphi'}$ 是 AGV 從設施 φ 的 D 點空車移動到設施 φ' 的 P 點的搬運時間。其中 t_{00} 是 AGV 從倉儲中心 P 點到倉儲中心 D 點的移動時間。但是倉儲中心 P 點是胚件出庫點，D 點是完工件入庫點，因此 AGV 不可能從倉儲中心 P 點搬胚件到倉儲中心 D 點。相對地 t'_{00} 則是 AGV 從倉儲中心 D 點到倉儲中心 P 點的移動時間。當胚件要開始進行加工時，停駐在倉儲中心 D 點被派遣的 AGV 會空車移動到倉儲中心 P 點上載胚件，移動時間是 t'_{00} 。假設 AGV 當下的位置在機台 2，經派遣要從機台 1 上載工件，搬運到機台 3，此時 AGV 會先從機台 2 的 D 點空車移動到機台 1 的 P 點，運行時間是 t'_{21} ；再從機台 1 的 P 點負載移動到機台 3 的 D 點，運行時間是 t_{13} 。

本問題假設每個站點的駐車容量都無上限，即倉儲中心和機台的 P 點和 D 點都可以讓所有 AGV 同時停駐。此外，機台上 P 點能存放的待送出工件和 D 點能存放的待加工件空間夠大且無容量限制。AGV 一次只能搬運一個工件，共有三種搬運作業。第一種是將產品胚件從倉儲中心搬運至第一個加工作業所選擇的機台，第二種是將加工後工件搬運至下一個加工作業所選擇的機台，第三種是將已完成所有加工作業的完工件搬運至倉儲中心儲存。

一開始產品的胚件由倉儲中心的 P 點出庫，且令所有 AGV 都停駐在倉儲中心的 D 點。AGV 經排程開始執行搬運作業時，先由倉儲中心 D 點空車移動到倉儲中心 P 點裝載胚件(耗時 t'_{00})，將胚件搬運到產品第一個加工作業選擇的機台上加工。因此進行任一個加工作業的排程時，須選擇一台 AGV 前往該加工作業的工件所在設施的 P 點上載工件。AGV 經指派後，由現駐的設施 D 點空車移動至該工件所在的設施 P 點。本問題不考慮工件上下載耗用的時間。此種由 D 到 P 的運行稱為「空車取貨」。AGV 從 P 點上載工件負載運送到下個加工作業所選擇的機台 D 點，此種由 P 到 D 的搬運稱為「負載運送」。工件卸載到機台上後，依照先到先服務的原則等待機台逐一加工。加工完成的工件會移至機台 P 點存放，等待系統派遣 AGV 運送到下一加工作業選擇的機台或倉儲中心。P 點依序存放加工完的工件，離開 P 點並非依照先到先離開的法則，而是依照他們下個加工作業的排程派來的 AGV 到達時間先後依序離開。各個產品的加工作業排程重

複以上的安排步驟，直到產品完成所有加工作業，由 AGV 送回倉儲中心 D 點存放。

令空車取貨作業 a_{ij} 是經派遣的 AGV 前往提取 o_{ij} 的工件的作業，負載運送作業 b_{ij} 是該 AGV 運送上載後的工件到執行 o_{ij} 的機台或是運送到倉儲中心存放； $j = 1, 2, \dots, r_i + 1$ 。若 o_{ij} 是產品的第一個加工作業， $j = 1$ ， a_{ij} 是某 AGV 空車移動至倉儲中心 P 點上載胚件， b_{ij} 是 AGV 從倉儲中心 P 點負載移動至執行 o_{i1} 的機台 D 點卸下胚件，也稱為產品 i 胚件的出庫作業。若 o_{ij} 不是產品的第一個加工作業， $1 < j \leq r_i + 1$ ， a_{ij} 是 AGV 空車移動至執行 $o_{i,j-1}$ 的機台 P 點上載工件， $b_{ij} = b_{i,r_i+1}$ 是 AGV 負載移動至執行 o_{ij} 的機台 D 點卸下工件。若 $j = r_i + 1$ ， a_{ij} 表示派遣的 AGV 空車移動到執行 o_{ir_i} 的機台 P 點上載完工件， b_{ij} 表示 AGV 負載移動到倉儲中心 D 點卸下產品 i 的完工件，也是產品 i 的入庫作業。因此產品 i 的空車取貨作業集合是 $A_i = \{a_{i1}, a_{i2}, \dots, a_{i,r_i+1}\}$ ， $i \in J$ 。產品 i 的負載運送作業集合是 $B_i = \{b_{i1}, b_{i2}, \dots, b_{i,r_i+1}\}$ ， $i \in J$ 。

3.1.3. 產品、機台、和 AGV 的狀態轉換

AGV 和機台初始狀態皆為「閒置」，等待搬運指令或工件抵達。產品胚件在倉儲中心 P 點等待出庫，其初始狀態是「等待 AGV」。進行加工作業排程時，須選定一可行的加工作業，再選定一 AGV 先執行空車取貨再執行負載運送作業。本研究派遣 AGV 有兩種排程安排模式，分別是「無等待」和「無閒置」模式。兩種模式會產生不同的 AGV 排程，有不同的狀態設定，但不會影響產品排程、機台排程、或總完工時間。本研究定義產品的可搬時間是工件加工作業完成的時間；AGV 的可行時間是完成一搬運任務的時間；機台的可行時間是完成一加工作業的時間。

在「無等待」模式中，AGV 上載工件時，工件已備妥(可搬)不會有等待狀態。因此若時間充裕，AGV 不會先執行「空車取貨」作業，先移動到機台等待工件加工完成。意即不會有等待工件上車的狀態。系統根據 AGV 當下的位置比較先移動至取貨點的抵達時間及工件的可搬時間，決定 AGV 執行「空車取貨」的開始時間。兩時間比較若時間充裕，抵達取貨點時須等待工件，則不會立即出

發。先將 AGV 轉成在原地「閒置」狀態，再緊接一段「空車取貨」時段，恰好在工作可搬的時間點抵達取貨點。若時間不充裕，則無「閒置」狀態，逕行執行「空車取貨」作業，抵達時是工件處於「等待 AGV」狀態可立即上載。取貨點可能是倉儲中心或是機台 P 點。此種模式下，AGV 抵達 P 點可以立即上載工件，完成「空車取貨」並緊接著執行「負載運送」作業將工件運送至下個加工作業選擇的機台。例如：工件在第 100 秒完成加工，派遣的 AGV 前往 P 點的搬運時間需 30 秒，則系統會設定 AGV 「空車取貨」作業的出發時間是第 70 秒。如圖 3-3 所示，因時間充裕，AGV 先進入「閒置」狀態，再轉為「空車取貨」狀態，在產品可搬的時間點抵達。反之，若時間不充裕，AGV 不會有「閒置」狀態。AGV 立即前往取貨點上載工件，進入「負載運送」狀態。工件上載 AGV 後，產品狀態從「加工中」轉為「轉運中」。

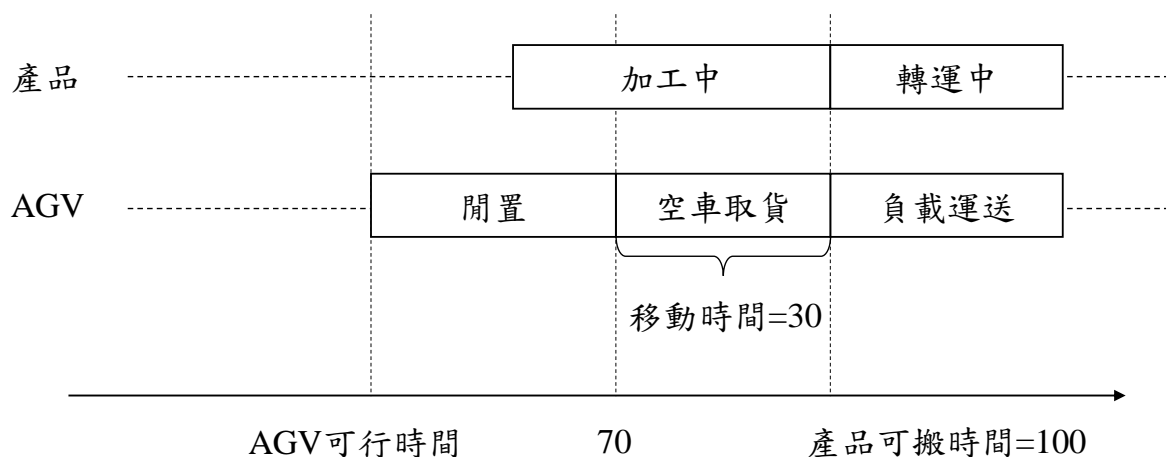


圖 3-3 AGV 無等待派遣模式

相反地，在「無閒置」模式中，AGV 先執行空車取貨作業，移動至 P 點。因此沒有閒置的狀態。但時間充裕時，會在 P 點等待工件。如圖 3-4 所示，AGV 會先進入「空車取貨」狀態下到 P 點。若 AGV 抵達 P 點時，工件尚未加工完成，AGV 狀態轉為「等待工件」，等待工件加工完成時，上載工件，進入「負載運送」狀態。產品在工件上載時轉入「轉運中」狀態。例如：工件預定在第 100 秒加工完成，派遣的 AGV 可用時間是第 50 秒，移動至工件取貨點的搬運時間是 20 秒，因此 AGV 會在第 70 秒抵達取貨點，在取貨點等待 30 秒後，上載工件開始運送。

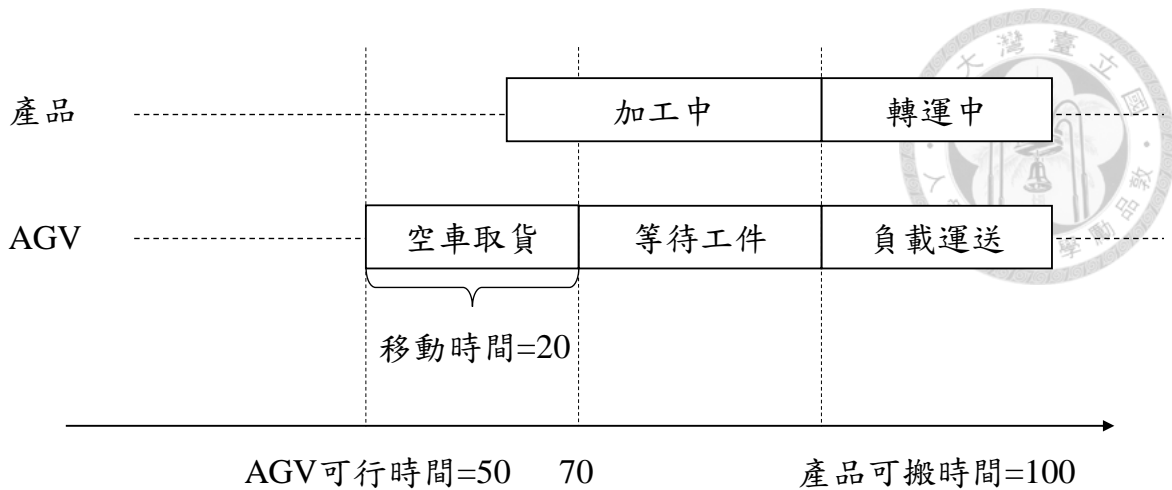


圖 3-4 AGV 無閒置派遣模式

比較圖 3-3 及圖 3-4，二種模式的差異只有 AGV 的狀態，產品的時程不變。相對地當時間緊迫，不管何種模式因工件加工完成，AGV 尚未抵達 P 點，也就是 AGV 抵達 P 點的時間晚於產品可搬時間，此時產品的狀態會進入「等待 AGV」，如圖 3-5 所示。此種緊迫情境下，無閒置模式的 AGV 不會有「等待工件」狀態，無等待模式的 AGV 也不會有「閒置」狀態。

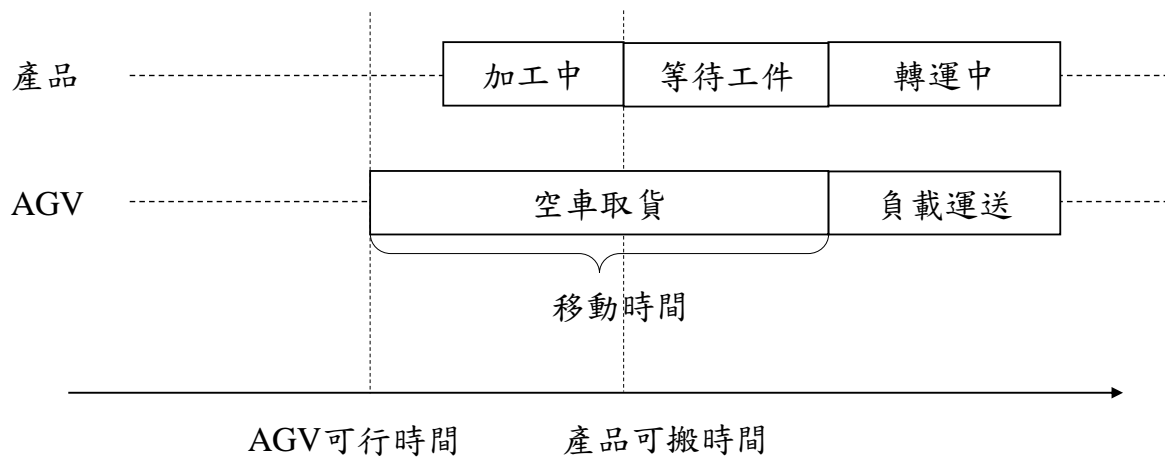


圖 3-5 產品等待 AGV 示意圖

當 AGV 負載運送抵達 D 點時，卸載工件完成一個搬運任務，更新 AGV 的可行時間及最新地點，等候下一個搬運指令。當工件卸載後，若機台的狀態是「忙碌」，表示機台正在加工其他產品，此工件會進入「等待機台」狀態，加入

待加工的工作件佇列。直到機台加工完前面的佇列工作，開始加工該工作件時，機台進入「忙碌」，而工作件進入「加工中」狀態。

反之，若 AGV 抵達 D 點卸貨時，機台處於「閒置」狀態，機台會立即加工該卸載的工作件，狀態則轉成「忙碌」。工作件的狀態則變成「加工中」。圖 3-6 至圖 3-8 分別展示工作件(產品)、AGV 和機台的狀態轉換。狀態轉換圖採用 UML (統一塑模語言) 格式，其中圓角矩形表示狀態，實心黑色的圓型是初始狀態，被空心圓形包覆的實心黑色圓型是結束狀態，有向弧表示事件，菱形是決策點，中括號則是條件狀態。

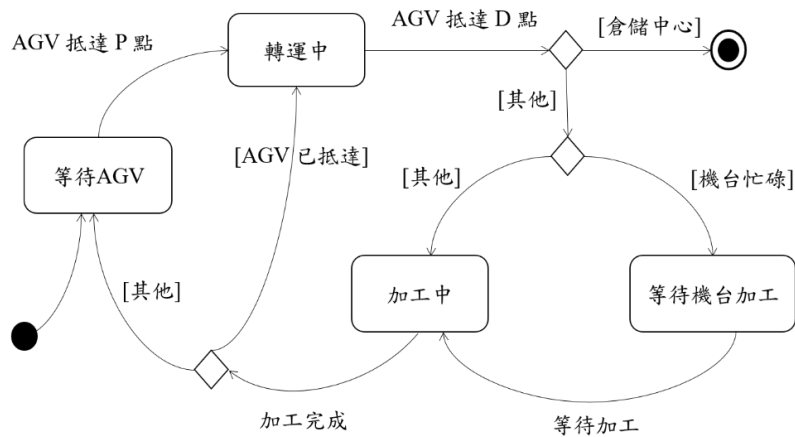


圖 3-6 產品的狀態轉換示意圖

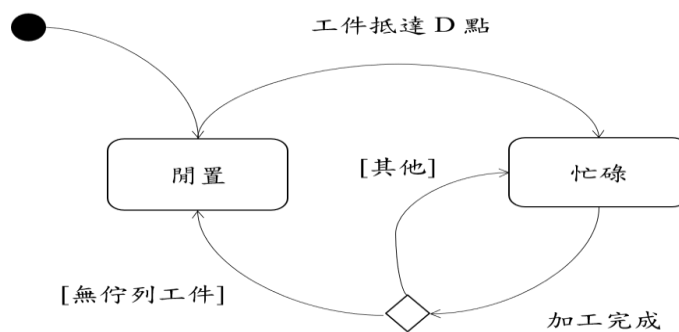


圖 3-7 機台的狀態轉換示意圖

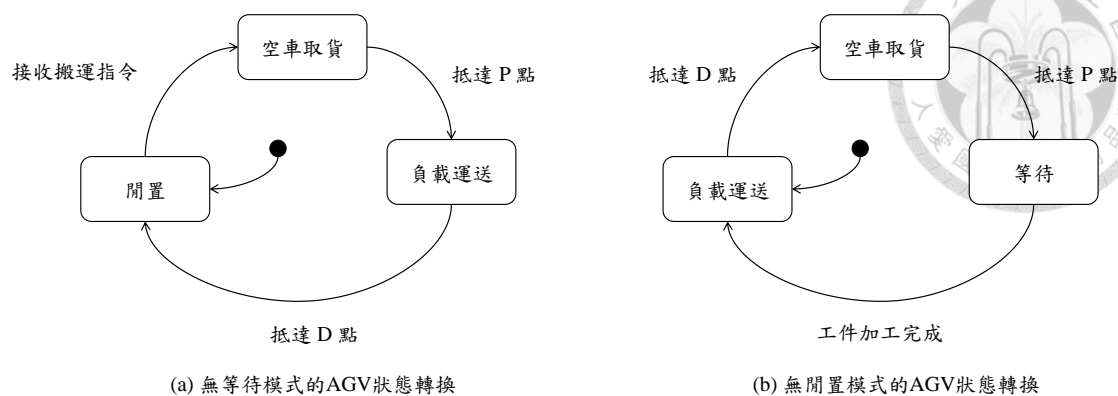



圖 3-8 AGV 兩種排程模式的狀態轉換示意圖

根據前面所述，本問題的求解目標是最小化產品的最大完工時間，即完工件入庫時間。本問題的排程需考量各產品加工作業的處理順序、機台加工、和 AGV 搬運。產品的工件由 AGV 搬運至機台上加工，因此各產品的排程和 AGV 與機台的排程緊密相關。每種產品的每個加工作業排程都會受到派遣的 AGV 當下位置以及選定的機台加工忙碌狀態影響，且需在依循加工作業前後順序的限制下，排定每個加工作業的開工時間。當所有產品完成系列的加工後入庫，才完成本問題的生產排程解。

總的來說，本問題是須安排 AGV 運送以及選定加工機台的彈性零工式生產排程問題。文獻探討顯示只有少數研究提出彈性零工生產排程問題的整數規劃模型，未有對本複雜問題研擬的求解模型。此外文獻提出的 FJSP 數學規劃模型因複雜度太高，求解效能差，無法迎合實務應用。如前所述本問題除了機台的選擇外，還需考量 AGV 的派工及工件的搬運，才符合現今高度自動化的工廠實務。因此加入 AGV 的搬運排程使本問題的複雜度遠比彈性零工生產問題更高，且問題的限制也更多。若要建構本問題的混整數規劃求解模型，須使用大量的 0-1 決策變數設定某加工作業選擇某候選機台與否，以及某 AGV 選擇與否。會有複雜無數的單選變數及衍生無數的限制條件，複雜化整個求解模型。此外，每個加工作業都需先經過 AGV 的空車取貨和負載運送後，才能開始安排加工排程。加工作業選定機台後，AGV 的搬運路線也會被限縮。AGV 的搬運作業須知道取貨和卸貨的設施；也須確認搬運路徑是從 D 點到 P 點或是從 P 點到 D 點，以設定運行時間。此外，產品的各加工作業的加工排程和 AGV 負載運送的排程不能重



疊，且不能違反前後順序限制。產品、機台、和 AGV 的排程相互影響，難以用線性的整數規劃建構本問題的數學模型。因此，部分文獻研擬各種啟發式演算法求解考量搬運的彈性零工式生產排程問題。啟發式演算法雖然可以快速求出不錯的解，但是這些演算法有演化速度慢或是容易陷入區域最佳解的缺點。操作者必須要各個標竿問題有適度的分析，才能針對不同的問題調整演算法的參數，發揮啟發式演算的效能。

另一方面，「限制規劃」可以突破傳統數學規劃的約束，將難以線性化的限制條件用容易建置的陳述式來建模。除了能求解限制條件滿足問題外，也能求解一般有限制的優化問題。經精心規劃建立的 CP 求解模型常具泛用性，可以求解同型態的標竿問題；意即問題模式化得宜時不需針對不同數據的問題調整求解模型的參數。也就是說應用於求解本研究問題時，AGV、產品、和機台的數量不同或加工作業的候選機台集合變更、以及 AGV 行走各機台間的從至時間變更都可以使用同一個限制規劃模型求解。下節將介紹時段與時段變數，讓讀者了解兩者的關聯關係，接著才會說明本研究研擬的限制規劃模型。

3.2 時段與時段變數

若使用整數規劃求解彈性零工生產與搬運排程問題，一個加工時間未定的加工作業的排程決策變數有開始時間和結束時間二個實數變數。而若使用限制規劃，即可以一個區段取代。本問題的加工作業在執行時，會占用選定的機台資源一段時間。工件轉運需要的空車取貨作業和負載運送作業則是占用 AGV 資源的一段時間。因此，本問題的資源使用作業都可以使用 CP 的區段變數代表並以其對應的時間區段展示排程解，每個時間區段可設定其依屬實務的狀態標籤 (label) 和屬性(attribute)，以展示排程內容。因區段用於本問題的排程解是執行的一段時間，以下簡稱為「時段」。時段的標籤也代表所屬的物件(AGV、產品、機台)在該時段內處於某一個狀態。當事件發生時該物件的狀態結束，接續新的時段轉入下個狀態。

本研究研擬的 AGV 排程採用「無閒置」的排程模式。AGV 完成一搬運任務時即停駐於卸貨 D 點。安排新搬運指令時，AGV 立即執行空車取貨作業，抵達

工件所在的設施 P 點。若工件尚未加工完成，AGV 轉入「等待」狀態，在取貨點等待工件加工完成。每個加工作業的執行前都需要 AGV 先執行空車取貨作業再接續負載運送作業，之後才能安排加工作業的排程。也就是說，加工作業 o_{ij} 前會有一段空車取貨作業 a_{ij} 和負載運送作業 b_{ij} 。圖 3-9 展示 o_{ij} 、 a_{ij} 、和 b_{ij} 分別在機台、產品、和 AGV 上的排程關係。圖中的區塊是某作業的排程，不同顏色的區塊代表不同的時段，也表示不同的狀態。藍色的時段代表加工作業，也表示產品「加工中」。紅色的時段代表負載運送作業，也表示 AGV「負載運送」。綠色的時段代表空車取貨作業，也表示 AGV「空車取貨」。

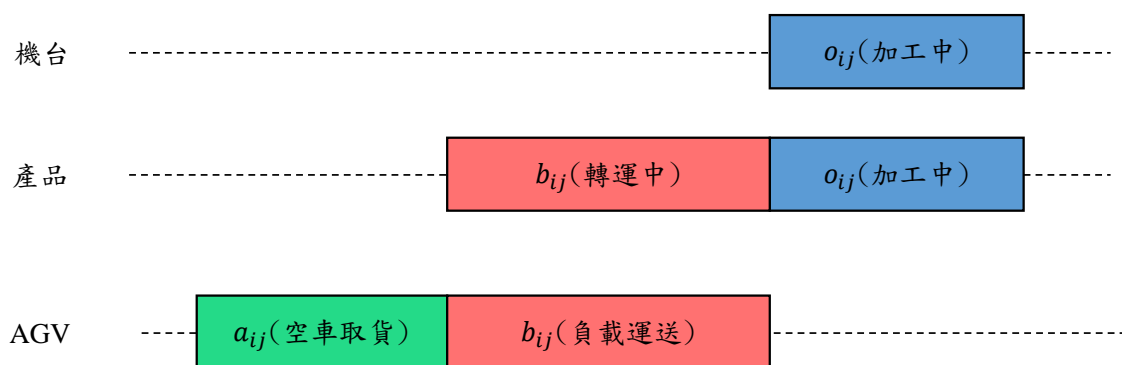


圖 3-9 加工作業 o_{ij} 、空車取貨作業 a_{ij} 、和負載運送作業 b_{ij} 的時段及狀態排程關係

CP Optimizer 提供區段變數作為決策變數以模式化排程問題，概念與數學的區段相似。因 CP 的區段變數用於本問題的排程解是資源使用的一段時間，本研究具體稱它為「時段變數」。本研究研擬的限制規劃模型會以時段變數作為加工作業和負載運送作業的排程解，但是不使用時段變數表示空車取貨作業的排程解。原因會在後面詳加描述。

簡而言之，在本問題中，時段表示作業的排程所對應的時間區段。而時段變數是 CP Optimizer 提供使用者建立限制規劃模型的決策變數，表示作業的排程解。時段變數由開始時間、結束時間、作業時間、此時段是否是可選的布林值、以及型態所組成。CP 求解結果可選擇的時段若未被選取，不會出現在排程解中。



CP 求解引擎會設定加工作業和負載運送作業的排程，其排程就是時段變數對應的時間區段。空車取貨作業的排程不是以時段變數表示，其排程會以其他方式求得，方法也會在後面詳加描述。

3.3 限制規劃求解模型

限制規劃的求解模型沒有固定的表達形式，不同的 CP 求解引擎會有各自的模型格式。如第二章所述，IBM ILOG® CPLEX 的 CP Optimizer 建模套件是用以建立 CP 模型並使用限制規劃的技術求解組合優化問題和限制優化問題。CP Optimizer 函式庫包含建構問題使用的變數、函式、運算式、和限制式等。本節將展示本研究研擬的彈性零工生產與搬運排程問題的限制規劃求解模型。模型以 CP Optimizer 的時段變數和序列變數做為主要的決策變數，以求解彈性零工生產與搬運排程問題的最佳排程。求解目標分為(1)最小化產品最大完工時間和(2)最小化所有產品完工時間總和。本限制規劃模型限制式較複雜，為容易說明，依限制式限定的對象分為產品排程前後順序限制、負載運送作業相關限制、和空車取貨作業相關限制三個部分解說。以下將依模型架構功能，逐一引用需要的變數、展示運算式的表示法、以及定義各種限制條件。

3.3.1. 時段變數和序列變數

本問題的加工作業使用機台資源加工一段時間，因此可以時段變數表示加工作業的排程結果。本研究定義 o_{ij} 是加工作業 o_{ij} 的時段變數， o_{ij} 的排程開始時間求解後是 $StartOf(o_{ij})$ 、結束時間是 $EndOf(o_{ij})$ ，而兩者的間距即是在選取的某候選機台上加工耗用的時間。

AGV 的搬運作業有兩種，分別是空車取貨作業和負載運送作業。已知產品 i 有 r_i 個加工作業，若 $1 \leq j \leq r_i$ ， o_{ij} 加工前，需由派遣的 AGV 先執行空車取貨作業。AGV 從它當下的停駐點，也就是上次運送的工件卸貨的機台或是倉儲中心 D 點，空車移動到 o_{ij} 的取貨機台或倉儲中心 P 點，此搬運作業是空車取貨作業 a_{ij} 。AGV 上載工件後，再執行負載運送作業，將工件由機台或倉儲中心 P 點搬運到

o_{ij} 選定的機台或倉儲中心 D 點，此搬運作業是負載運送作業 b_{ij} 。若 $j = r_i + 1$ ， a_{ij} 是派遣的 AGV 空車移動到執行 o_{ij} 的機台 P 點；而 b_{ij} 是該的 AGV 將產品 i 已完
成加工作業 o_{ij} 的完工件上載，負載運送到倉儲中心 D 點卸載儲存。

負載運送作業是使用 AGV 搬運資源一段時間，因此 b_{ij} 的排程也以時段變數表示。本求解模型定義 b_{ij} 是 b_{ij} 的時段變數， b_{ij} 內有搬運作業的開始時間和結束時間，兩者的間距即是 AGV 由 P 點負載運行到 D 點的移動時間。空車取貨作業也是使用 AGV 搬運資源一段時間，但本研究不以時段變數表示 a_{ij} 的排程解。原因會在本章後段詳述。

如第二章所述，CP Optimizer 內的序列變數可表示使用同一資源的時段變數的序列排程，會由 CP 求解引擎根據序列變數的限制排內轄的時段變數的順序，也就是這些變數使用該資源的順位。

本問題內的機台是提供加工服務的資源，機台須排定其執行加工作業的順序，因此每台機台需有其對應的序列變數。本模型定義機台 r 的序列變數是 D_r ， D_r 內轄有所有可能在機台 r 執行的加工作業所對應的時段變數。此外，本問題的 AGV 是執行搬運任務的資源，也須排定它負責的搬運任務的順序。因此令 AGV v 的序列變數是 S_v ， S_v 管轄的是所有可能由 AGV v 執行的負載運送作業的時段變數。這些變數會在後續定義，屆時再舉例解說。

3.3.2. 限制條件

產品排程前後順序限制

產品排程前後限制是在規範產品排程結果中加工作業和負載運送作業的前後順序。如問題的描述，每種產品須經系列的加工作業加工始成完工件。產品的加工作業有前後順序限制，須完成前面的加工作業後，並由派遣的 AGV 搬運工件到下個加工作業選定的機台，才能執行下個加工作業。因此各產品的排程結果中，加工作業的前後都有個負載運送作業須完成，且運送作業和加工作業對應的時段不得重疊。

圖 3-10 是 CP 求解引擎排定產品*i*的排程結果示意圖，內有加工作業和負載運送作業的時段變數。如圖所示，產品*i*的排程結果顯示，負載運送作業的時段變數 **b_{ij}** 一定會排在加工作業的時段變數 **o_{ij}** 之前， **$b_{i,j+1}$** 也一定會排在 **o_{ij}** 之後。

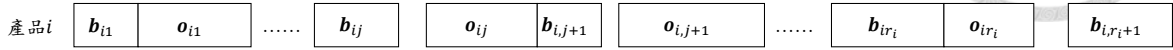


圖 3-10 產品*i*的加工作業和負載運送作業的時段變數排程關聯關係

派遣的 AGV 執行負載運送作業 **b_{ij}** ，將工件搬運到 **o_{ij}** 選擇的機台後， **o_{ij}** 才可開始執行。意即就產品*i*而言， **b_{ij}** 的時段變數 **b_{ij}** 須排在 **o_{ij}** 的時段變數 **o_{ij}** 之前，因此本模型添加限制式

$$EndBeforeStart(b_{ij}, o_{ij}), \forall i \in J, j = 1, 2, \dots, r_i. \quad (3.1)$$

o_{ij} 加工完成後，派遣的 AGV 才可執行下一個負載運送作業 **$b_{i,j+1}$** 。因此 **$b_{i,j+1}$** 須排在 **o_{ij}** 之後，添加限制式

$$EndBeforeStart(o_{ij}, b_{i,j+1}), \forall i \in J, j = 1, 2, \dots, r_i. \quad (3.2)$$

加工作業可選擇的時段變數及相關限制

加工作業須從候選機台選擇一個加工機台，因此得建立相關限制式讓加工作業就候選的機台擇一批配。此外也須設定機台同一時間只能執行一個加工作業。機台是提供加工服務的資源，CP 求解引擎會依限制條件排定各加工作業的執行以及各加工作業在機台上執行的順位。

因加工作業在不同的機台上加工的時間不同，其排程結果不會確定時段長度，但長度應是由候選的機台加工時間擇一匹配。因此，加工作業在各候選機台上加工的時段都需以時段變數代表，且設這些變數是可選擇的，由加工作業的時段變數擇一匹配。CP 引擎求解時會從候選變數擇一匹配，只有被選中的時段變數會出現在排程解。



已知加工作業 o_{ij} 有 θ_{ij} 個候選機台，時段變數 o_{ij} 需有 θ_{ij} 個可選擇匹配的時段變數以設定它的間距。未排定 o_{ij} 的排程前， o_{ij} 的加工時間未知，因此時段變數 o_{ij} 的間距也未定。

本研究的 CP 求解模型針對 o_{ij} 的第 k 台候選機台定義時段變數 \bar{o}_{ijk} ，因此 o_{ij} 在候選機台加工對應的時段變數集合是 $\mathbf{O}_{ij} = \{\bar{o}_{ijk} \mid k = 1, 2, \dots, \theta_{ij}\}, \forall i \in J, j = 1, 2, \dots, r_i$ 。 \bar{o}_{ijk} 的間距已先設定成常數，就是 o_{ij} 在第 k 台候選機台上加工耗用的時間 \bar{p}_{ijk} 。因此此模型須添加限制式

$$\text{LengthOf}(\bar{o}_{ijk}) = \bar{p}_{ijk}。 \quad (3.3)$$

o_{ij} 只能選擇一個候選機台執行。因此設定 \bar{o}_{ijk} 是可選的時段變數，供 o_{ij} 擇一匹配，限制式為

$$\text{SetOptional}(\bar{o}_{ijk}), \forall i \in J, j = 1, 2, \dots, r_i, k = 1, 2, \dots, \theta_{ij}。 \quad (3.4)$$

\mathbf{O}_{ij} 內的時段變數即是 o_{ij} 的擇一匹配變數，求解時由 CP 引擎選擇一個與它匹配。因此設定加工作業的擇一匹配限制式

$$\text{Alternative}(o_{ij}, \mathbf{O}_{ij}), \forall i \in J, j = 1, 2, \dots, r_i。 \quad (3.5)$$

此限制式設定 o_{ij} 從 \mathbf{O}_{ij} 中選擇一個候選時段變數匹配。意即 \mathbf{O}_{ij} 的集合中只會有一個時段變數存在於排程解。同時， o_{ij} 的起始與結束時間都會與匹配的時段變數相同。舉例來說，假設 o_{21} 選擇在第 3 台候選機台加工，則 \bar{o}_{213} 是 \mathbf{O}_{21} 中唯一存在於排程解的時段變數。也就是說，主要的時段變數 o_{21} 從 \mathbf{O}_{21} 中選擇 \bar{o}_{213} 匹配，因此 o_{21} 的開始時間、結束時間、間距都會與 \bar{o}_{213} 相同。

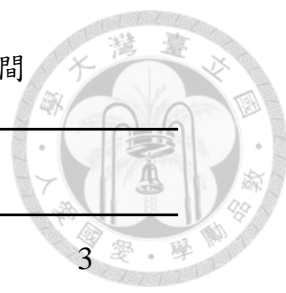
假設 w_{ijk} 是 o_{ij} 的第 k 台候選機台。機台 r 的序列變數 D_r 內轄機台 r 的時段變數，由 CP 求解時排定有出現在排程解的時段變數使用機台 r 的順位，因此 $D_r \equiv \{\bar{o}_{ijk} \mid w_{ijk} = r, \forall i \in J; j = 1, 2, \dots, r_i; k = 1, 2, \dots, \theta_{ij}\}$ 。 D_r 內的時段變數都是可選擇的，求解後不一定存在於最終的排程解。存在的時段會由 CP 求解引擎排定順位，在機台 r 上執行的加工順序。

表 3-1 是一個小型的彈性零工式生產問題範例，內有 3 種產品和 3 台機台。表 3-2 則是 AGV 在三台機台和倉儲中心間的搬運時間。 $t_{\rho\rho'}$ 是設施 ρ 的 P 點到設施 ρ' 的 D 點的移動時間， $t'_{\rho\rho'}$ 則是設施 ρ 的 D 點到設施 ρ' 的 P 點的移動時間。 o_{11} 的第 1 個候選機台、 o_{21} 的第 1 個候選機台、和 o_{31} 第 1 個候選機台都是機台 1，因此機台 1 的序列變數 $D_1 = \{\bar{o}_{111}, \bar{o}_{211}, \bar{o}_{311}\}$ ，由這三個加工作業可選的時段變數所組成。因為 \bar{o}_{111} 、 \bar{o}_{211} 、和 \bar{o}_{311} 都是選擇性的，排程最終結果都不一定存在。當三個時段變數都不存在時，表示機台 1 沒有執行任何加工作業。

表 3-1 三種產品和三台機台的範例問題

產品	加工作業	機台		
		1	2	3
1	o_{11}	10	20	-
	o_{12}	-	80	50
	o_{13}	-	40	70
2	o_{21}	60	50	-
	o_{22}	-	80	60
3	o_{31}	30	-	-
	o_{32}	-	20	40

表 3-2 AGV 在三台機台和倉儲中心間的搬運時間



D到P/P到D	倉儲中心		機台		
	0	1	2	3	
倉儲中心 0	1/1	1/5	8/10	15/25	
1	10/15	3/3	20/10	15/20	
機台 2	15/20	20/20	25/30	20/10	
3	5/10	10/20	15/25	3/3	

機台同一時間下只能執行一個加工作業。也就是說，在機台 r 上有出現的加工作業排程對應的時段不可重疊，才是合理的排程。因此設定 D_r 內轄的時段變數須無重疊，使 D_r 中的所有時段變數相互隔開，限制式為

$$NoOverlap(D_r), r = 1, 2, \dots, m. \quad (3.6)$$

負載運送作業可選擇的時段變數及相關限制

一個加工作業的負載運送作業其取貨和卸貨機台以及負責的 AGV 只能在求解後才確定，模型必須限定這些選擇是在可行的範圍內。本問題的加工作業在機台上執行前，須由派遣的 AGV 先空車取貨再執行負載運送作業，將工件運送至選擇的機台。因此就產品的排程而言，兩相鄰的加工作業之間一定會有段負載運送作業。此外，產品的第一個加工作業執行前有個負載運送作業將胚件出庫送至選擇的機台，最後一個加工作業執行完成後也有個負載運送作業將完工件入庫存放。

負載運送作業的搬運途徑會與相鄰的兩加工作業選擇的機台相關，此二機台分別是 AGV 負載運送的起點和目的地。不同的 AGV 選擇以及不同的搬運途徑須在排程結果上視為不同的時段，因此這些時段須設成時段變數。CP 求解引擎會選定一台 AGV 和一種搬運途徑組合，意即從候選的時段變數中擇一匹配。

負載運送作業 b_{ij} 是從執行 $o_{i,j-1}$ 的機台將工件搬運到 o_{ij} 選擇的機台上。 b_{ij} 的時段變數 b_{ij} 的間距即是 AGV 由 P 點負載運行至 D 點的移動時間，因此間距會根據 AGV 出發的設施和抵達的設施而不同。若 o_{ij} 選擇在第 k' 個候選機台上加工，即機台 $w_{ijk'}$ 上加工，且 $o_{i,j-1}$ 選擇在第 k 個候選機台上加工，即機台 $w_{i,j-1,k}$ 上加工。令 $\beta = w_{ijk'}$ 且 $\alpha = w_{i,j-1,k}$ ，則 AGV v 從 α 的 P 點移動至 β 的 D 點的負載運送作業 b_{ij} 可定義其時段變數是 $\bar{b}_{ijvkk'}$ 。

因 o_{ij} 有 θ_{ij} 個候選機台， $o_{i,j-1}$ 有 $\theta_{i,j-1}$ 個候選機台，所以 b_{ij} 的可能的搬運途徑數量有 $\mu_{ij} = \theta_{ij} \cdot \theta_{i,j-1}$ 個。其中若 $j = 1$ ， b_{ij} 是胚件的出庫作業，AGV 都從倉儲中心 P 點出發， $\mu_{i1} = \theta_{i1}$ 。 b_{i,r_i+1} 是產品完工件的入庫作業，AGV 從產品 i 最後一個加工作業 o_{ir_i} 選定的機台出發，運送完工件回到倉儲中心 D 點，有 θ_{ir_i} 個途徑， $\mu_{i,r_i+1} = \theta_{ir_i}$ 。再因系統內有 g 台 AGV，每台 AGV 都有可能執行 b_{ij} ，因此主要的時段變數 b_{ij} 有 $\mu_{ij} \cdot g$ 個可選的時段變數擇一匹配。

本模型定義時段變數 b_{ij} 對應的擇一匹配時段變數集合是 $B_{ij} = \{\bar{b}_{ijvkk'} \mid v = 1, 2, \dots, g; k = \begin{cases} 1, & \text{if } j = 1 \\ 1, 2, \dots, \theta_{i,j-1}, & \text{otherwise} \end{cases}; k' = \begin{cases} 1, & \text{if } j = r_i + 1 \\ 1, 2, \dots, \theta_{ij}, & \text{otherwise} \end{cases}\}$, $\forall i \in J, j = 1, 2, \dots, r_i + 1$ 。 $\bar{b}_{ijvkk'}$ 的間距是 AGV 的負載運送的移動時間。因此設定即限制式

$$\begin{aligned} \text{LengthOf}(\bar{b}_{ijvkk'}) &= t_{\alpha\beta}, \\ \alpha &= \begin{cases} 0, & \text{if } j = 1 \\ w_{i,j-1,k}, & \text{otherwise} \end{cases}; \beta = \begin{cases} 0, & \text{if } j = r_i + 1 \\ w_{ijk'}, & \text{otherwise} \end{cases}. \end{aligned} \quad (3.7)$$

b_{ij} 只能由候選的搬運途徑及 AGV 組合選擇一個途徑。因此設定 $\bar{b}_{ijvkk'}$ 時段變數是選擇性的，供 b_{ij} 擇一匹配。本模型設定選擇性的時段變數，限制式

$$\text{SetOptional}(\bar{b}_{ijvkk'}). \quad (3.8)$$

時段變數 b_{ij} 在設定 b_{ij} 的排程會由 B_{ij} 內的時段擇一匹配，因此求解模型設定負載運送作業的擇一匹配限制式

$$\text{Alternative}(\mathbf{b}_{ij}, \mathbf{B}_{ij}), \forall i \in J, j = 1, 2, \dots, r_i + 1. \quad (3.9)$$

以表 3-1 和表 3-2 的範例為例， o_{21} 有兩個候選機台，分別是機台 1 和機台 2。 o_{22} 有兩個候選機台，分別是機台 2 和機台 3。AGV 執行負載運送作業 b_{22} 時有 $\mu_{22} = 2 \times 2 = 4$ 種路線，分別是從機台 1 的 P 點到機台 2 的 D 點(耗時 $t_{12} = 20$)、從機台 1 的 P 點到機台 3 的 D 點(耗時 $t_{13} = 15$)、從機台 2 的 P 點到機台 2 的 D 點(耗時 $t_{22} = 25$)、和從機台 2 的 P 點到機台 3 的 D 點(耗時 $t_{23} = 20$)。如圖 3-11 所示， o_{21} 和 o_{22} 各有兩個候選的時段變數；而 b_{22} 則有 $\mu_{22} \times 3 = 12$ 個候選的時段變數，分別由 AGV 1、AGV 2、和 AGV 3 在機台 1/2 及機台 2/3 間搬運。圖中以虛線繪製的時段表示選擇性的時段變數，塗灰色的時段是出現在最終解的時段變數。時段變數的間距是 AGV 負載運送的移動時間。加工作業須選定一個機台加工且其負載運送作業須選定一個 AGV 在選定的機台間運送工件，因此 CP 求解時非選擇性的時段變數 o_{21} 、 o_{22} 、和 b_{22} 都會從各自的擇一匹配時段變數集合選擇一個時段變數匹配。此範例顯示時段變數 o_{21} 選擇時段變數 \bar{o}_{211} 匹配，即在機台 1 加工；時段變數則 o_{22} 選擇時段變數 \bar{o}_{222} 匹配，即在機台 3 加工；時段變數 b_{22} 則選擇時段變數 \bar{b}_{22112} 匹配，即由 AGV 1 搬運。

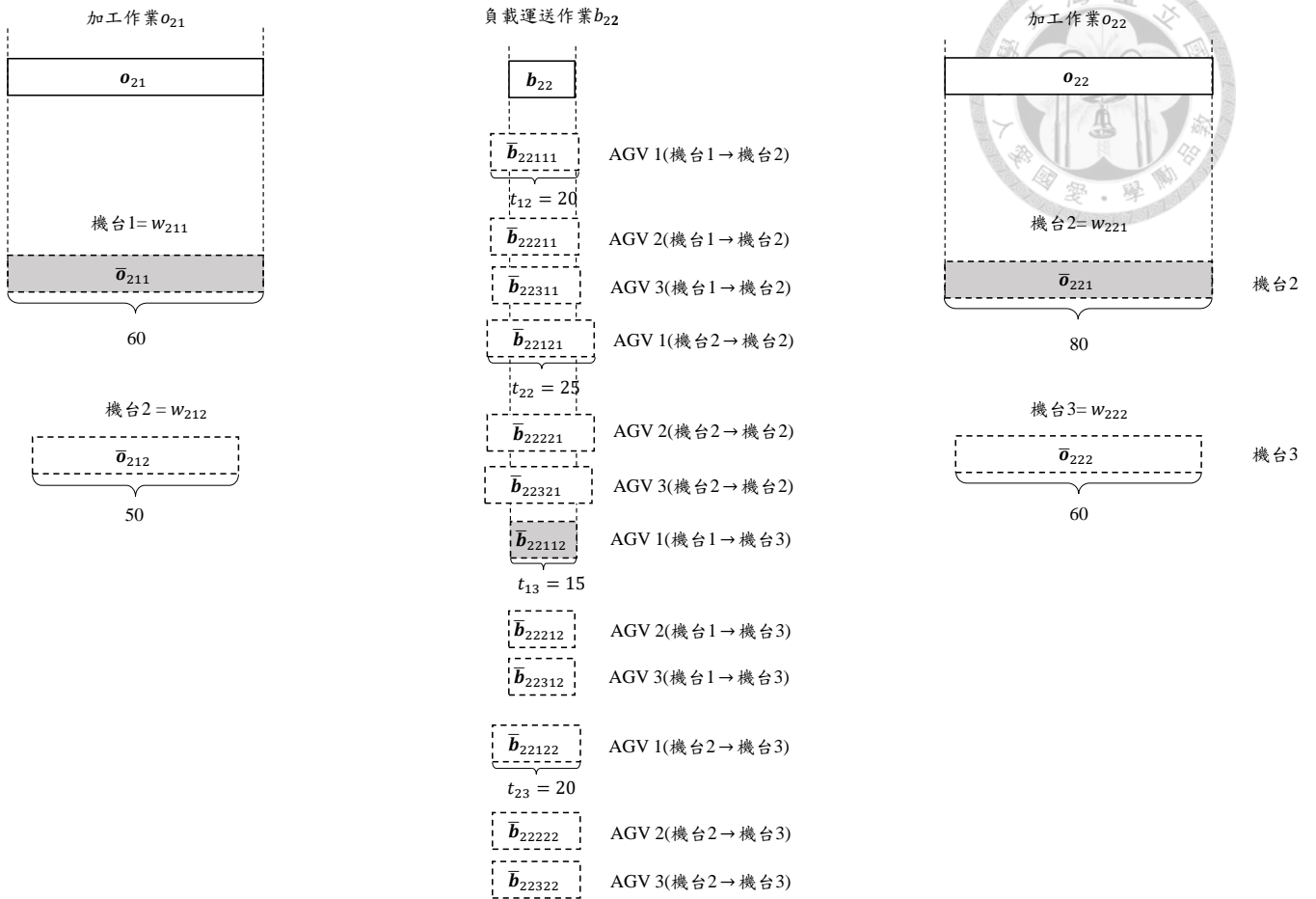


圖 3-11 加工作業和負載運送作業的候選時段變數設定範例以及擇一匹配示意圖

如前所述，AGV 執行 b_{ij} 是從執行 $o_{i,j-1}$ 的機台 P 點移動到 o_{ij} 選擇的機台 D 點。因此， b_{ij} 選擇的搬運途徑須考量 $o_{i,j-1}$ 和 o_{ij} 選擇的機台。再以圖 3-11 為例，因 b_{22} 選擇 \bar{b}_{22112} 匹配，表示 b_{22} 從機台 1 移動到機台 3。因此，加工作業 o_{21} 也必須選擇在機台 1 加工，而加工作業 o_{22} 也必須選擇在機台 3 加工，才是合理的排程結果。意即若 \bar{b}_{22112} 存在，則 \bar{o}_{211} 和 \bar{o}_{222} 必是存在的時段變數，也分別是 o_{21} 和 o_{22} 選擇擇一匹配的的時段變數。

也就是說，假設 b_{ij} 選擇 $\bar{b}_{ijvkk'}$ 為擇一匹配的時段變數，表示 AGV 執行 b_{ij} 的搬運途徑是從 $w_{i,j-1,k}$ 移動到 $w_{ijk'}$ 。則 $o_{i,j-1}$ 一定是選擇 $\bar{o}_{i,j-1,k'}$ 匹配，且 o_{ij} 一定選擇 $\bar{o}_{ijk''}$ 匹配。意即若 $\bar{b}_{ijvkk'}$ 存在，則 $\bar{o}_{i,j-1,k}$ 和 $\bar{o}_{ijk'}$ 必存在於最終解。如第二章所述， $PresenceOf(\cdot)$ 運算子能取得某個時段是否存在的布林值。當 $j = 1$ ， b_{ij} 前無其他產品 i 的加工作業時段變數，則只有 \bar{o}_{i1k} 存在。因此模型添加限制式



$$IfThen\left(PresenceOf\left(\bar{b}_{i1vkk'}\right), PresenceOf\left(\bar{o}_{i1k'}\right)\right), \forall i \in J, v \in G \quad (3.10)$$

當 $j = 2, 3, \dots, r_i$ ，模型添加限制式

$$IfThen\left(PresenceOf\left(\bar{b}_{ijvkk'}\right), AND\left(PresenceOf\left(\bar{o}_{i,j-1,k}\right), PresenceOf\left(\bar{o}_{ijk'}\right)\right)\right), \forall i \in J, v \in G. \quad (3.11)$$

當 $j = r_i + 1$ ， b_{ij} 之後無其他產品 i 的加工作業時段變數，則只有 $\bar{o}_{ir_{i,k}}$ 存在。因此限制式是

$$IfThen\left(PresenceOf\left(\bar{b}_{i,r_i+1,v,k,k'}\right), PresenceOf\left(\bar{o}_{ir_{i,k}}\right)\right), \forall i \in J, v \in G. \quad (3.12)$$

型別轉換間隔時間矩陣及負載運送作業時段變數相關限制

此區塊建立 AGV 空車取貨的時段，並限制 AGV 同一時間只能運送一個工件。上一區塊的限制式已確定各 AGV 負責的負載運送作業，此區塊限定 CP 求解引擎排定各 AGV 負載運送時段變數的順位。AGV 是一服務資源且係單一負載型，意即 AGV 同一時間只能運送一個工件。此外，AGV 負載運送的時段變數不能重疊，同一 AGV 的兩負載運送作業之間會至少間格空車取貨作業需要的移動時間，因此也須確保足夠的時間隔開，以下說明如何設定這些限制。

空車取貨作業會使用 AGV 一段時間，但模型建構時難以時段變數表示。空車取貨作業是派遣的 AGV 從當下的位置移動至工件所在的設施 P 點，但當下位置未定，可能在任何設施的 D 點上，無法確定起點難以限定時段變數的間距。如前所述，本問題的負載運送作業須考量排程解中前後加工作業選擇的機台，經由擇一匹配選定一個可行的搬運途徑設定間距。AGV 負載運送是從執行加工作業 $o_{i,j-1}$ 的機台 P 點移動到執行加工作業 o_{ij} 的機台 D 點，因此 AGV 出發的機台是 $o_{i,j-1}$ 的候選機台之一。相對地，若想以時段變數表示空車取貨的時程，在無法得知 AGV 當下位置的情況下，得添加所有設施點到該目的機台的可選時段變數，再建立擇一匹配的限制式。麻煩的是這些可選的時段變數與前述負載運送作業時段變數因使用 AGV 資源，須分別兩兩建立若同時出現的 $IfThen(\cdot)$ 限制條件，徒



增無數的變數和錯綜複雜的限制條件，形成一個複雜不易求解的 CP 模型。因此，本研究不使用時段變數代表空車取貨作業的時程。

當 AGV 經排程選定，先執行空車取貨作業前往工件所在設施，將工件上載後搬運至該工件選定的機台。因此 AGV 空車取貨作業後會執行負載運送作業，接著再執行下一空車取貨作業和負載運送作業，重複以上的排程直到完成所有交付給該 AGV 的搬運作業。因此任一 AGV 的兩相鄰負載運送作業排程間，都有一段空車取貨作業。本研究研擬的 CP 模型只定義負載運送時段變數，但會區別搬運途徑以設定時段變數的型別。就 AGV 的排程結果而言，本模型在兩負載運送時段變數間隔開一段足夠的時間，以容納空車取貨作業耗用的時段。CP 求解引擎須在兩負載運送時段變數間隔一段大於或等於空車取貨作業的間距。因此在兩負載運送時段變數上設定型別，兩者型別變化形成型別轉換，據此可以加入型別轉換的間隔時間限制。由 CP 求解引擎參看型別轉換的間隔時間矩陣，來隔開兩負載運送時段變數，代表兩者間有執行空車取貨作業；意即添加間隔時間，模擬空車取貨的時段。

假設 o_{ij} 選擇在第 k' 個候選機台上加工，此機台是 β ，而 $o_{i,j-1}$ 選擇在第 k 個候選機台上加工，此機台是 α 。假設 $\bar{b} = \bar{b}_{ijvkk'}$ 是選定 AGV v 從設施 α 的 P 點移動至設施 β 的 D 點的負載運送的時段變數。再假設 $\bar{b}' = \bar{b}_{ijvhh'}$ 是 AGV v 接續前面的附載運送後緊接著從設施 γ 的 P 點移動至設施 δ 的 D 點的負載運送的時段變數。如圖 3-12 所示，時段變數 \bar{b} 和 \bar{b}' 間須隔開一個時距以容納空車取貨作業；意即我們須定義 \bar{b} 和 \bar{b}' 的型別(整數)並定義兩型別的間隔時間是由設施 β 的 D 點到設施 γ 的 P 點，也就是 $t'_{\beta\gamma}$ 。令兩整數型別分別是 l 和 l' ，則轉換間隔時間 $\tau_{ll'} = t'_{\beta\gamma}$ 。已知系統內有 m 台機台，本問題的負載運送作業會由任一設施 P 點到包含同設施的任一設施 D 點，因此需要區分的型別個數是 $(m + 1)^2$ 。本研究將各 AGV 負載運送作業時段變數由設施 α 運送到設施 β 的型別設成 $\alpha(m + 1) + \beta; \alpha, \beta = 0, 1, \dots, m$ ，即設定

$$TypeOf(\bar{b}_{ijvkk'}) = \alpha(m + 1) + \beta, \forall i \in J, j = 1, 2, \dots, r_i, v \in G,$$



$$k = \begin{cases} 1, & \text{if } j = 1 \\ 1, 2, \dots, \theta_{i,j-1}, & \text{otherwise} \end{cases}, k' = \begin{cases} 1, & \text{if } j = r_i + 1 \\ 1, 2, \dots, \theta_{ij}, & \text{otherwise} \end{cases}$$

$$\alpha = \begin{cases} 0, & \text{if } j = 1 \\ w_{i,j-1,k'}, & \text{otherwise} \end{cases}; \beta = \begin{cases} 0, & \text{if } j = r_i + 1 \\ w_{ijk'}, & \text{otherwise} \end{cases} \quad (3.13)$$

舉例而言，假設 \bar{b} 是倉儲中心 D 點到倉儲中心 P 點的負載運送時段變數，其型別是 $0(m+1) + 0 = 0$ 。

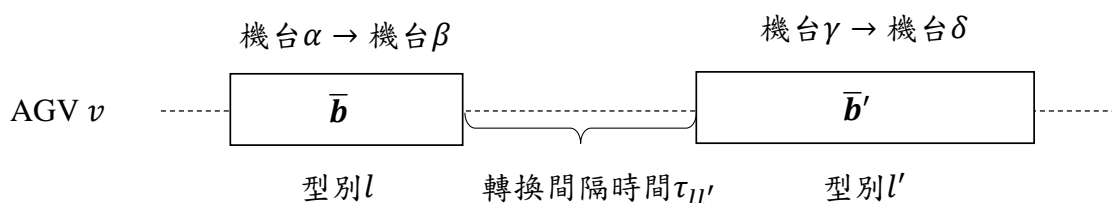


圖 3-12 時段變數 \bar{b} 和 \bar{b}' 的轉換間隔時間

本 CP 模型在 AGV 負載運送時段變數依其運送途徑不同設定型別，共有 $(m+1)^2$ 個不同起迄的型別；型別集合 $\{0, 1, 2, \dots, (m+1)^2 - 1\}$ ，各 AGV 負載運送時段變數須檢視搬運的起迄設施分別設定型別。因此配合型別轉換，可定義型別轉換間隔時間矩陣

$$\tau = [\tau_{ll'}]_{(m+1)^2 \times (m+1)^2} \quad (3.14)$$

其中

$$\tau_{ll'} = t_{\rho\rho'}, \text{ where } \rho = l \bmod (m+1), \rho' = \lfloor l' / (m+1) \rfloor \quad (3.15)$$

本問題 AGV 一開始停駐在倉儲中心 D 點。因此 AGV 的第一個搬運作業必是空車取貨作業，才能接續執行負載運送作業。都圖 3-13 是 AGV v 的排程結果模擬。AGV v 依序執行 b_{12} 、 b_{23} 、和 b_{14} 三個負載運送作業。灰色的時段是 AGV v 執行的空車取貨作業，分別是 a_{12} 、 a_{23} 、和 a_{14} 。AGV 的空車取貨作業是由兩相鄰的負載運送時段變數隔開間隔時間，模擬設定而得。AGV v 一開始停駐在倉儲中心 D 點，第一個負載運送作業 b_{12} 前沒有任何負載運送作業時段變數，無法隔



開空車取貨作業 a_{12} 需要的時間。因此，本研究引入虛擬的負載運送時段變數 $b'_v, v \in G$ 。 b'_v 的間距是0，即

$$LengthOf(b'_v) = 0, v \in G, \quad (3.16)$$

，且型別設定為0，即

$$TypeOf(b'_v) = 0, v \in G. \quad (3.17)$$

如此才能由 b'_v 和 b_{12} 建立型別轉換隔開AGV v 的第一個空車取貨作業 a_{12} 耗用的時間。



圖 3-13AGV v 的排程結果模擬

如前定義，AGV v 的序列變數 S_v 轄有使用同一AGV資源的時段變數集合，內轄的時段變數是選擇性的。CP求解引擎會選定存在於排程解的時段變數，並排定它們在AGV上執行排程順位。因此 $S_v \equiv \{\bar{b}_{ijvkk'} \mid \forall i \in J; j = 1, 2, \dots, r_i + 1; k = 1, 2, \dots, \theta_{i,j-1}; k' = 1, 2, \dots, \theta_{ij}\} \cup \{b'_v\}, v \in G$ 。

AGV v 同一時間只能運送一個工件，因此AGV負載運送的時段變數須相互隔開，且間隔時間須大於或等於兩變數型別轉換所需的轉換間隔時間，也就是空車取貨作業的時間。因此本模型須添加AGV v 的無重疊且須隔開限制

$$NoOverlap(S_v, \tau), v \in G, \quad (3.18)$$

其中 τ 是型別轉換間隔時間矩陣。此外，虛擬的負載運送時段變數 b'_v 須是 S_v 的第一個時段變數，因此添加限制式

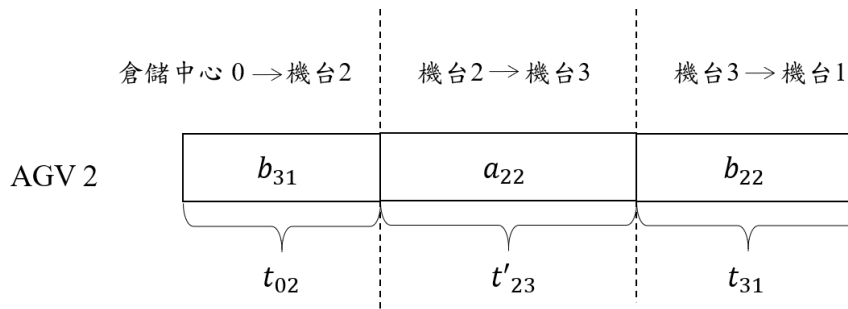
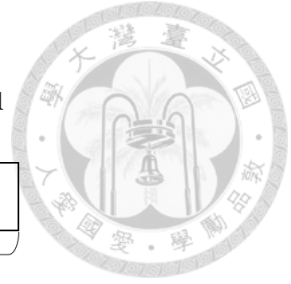
$$first(S_v, b'_v), v \in G. \quad (3.19)$$

CP求解引擎求出排程解後，可藉由AGV負載運送作業的排程，推算空車取貨作業的排程。假設系統排定AGV v 先執行負載運送作業 b_{ij} ，是將工件從機台 ρ

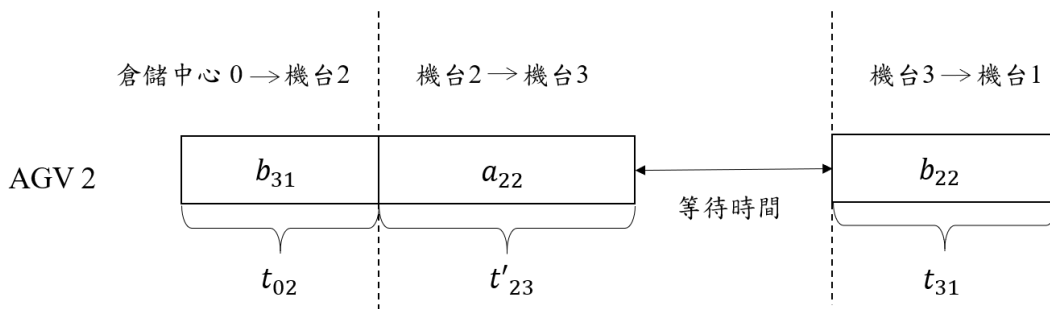
搬運至機台 ρ' ，移動時間是 $t_{\rho\rho'}$ 。再執行負載運送作業 $b_{i'j'}$ ，將工件從機台 ρ'' 搬運至機台 ρ''' ，移動時間是 $t_{\rho''\rho'''}$ 。AGV 空車取貨後，才可執行負載運送作業。由此可推得，AGV v 執行完 b_{ij} 後，在「無閒置」模式下，立即執行空車取貨作業 $a_{i'j'}$ ，即是從機台 ρ' 空車移動至機台 ρ'' ，移動時間是 $t'_{\rho'\rho''}$ 。將工件上載後，才能再執行 $b_{i'j'}$ 。

本研究產出的 AGV 排程採用「無閒置」的派遣模式。因此 AGV 完成負載運送作業後，會立即空車前往下個工件所在設施。若 AGV 抵達時，工件尚未加工完成，AGV 會在取貨點等待工件加工完成。以 AGV 的排程來看，若 AGV 上一個負載運送作業的結束時間到下一個負載運送作業的間隔時間剛好是空車取貨作業的時間，由 CP 求解引擎依給定的轉換間隔時間隔開，表示 AGV 抵達工件所在設施時，工件已加工完成，因此 AGV 可立即上載工件，負載運送至該工件的加工機台。若 AGV 上一個負載運送作業的結束時間到下一個負載運送作業的間隔時間大於空車取貨作業的時間，表示 AGV 抵達工件所在設施時，工件尚未加工完成，因此 AGV 在取貨點等待工件加工完成後才執行下一個負載運送作業。

當 CP 求解引擎排定 AGV 的負載運送作業的排程後，可以根據負載運送作業的排程推算空車取貨的排程和 AGV 的等待時間。以圖 3-14 為例，AGV 2 執行負載運送作業 b_{31} ，將工件從倉儲中心 P 點搬運至機台 2 的 D 點，移動時間是 t_{02} 。接下來，AGV 2 會先執行空車取貨作業 a_{22} ，從機台 2 的 D 點空車移動至機台 3 的 P 點，移動時間是 t'_{23} 。接著才能執行下個搬運作業 b_{22} ，將工件從機台 3 的 P 點搬運至機台 1 的 D 點，移動時間是 t_{31} 。圖(a)是 AGV 2 執行完 b_{31} 後空車移動至機台 3 的 P 點，抵達時 o_{22} 已加工完成，可立即上載工件，不須等待，負載運送至機台 1 的 D 點。圖(b)是 AGV 2 執行完 b_{31} 後立即空車移動至機台 3 的 P 點，抵達時 o_{22} 尚未加工完成。AGV 2 在原地等待一段時間後才將工件負載運送至機台 1 的 D 點。等待的時間是 b_{22} 開始時間和 a_{22} 結束時間的間隔，也就是 $EndOf(b_{22}) - EndOf(b_{31}) - t'_{23}$ 。



(a) AGV 2到達機台3時 o_{22} 已加工完成，不須等待時間



(b) AGV 2在機台3上等待 o_{22} 加工完成

圖 3-14 根據 AGV 負載運送作業推算空車取貨作業時間和等待時間

3.3.3. 目標函式

本 CP 模型設有兩種目標函式，可任選其一執行；分別是(1)最小化產品的最大完工時間，以及(2)最小化產品的總完工時間。兩種模式的求解結果比較將會在第四章詳談。

最小化產品的最大完工時間

每種產品的最大完工時間是產品加工完最後一個加工作業，由 AGV 將完工件搬運回倉儲中心 D 點的負載運送作業的結束時間，也就是產品的入庫時間。透過 CP Optimizer 的函式 $EndOf(\cdot)$ 能取得時段變數的結束時間。產品 i 的最後一個負載運送作業是 b_{i,r_i+1} ，對應的時段變數是 b_{i,r_i+1} 。 $EndOf(b_{i,r_i+1})$ 能求得 b_{i,r_i+1} 的結束時間，也就是產品 i 的完工時間。產品的最大完工時間可用以下運算式求得

$$C_{max} = \text{Max}_{i \in J} \left(\text{EndOf} \left(b_{i,r_i+1} \right) \right). \quad (3.20)$$



此種模式的目標是最小化產品的最大完工時間，目標函式為

$$\text{Minimize } C_{max} \circ \quad (3.21)$$

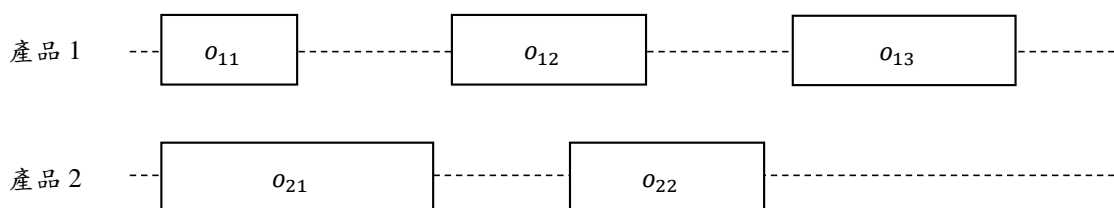
最小化產品的總完工時間

圖 3-15 是產品數量 2 的兩種不同排程結果。如圖所示，圖(a)和圖(b)的產品最大完工時間相同，但圖(a)各產品完工時間的總和比圖(b)短。此時，最小化產品的總完工時間將會是較佳的目標函式。故本模型另設目標函式是產品的總完工時間。產品的總完工時間可用以下運算式求得

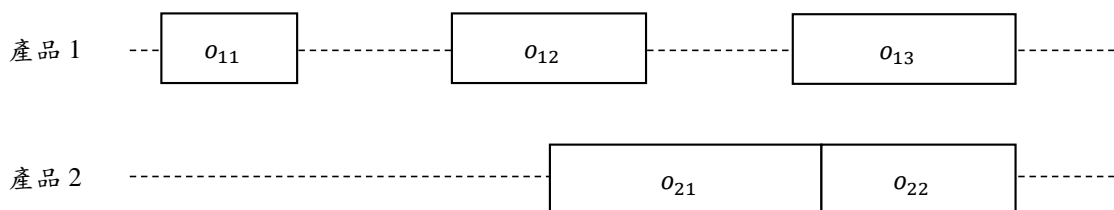
$$C_{total} = \sum_{i \in J} \text{EndOf}(b_{i,r_i+1}) \circ \quad (3.22)$$

此種模式的目標是最小化產品的最大完工時間，目標函式為

$$\text{Minimize } C_{total} \circ \quad (3.23)$$



(a) 產品總完工時間較短的排程結果



(b) 產品總完工時間較長的排程結果

圖 3-15 C_{max} 相同但 C_{total} 不同的兩種排程結果



3.4 小節

綜合前述，本節整理彈性零工生產與搬運排程問題的限制規劃求解模型。

時段變數

$o_{ij}, \forall i \in J, j = 1, 2, \dots, r_i$ ，加工作業 o_{ij} 的時段變數

$b_{ij}, \forall i \in J, j = 1, 2, \dots, r_i$ ，負載運送作業 b_{ij} 的時段變數

$b'_v, \forall v \in G$ ，AGV v 的虛擬的負載運送時段變數

可選擇的時段變數

\bar{o}_{ijk} ， o_{ij} 在第 k 台候選機台上執行對應的時段變數， $\forall i \in J, j = 1, 2, \dots, r_i, k =$

$1, 2, \dots, \theta_{ij}$

$\bar{b}_{ijvkk'}$ ，AGV v 從機台 w_{ijk} 的 P 點移動至機台 $w_{i,j-1,k}$ 的 D 點的負載運送作業 b_{ij} 對

應的時段變數， $\forall i \in J, j = 1, 2, \dots, r_i, k = \begin{cases} 1, & \text{if } j = 1 \\ 1, 2, \dots, \theta_{i,j-1}, & \text{otherwise} \end{cases}, k' =$

$\begin{cases} 1, & \text{if } j = r_i + 1 \\ 1, 2, \dots, \theta_{ij}, & \text{otherwise} \end{cases}$

序列變數

D_r ，機台 r 的序列變數， $D_r \equiv \{\bar{o}_{ijk} \mid w_{ijk} = r, \forall i \in J; j = 1, 2, \dots, r_i; k =$

$1, 2, \dots, \theta_{ij}\}$

S_v ，AGV v 的序列變數， $S_v \equiv \{\bar{b}_{ijvkk'} \mid \forall i \in J; j = 1, 2, \dots, r_i + 1; k =$

$1, 2, \dots, \theta_{i,j-1}; k' = 1, 2, \dots, \theta_{ij}\} \cup \{b'_v\}, v \in G$

轉換間隔時間矩陣

$$\tau = [\tau_{ll'}]_{(m+1)^2 \times (m+1)^2} \quad (3.14)$$

$$\tau_{ll'} = t_{\rho\rho'}, \text{ where } \rho = l \bmod (m+1), \rho' = \lfloor l'/(m+1) \rfloor \quad (3.15)$$



時段變數型別設定

$$TypeOf(\bar{\mathbf{b}}_{ijvkk'}) = \alpha(m+1) + \beta, \forall i \in J, j = 1, 2, \dots, r_i, v \in G,$$

$$k = \begin{cases} 1, & \text{if } j = 1 \\ 1, 2, \dots, \theta_{i,j-1}, & \text{otherwise} \end{cases}, k' = \begin{cases} 1, & \text{if } j = r_i + 1 \\ 1, 2, \dots, \theta_{ij}, & \text{otherwise} \end{cases}$$

$$\alpha = \begin{cases} 0, & \text{if } j = 1 \\ w_{i,j-1,k'}, & \text{otherwise} \end{cases}; \beta = \begin{cases} 0, & \text{if } j = r_i + 1 \\ w_{ijk'}, & \text{otherwise} \end{cases} \quad (3.13)$$

$$TypeOf(\mathbf{b}'_v) = 0, v \in G \quad (3.17)$$

限制式

$$LengthOf(\bar{\mathbf{o}}_{ijk}) = \bar{p}_{ijk} \quad (3.3)$$

$$LengthOf(\bar{\mathbf{b}}_{ijvkk'}) = t_{\alpha\beta}, \alpha = \begin{cases} 0, & \text{if } j = 1 \\ w_{i,j-1,k'}, & \text{otherwise} \end{cases}; \beta = \begin{cases} 0, & \text{if } j = r_i + 1 \\ w_{ijk'}, & \text{otherwise} \end{cases} \quad (3.7)$$

$$LengthOf(\mathbf{b}'_v) = 0, v \in G \quad (3.16)$$

$$Alternative(\mathbf{o}_{ij}, \mathbf{O}_{ij}), \forall i \in J, j = 1, 2, \dots, r_i \quad (3.5)$$

$$Alternative(\mathbf{b}_{ij}, \mathbf{B}_{ij}), \forall i \in J, j = 1, 2, \dots, r_i + 1 \quad (3.9)$$

$$EndBeforeStart(\mathbf{b}_{ij}, \mathbf{o}_{ij}), \forall i \in J, j = 1, 2, \dots, r_i \quad (3.1)$$

$$EndBeforeStart(\mathbf{o}_{ij}, \mathbf{b}_{i,j+1}), \forall i \in J, j = 1, 2, \dots, r_i \quad (3.2)$$

$$NoOverlap(D_r), r = 1, 2, \dots, m \quad (3.6)$$

$$NoOverlap(S_v, \tau), v \in G \quad (3.18)$$

$$first(S_v, \mathbf{b}'_v), v \in G \quad (3.19)$$

$$IfThen(PresenceOf(\bar{\mathbf{b}}_{i1vkk'}), PresenceOf(\bar{\mathbf{o}}_{i1k'})), \forall i \in J, v \in G \quad (3.10)$$

$$IfThen(PresenceOf(\bar{\mathbf{b}}_{ijvkk'}), AND(PresenceOf(\bar{\mathbf{o}}_{i,j-1,k}), PresenceOf(\bar{\mathbf{o}}_{ijk'}))),$$

$$\forall i \in J, j = 2, \dots, r_i, v \in G \quad (3.11)$$

$$IfThen(PresenceOf(\bar{\mathbf{b}}_{i,r_i+1,v,k,k'}), PresenceOf(\bar{\mathbf{o}}_{ir_{ik}})), \forall i \in J, j = r_i + 1, v \in G \quad (3.12)$$

目標函式

模型 1：最小化最大完工時間

Minimize C_{max}

$$C_{max} = \text{Max}_{i \in J} \left(\text{EndOf}(\mathbf{b}_{i,r_i+1}) \right)$$

模型 2：最小化總完工時間

Minimize C_{total}

$$C_{total} = \sum_{i \in J} \text{EndOf}(\mathbf{b}_{i,r_i+1})$$



(3.21)

(3.20)

(3.23)

(3.22)

第4章 彈性零工生產與搬運排程問題的求解測試及應用

前一章已經完整定義問題，並詳細解說本研究研擬的限制規劃求解模型。本章首先介紹本研究測試使用的標竿問題來源，接著說明開發的求解系統架構。最後以 IBM ILOG® CPLEX 的 CP Optimizer 實作本研究研擬的限制規劃模型。測試本求解模型求解標竿問題的效能，並與文獻中的求解結果比較。

4.1 標竿問題

目前文獻中少有考量搬運的彈性零工生產問題，也缺乏各種尺度的標竿問題。本研究驗證 CP 求解模型效能。測試的標竿問題參考自(李佳陽, 2019)文獻中的二類標竿問題。第一類是(Liang et al., 2012)文獻中展示的標竿問題，以下稱為 Y 群問題。第二類是(李佳陽, 2019)參考(Brandimarte, 1993)和(Kacem et al., 2002)的 FJSP 標竿問題，添加 AGV 搬運架構而產生的標竿問題，稱為 L 群問題。下文進一步說明標竿問題文字檔的格式，並介紹 Y 群和 L 群問題。

4.1.1. 標竿問題格式

本問題的標竿問題格式來自(李佳陽, 2019)，係以文字檔儲存。內容包括問題的標題(title)、AGV 和機台的數量、P 點到 D 點與 D 點到 P 點的從至時間矩陣、產品的數量、各加工作業的候選加工作業機台、和其對應的加工時間。圖 4-1 是標竿問題的範例檔案。區塊 A 記錄此標竿問題的名稱。區塊 B 記錄 AGV 數量 g 和機台數量 m 。區塊 C 先記錄從至時間矩陣的大小，也就是機台加上一個倉儲中心的數量。內容以斜線隔開的成對數字分別是 P 點到 D 點的從至時間矩陣 T 和 D 點到 P 點的從至時間矩陣 T' 。時間值是由 AGV 實際在工廠中每秒 1-2 公尺的運行速度和兩站點之間的距離換算而得。區塊 D 依序紀錄產品數量 n 、該產品的加工作業數量、各加工作業的候選機台數。候選機台數量的下一列是成對出現的加工機台編號和對應的加工時間。

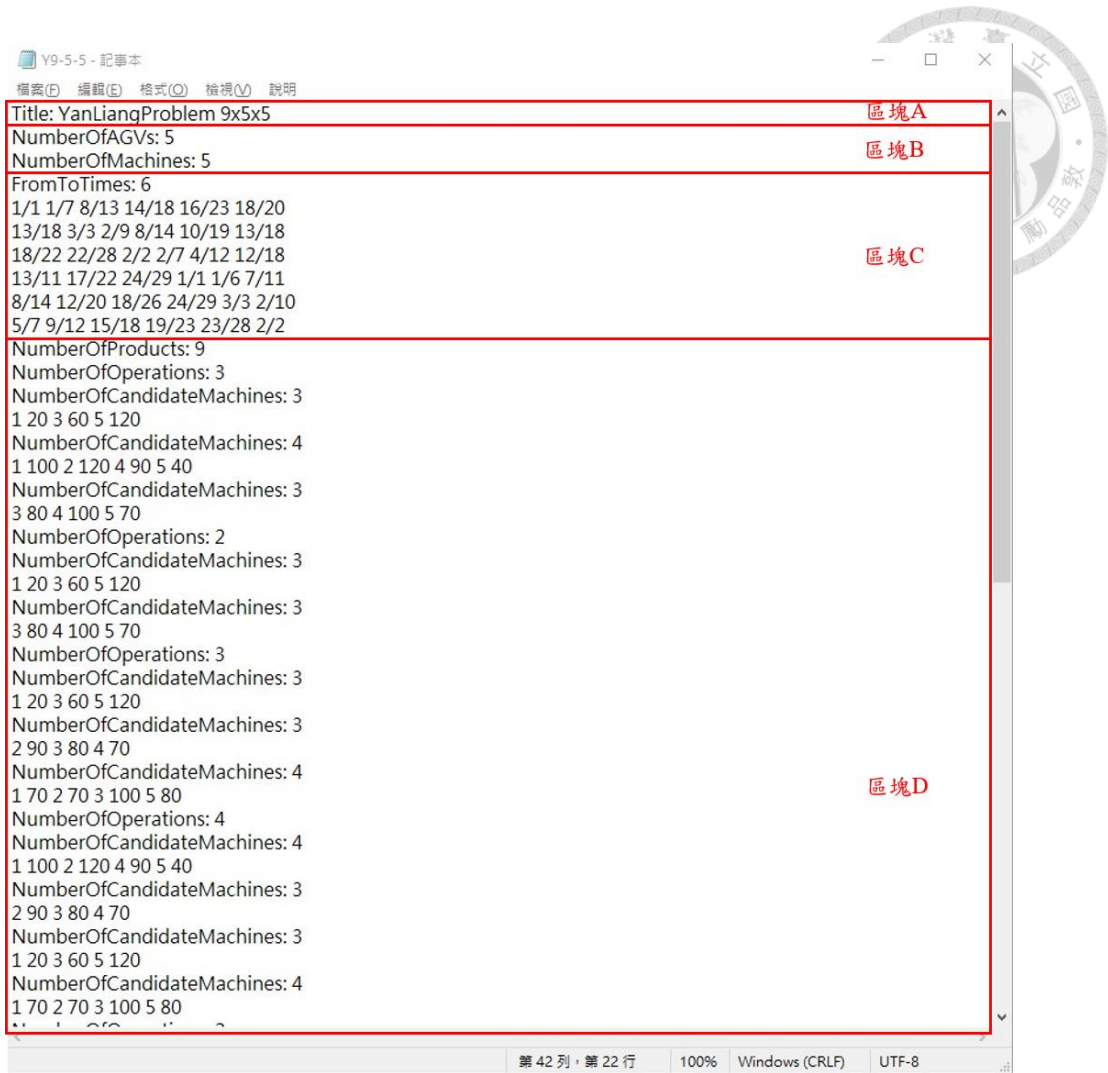


圖 4-1 標竿問題檔案內容範例

4.1.2. Y 群問題

Y 群問題是(Liang et al., 2012)文獻展示的標竿問題。Y 群問題有 3 個，其中有 1 個小型問題和 2 個中型問題。問題的命名是根據問題中的產品、機台、和 AGV 數量，例如： Y_{3-4-3} 表示該問題屬於 Y 群問題，且有 3 種產品、4 台加工機台、和 3 台 AGV。表 4-1 是 Y_{3-4-3} 問題各加工作業的候選機台和對應的加工時間。表 4-2 是 Y_{3-4-3} 問題 AGV 在機台和倉儲中心間移動的時間。

表 4-1 Y_{3-4-3} 問題各加工作業的候選機台和對應的加工時間

產品	加工作業	機台			
		1	2	3	4
1	o_{11}	60	92	-	100
	o_{12}	-	80	40	20
	o_{13}	-	50	100	-
	o_{14}	100	-	-	70
2	o_{21}	60	92	-	100
	o_{22}	-	80	40	20
3	o_{31}	100	-	-	70
	o_{32}	-	80	40	20
	o_{33}	-	50	100	-

表 4-2 Y_{3-4-3} 問題 AGV 在機台和倉儲中心間移動的時間

D到P / P到D	倉儲中心		機台			
	0	1	2	3	4	
倉儲中心	0	1/1	1/7	8/13	16/23	18/20
	1	13/18	3/3	2/9	10/19	13/18
	2	18/22	22/28	2/2	4/13	12/18
機台	3	8/14	12/20	18/26	3/3	2/10
	4	5/7	9/12	15/18	23/28	2/2

另外兩個中型的標竿問題是 Y_{9-5-4} 和 Y_{9-5-5} 。這兩個標竿問題都有 9 種產品和 5 台加工機台，而 AGV 分別有 4 台和 5 台。 Y_{9-5-4} 和 Y_{9-5-5} 詳細的資料收於附錄一。



4.1.3. L 群問題

L 群問題是擴充自 Kacem 8x8、Kacem 15x10 兩個 FJSP 標竿問題(Kacem et al., 2002)、和 Brandimarte 15x8 FJSP 標竿問題(Brandimarte, 1993)。Kacem 和 Brandimarte 二者是學界探討彈性零工生產問題時常用來比較不同解法的標竿問題。以 Kacem 8x8 為例，8x8 是指問題內有 8 種產品和 8 台機台。Kacem 8x8 和 Brandimarte 15x8 是部分彈性的零工生產問題，而 Kacem 15x10 是完全彈性的零工生產問題，而每個加工作業都可在任何機台上加工。

(李佳陽, 2019)將這 3 個 FJSP 問題擴充成本研究探討的問題，他依據常見的 U 型和直線型機台布置設計 AGV 在設施間的搬運時間。L 群問題有 18 個，表 4-3 是這些問題的來源、設計方式、和問題符號。這些問題的詳細資料收於附錄二。

表 4-3 L 群問題的 18 個問題

來源	機台布置	加工時間比搬運時間	L 群標竿問題符號
Kacem 8x8	U 型	1:1	$L_{8-8-x}^{(1)}$
		仿(Liang, Lin et al. 2012)比例	L_{8-8-x}
		10:1	$L_{8-8-x}^{(10)}$
	直線型	1:1	$\bar{L}_{8-8-x}^{(1)}$
		仿(Liang, Lin et al. 2012)比例	\bar{L}_{8-8-x}
		10:1	$\bar{L}_{8-8-x}^{(10)}$
Kacem 15x10	U 型	1:1	$L_{15-10-x}^{(1)}$
		仿(Liang, Lin et al. 2012)比例	$L_{15-10-x}$
		10:1	$L_{15-10-x}^{(10)}$
	直線型	1:1	$\bar{L}_{15-10-x}^{(1)}$
		仿(Liang, Lin et al. 2012)比例	$\bar{L}_{15-10-x}$
		10:1	$\bar{L}_{15-10-x}^{(10)}$
Brandimarte 15x8	U 型	1:1	$L_{15-8-x}^{(1)}$
		仿(Liang, Lin et al. 2012)比例	L_{15-8-x}
		10:1	$L_{15-8-x}^{(10)}$
	直線型	1:1	$\bar{L}_{15-8-x}^{(1)}$
		仿(Liang et al., 2012)比例	\bar{L}_{15-8-x}
		10:1	$\bar{L}_{15-8-x}^{(10)}$



4.2 求解系統

本研究依據前一章規劃的求解模型及演算流程，使用 Microsoft Visual Studio 2019 程式開發工具，在 Microsoft .Net Framework 4.6.1 平台下，以 C# 程式語言實作彈性零工生產與搬運排程問題的求解系統。本節介紹系統的操作介面和使用流程。

求解系統的主要人機介面如圖 4-2 所示。圖 4-2 區塊 A 內的「Open」指令由使用者選取標竿問題檔案，再由系統讀入標竿問題的基本資料。資料內容包含標竿問題名稱、AGV 數量、機台數量、AGV 搬運的時間矩陣、產品數量、各產品的加工作業數量、和各加工作業的候選機台數量以及分別對應機台的加工時間。區塊 B 供使用者調整問題中的 AGV 數量，也可設定 CP Optimizer 求解的停止條件和目標函式。再按下「Solve」讓系統開始求解。本求解系統的停止條件是限制規劃求解模型求解問題的錯誤次數，預設為 1,000,000 次。當 CP 求解引擎求解的錯誤次數到達設定的值時，本問題的解就是迄今找到的最佳解。系統提供兩種目標函式設定模式，分別是產品的最大完工時間和產品的完工時間總和。區塊 C 展示標竿問題的資料內容。區塊 D 是求解排程結果的詳細資訊和統計數據。區塊 E 展示排程結果的甘特圖。區塊 F 顯示求解過程花費的時間。

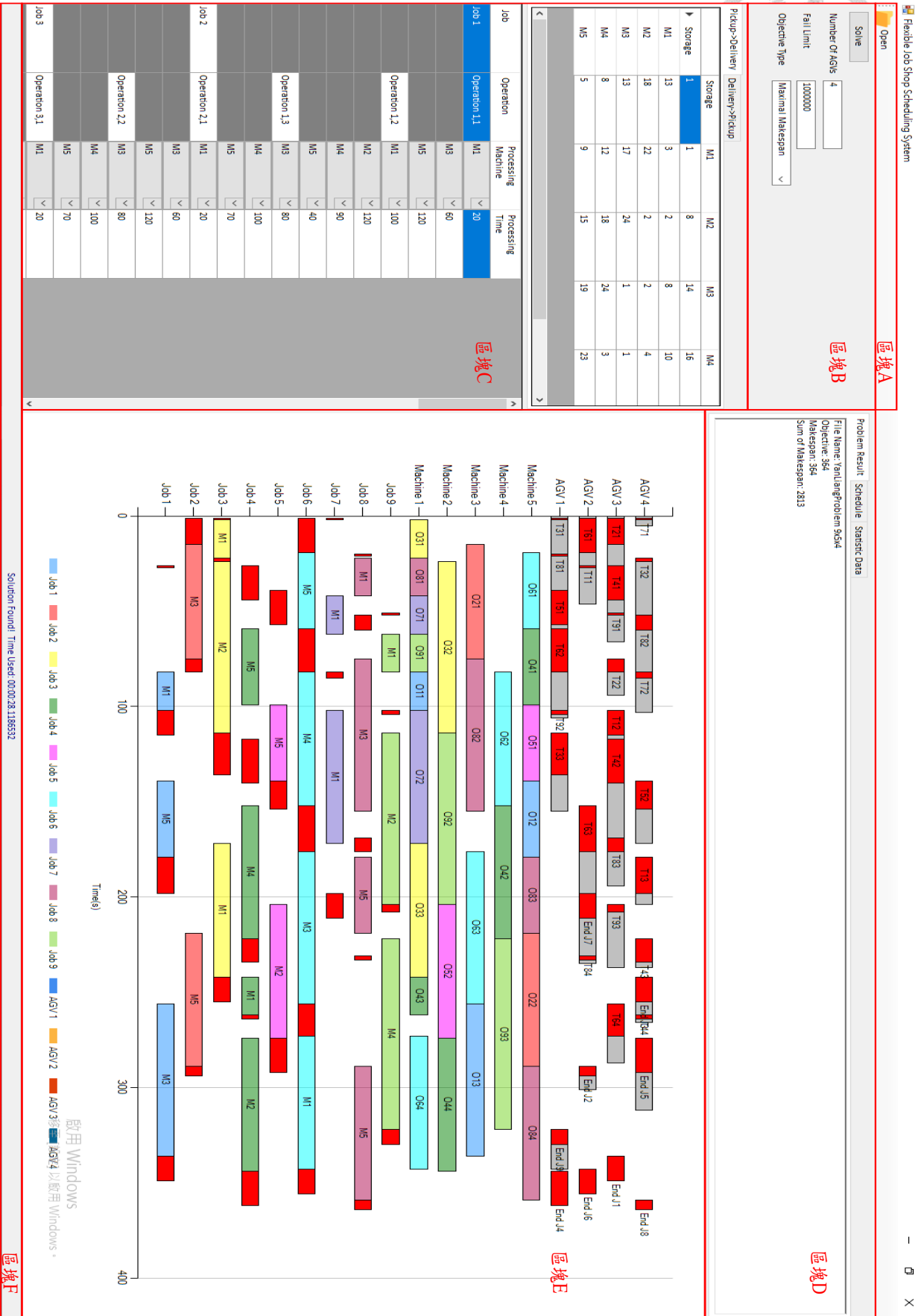


圖 4-2 求解系統人機介面



圖 4-3 上半部是 AGV 的搬運時間，有「Pickup->Delivery」和「Delivery-> Pickup」兩個分頁，分別是 AGV 在各設施間 P 點到 D 點以及 D 點到 P 點的移動時間。下半部是產品加工作業的詳細資料，產品和加工作業的名稱是系統由輸入的資料依序編成。

Pickup->Delivery		Delivery->Pickup				
	Storage	M1	M2	M3	M4	
Storage	1	1	8	14	16	
M1	13	3	2	8	10	
M2	18	22	2	2	4	
M3	13	17	24	1	1	
M4	8	12	18	24	3	
M5	5	9	15	19	23	

Job	Operation	Processing Machine	Processing Time
ob 1	Operation 1,1	M1	20
		M3	60
		M5	120
	Operation 1,2	M1	100
		M2	120
		M4	90
		M5	40
	Operation 1,3	M3	80
		M4	100
M5		70	
ob 2	Operation 2,1	M1	20
		M3	60
		M5	120
	Operation 2,2	M3	80

圖 4-3 標竿匯入問題的資訊

圖 4-4 是頁面「Problem Result」顯示的資訊，紀錄問題的基本資料和求解結果。內容是輸入問題的標題、限制規劃求出的最佳解、產品最大完工時間、以及產品的總完工時間。

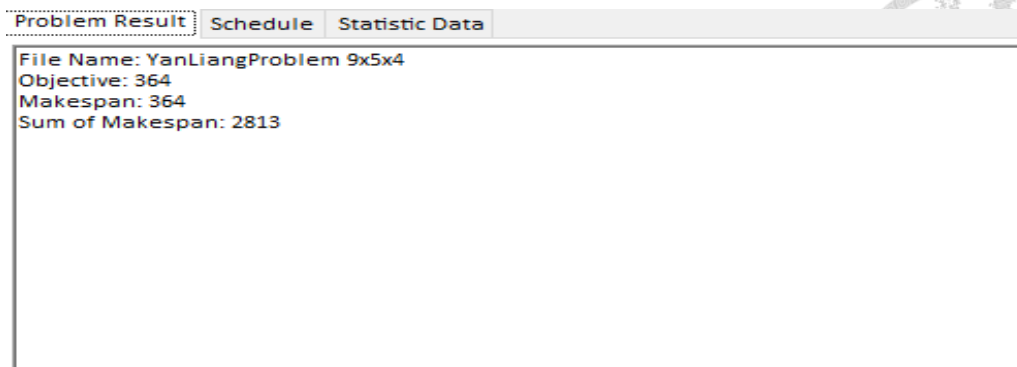


圖 4-4 問題標題和求解結果

圖 4-5 是頁面「Schedule」顯示的資訊，記錄詳細的排程結果。內容是各加工作業的名稱、開始時間、結束時間、選擇的機台和執行加工作業的搬運作業的 AGV。

Problem Result	Schedule	Statistic Data
加工作業 開始時間 / 結束時間 加工機台 搬運AGV		
Job 1		
Operation 1,1: 82 / 102 M1 AGV 2		
Operation 1,2: 139 / 179 M5 AGV 3		
Operation 1,3: 256 / 336 M3 AGV 4		
Job 2		
Operation 2,1: 15 / 75 M3 AGV 3		
Operation 2,2: 219 / 289 M5 AGV 3		
Job 3		
Operation 3,1: 2 / 22 M1 AGV 1		
Operation 3,2: 24 / 114 M2 AGV 4		
Operation 3,3: 172 / 242 M1 AGV 1		
Job 4		
Operation 4,1: 59 / 99 M5 AGV 3		
Operation 4,2: 152 / 222 M4 AGV 3		
Operation 4,3: 242 / 262 M1 AGV 4		
Operation 4,4: 274 / 344 M2 AGV 4		

圖 4-5 加工作業詳細排程結果

圖 4-6 是頁面「Statistic Data」顯示的資訊，紀錄求解後機台和 AGV 的統計資料。內容包含各 AGV 等待、負載運送、以及空車取貨狀態所佔的時間比例，另外還有機台閒置和忙碌狀態所佔的時間比例。此頁面同時記錄了 AGV 和機台的平均稼動率。

AGV Idle Rate / Delivery Trip Rate / Pickup Trip Rate

AGV1: 48.62% / 25.69% / 25.69%

AGV2: 56.74% / 21.35% / 21.91%

AGV3: 27.22% / 33.52% / 39.26%

AGV4: 37.91% / 26.92% / 35.16%

AGV平均稼働率: 0.57

Machine Idle Rate / Busy Rate

Machine1: 3.79% / 96.21%

Machine2: 6.98% / 93.02%

Machine3: 10.71% / 89.29%

Machine4: 25.47% / 74.53%

Machine5: 5.29% / 94.71%

機台平均稼働率: 0.9

圖 4-6 機台和 AGV 的統計資料

圖 4-7 展示排程結果甘特圖，分為上、中、和下三個部分。上部是 AGV 排程結果的甘特圖塊。AGV 有三種狀態，紅色的圖塊代表「負載運送」，灰色的圖塊代表「空車取貨」，而空白的部分代表 AGV 在設施上「等待」。圖塊中的標籤指出該 AGV 執行的負載運送作業，例如：「T32」是加工作業 O_{32} 的負載運送作業。中部是機台排程結果的甘特圖塊。機台有兩種狀態，有塗色的圖塊代表機台「忙碌」，不同顏色表示不同產品的加工作業，而空白的部分代表機台「閒置」。圖塊中的標籤指出該機台執行的加工作業，例如：「O51」是產品 5 的第 1 個加工作業 O_{51} 。下部是產品排程結果的甘特圖塊，每種產品的加工作業都以一種顏色表示。產品有四種狀態，有塗色的圖塊代表產品「加工中」，紅色的圖塊代表產品「轉運中」，空白的部分代表產品「等待機台加工」或是「等待 AGV」。圖塊中的標籤指出該加工作業執行的機台，例如：「M4」代表該加工作業在機台 4 加工。

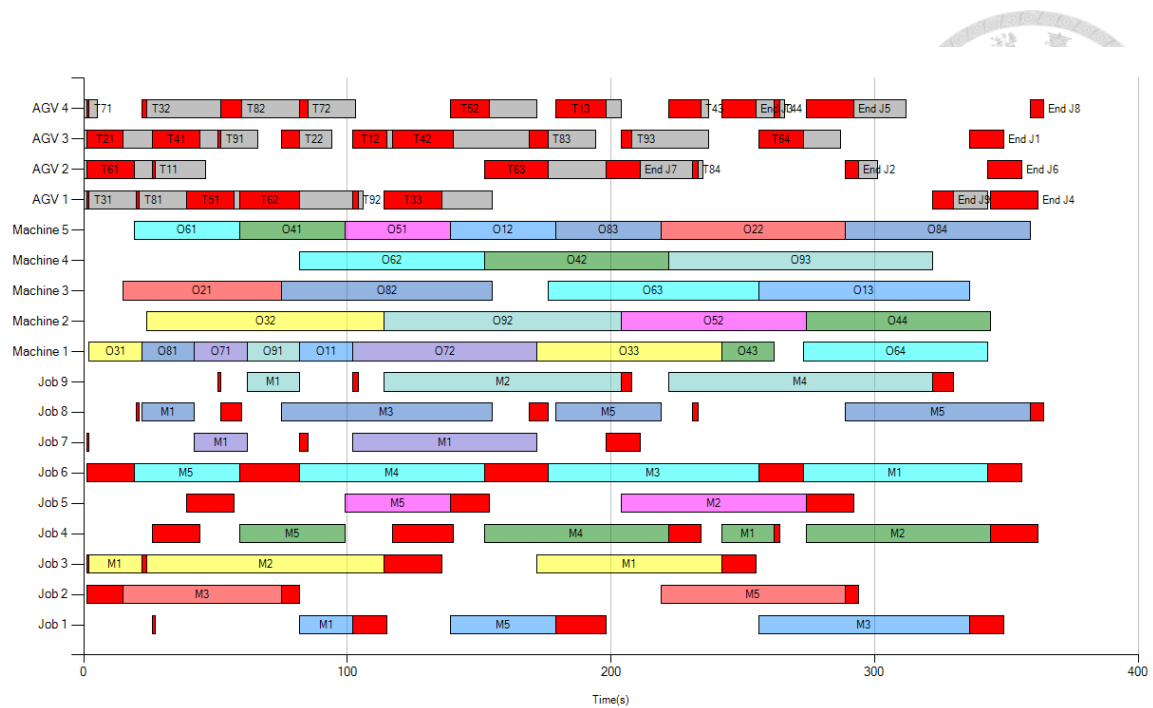


圖 4-7 排程結果甘特圖

4.3 範例測試與效能分析

本研究研擬彈性生產與搬運排程問題的限制規劃求解模型後，以 IBM ILOG® CPLEX 內的 CP Optimizer 建模套件實作求解模型。為驗證此求解模型的效能，本研究在配備 Intel® Core™ I5-8500 CPU 3.00GHz、64 位元作業系統、32G 記憶體、與 512G SSD 的個人電腦上進行各個範例測試。本研究執行三種測試，分別是(1) Y 群問題的範例求解測試，(2) L 群問題的範例求解測試，以及(3) 限制規劃模型設定不同目標函式的情境測試。

Y 群問題的範例測試

為驗證本研究研擬的限制規劃求解模型求解中小型問題的效能，本測試以 Y 群問題為求解案例。Y 群問題的 3 個標竿問題已有問題文獻發表求解成果，因此本測試以限制規劃求解模型求解相同的問題，並與文獻成果做比較。本測試探討範例求解時，分別設定 50,000、100,000、500,000、和 1,000,000 四種不同錯誤次數參數下，求解 Y 群問題的 3 個標竿問題的成效。在不同錯誤次數的測試下，探討標竿問題最大完工時間和求解使用的 CPU 時間的差異。

比較本研究的 CP 求解模型、(李佳陽, 2019)的GA⁺啟發式演算法、以及(Liang et al., 2012)的 RKPSO 演算法的求解數值結果。GA⁺的加工作業段染色體採 Order 交配和 Reciprocal 突變，AGV 段和機台段皆採 SVD 交配及泛用型隨機基因值重設法。GA⁺求解Y₃₋₄₋₃問題的染色體數量設定為 100，交配率 0.8，突變率 0.8，初始貪婪解占比設為 0.01，目標函式呼叫上限為 2,000,000；針對Y₉₋₅₋₄和 Y₉₋₅₋₅兩個大型的標竿問題，GA⁺演算法染色體數量設定為 400，交配率 0.8，突變率 0.8，初始貪婪解占比設為 0.01，目標函式呼叫上限為 1,000,000。(Liang et al., 2012)文獻中沒有說明針對Y₃₋₄₋₃問題的參數設定；RKPSO 演算法針對Y₉₋₅₋₄和 Y₉₋₅₋₅問題的參數設定是母體數 10 個，交配率 0.02，突變率 0.02，但文獻中沒有說明演化停止條件為何。

表 4-4 列出求解 Y 群問題時 CP 模型的變數和限制式數量。表 4-5 上半部列出(李佳陽, 2019)文獻中GA⁺法和(Liang et al., 2012)文獻中 RKPSO 法求解 Y 群問題的 30 次的平均最大完工時間以及求得最佳的最大完工時間。表 4-5 下半部是 CP 模型在不同錯誤次數的設定下求解 Y 群問題求得的最大完工時間，括號中的數值是 CPU 的求解時間。表中的*代表求解的結果是問題的全域最佳解。

表 4-4 Y 群問題的變數和限制式數量

CP模型參數	標竿問題		
	Y ₃₋₄₋₃	Y ₉₋₅₋₄	Y ₉₋₅₋₅
變數數量	219	1,211	1,474
限制式數量	214	1,174	1,437

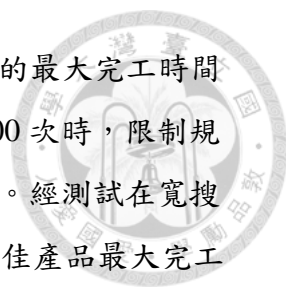
表 4-5 不同求解法求解 Y 群問題的結果

求解法		標竿問題		
		Y_{3-4-3}	Y_{9-5-4}	Y_{9-5-5}
GA ⁺ 法	30次平均	261	369	369
	最佳解	261	362	362
RKPSO法	30次平均	-	581.8	524.5
	最佳解	321	574	512
CP求解模型 (錯誤次數)	50,000次	261* (0.04s)	398 (1.41s)	403 (1.65s)
	100,000次	261* (0.04s)	383 (3.32s)	364 (3.64s)
	500,000次	261* (0.04s)	364 (16.11s)	362 (17.61s)
	1,000,000次	261* (0.04s)	364 (30.42s)	362 (36.82s)

因 Y 群問題的複雜度比 L 群問題低，錯誤次數最多設定在 1,000,000 次。比較求解 Y 群問題的結果，CP 求解模型求解結果皆優於(Liang et al., 2012)文獻展示的最佳解。

由表 4-5 的求解結果顯示限制規劃求解 Y_{3-4-3} 問題，在低停止條件(錯誤次數 50,000 次內)就能在 0.04 秒內求得最小的最大完工時間 261，求解效能與GA⁺演算法相同，且 CP 模型求解結果顯示此解為全域最佳解。此小型問題 RKPSO 法僅求得 321，效能不佳。

中型問題中 Y_{9-5-5} 的 AGV 數量比 Y_{9-5-4} 多一台，但 CP 模型的變數和限制式的數量分別增加了兩百多個。因此即使產品和機台的數量相同，AGV 的數量的增加會增加問題求解的複雜度。求解 Y_{9-5-5} 問題時限制規劃求解模型在錯誤次數設定 100,000 次下，求得的最大完工時間 364 已比GA⁺演算法求解 30 次的平均值 369 好。當錯誤次數設定 500,000 次時，求得的最佳值 362 與GA⁺演算法相同。



限制規劃求解 Y_{9-5-4} 問題在錯誤次數設定 500,000 下，求得的最大完工時間 364 較GA⁺演算法 30 次的平均 369 佳。當錯誤次數設定 1,000,000 次時，限制規劃的求得的最大完工時間 364 接近GA⁺演算法求得的最佳值 362。經測試在寬搜尋條件(錯誤次數設定達 4,500,000 次)下，CP 求解引擎可求得最佳產品最大完工時間 362 的排程解，數值與GA⁺法相同。由此可知，問題複雜度較低時，限制規劃模型的解搜尋不一定比GA⁺法好，但能在短時間內求得與GA⁺最佳解相近或是同樣的結果。此外，在相同錯誤次數下，問題複雜度愈高的問題，求解時間必然愈長。

求解測試 Y 群標竿問題，本研究研擬的限制規劃求解模型皆能求得比(李佳陽, 2019)文獻更佳或是相同的解。對於 Y_{3-4-3} 和 Y_{9-5-5} 兩個標竿問題，限制規劃模型在錯誤次數設定 500,000 次下，意即低停止條件時即可求得與(李佳陽, 2019)文獻展示的最佳解相同的結果。(Liang et al., 2012)文獻中提示的 RKPSO 演算法演化成效不彰，結果明顯劣於限制規劃法，如 Y_{9-5-5} 的最大完工時間 574 遠高於本模型求得的 362。該混合法演化效能不佳，這也是啟發式演算法常見且難以解決的問題。圖 4-8 至圖 4-10 是本研究求解 Y 群問題的最佳排程結果的甘特圖，詳細排程內容收錄於附錄三。

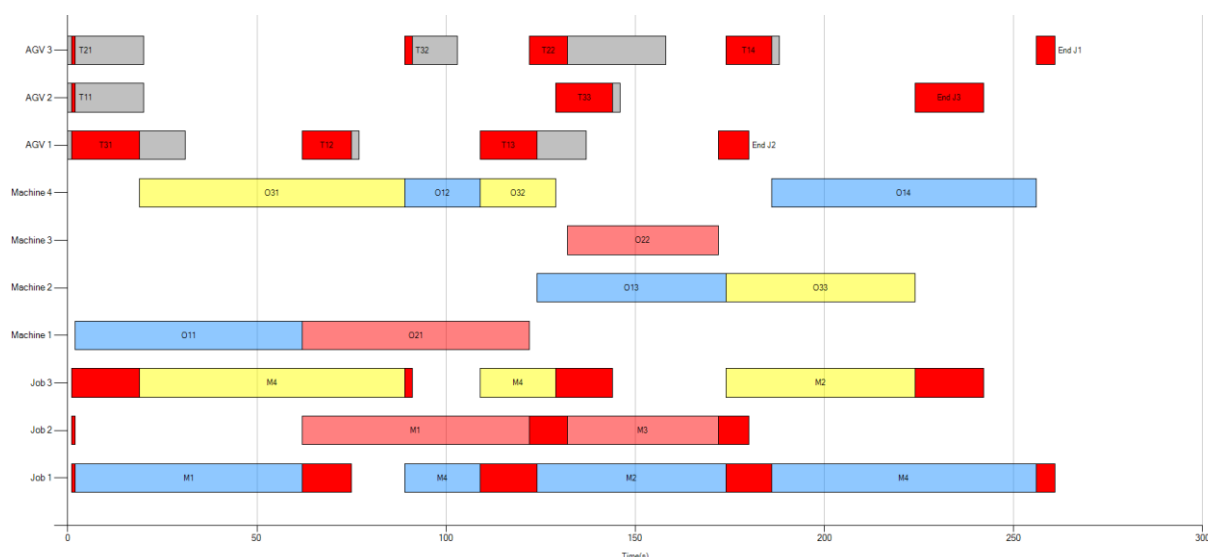


圖 4-8 Y_{3-4-3} 排程結果甘特圖

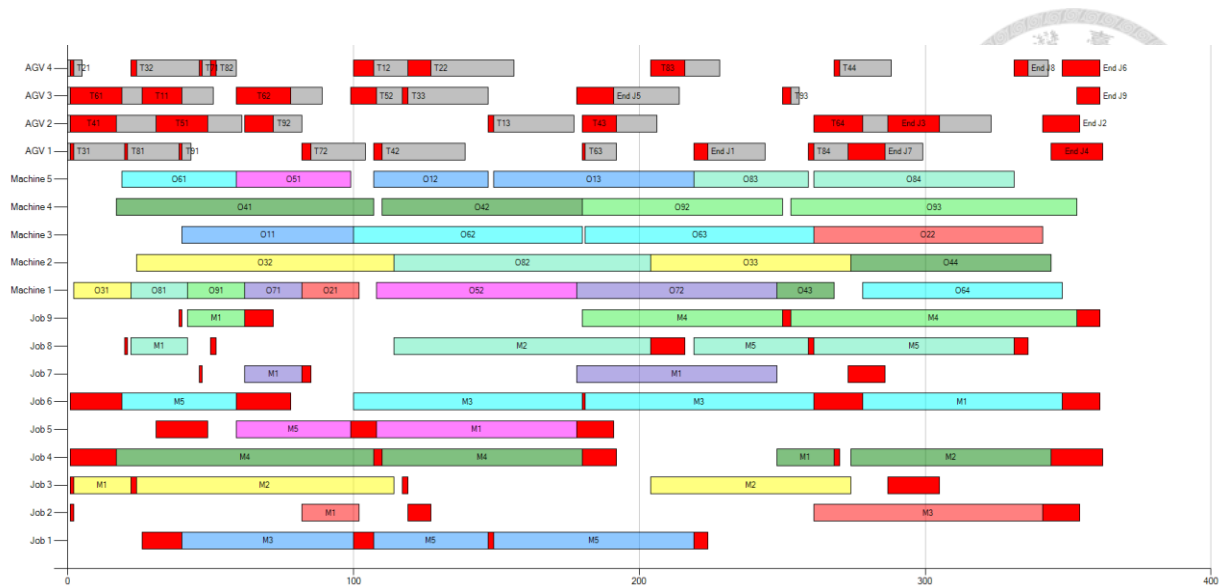


圖 4-9 Y_{9-5-4} 排程結果甘特圖

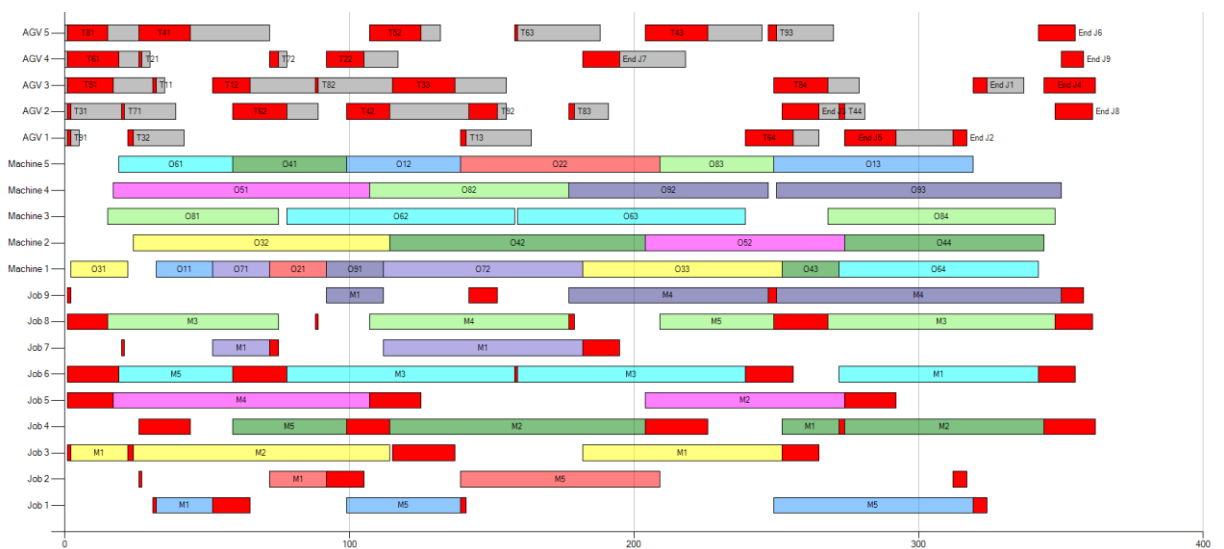


圖 4-10 Y_{9-5-5} 排程結果甘特圖

此外，由圖 4-9 及圖 4-10 能看出 Y_{9-5-5} 和 Y_{9-5-4} 問題的最大完工時間皆為 362。但 Y_{9-5-5} 中的 AGV 1 稼動率低，耗時許多時間停留在機台上等待工件加工完成。 Y_{9-5-4} 中各 AGV 都沒有太多的等待時間。因此能得知此中型問題僅須 4 台 AGV。

L 群問題的範例測試

為驗證本研究研擬的限制規劃求解模型求解大型問題的效率，本測試的求解案例皆採用 U 型機台布置以及仿文獻機台加工和搬運時間比例，分別是 L 群問題



中的 L_{8-8-5} 、 $L_{15-10-6}$ 、 L_{15-8-6} 。這3個標竿問題有求解成果資料，因此本測試以限制規劃求解模型求解相同的問題，比較求解成果。因L群問題的複雜度較高，本測試探討範例求解時，分別設定1,000,000、2,000,000、3,000,000、和4,000,000四種不同錯誤次數下，求解L群3個標竿問題。在不同錯誤次數的測試下，探討標竿問題最大完工時間和求解使用的CPU時間的差異。

測試本研究的CP求解模型求解標竿問題的結果並與(李佳陽, 2019)的GA⁺啟發式演算法比較。GA⁺的加工作業段染色體採Order交配和Reciprocal突變，AGV段和機台段皆採SVD交配及泛用型隨機基因值重設法。GA⁺求解L群問題的染色體數量設定為100，交配率0.8，突變率0.8，初始貪婪解占比設為0.01。針對 L_{8-8-5} 問題和 L_{15-8-6} 問題目標函式呼叫上限為2,000,000；而 $L_{15-10-6}$ 的目標函式呼叫上限設定為4,000,000。

表4-6列出求解Y群問題時CP模型的變數和限制式數量。表4-7上半部列出(李佳陽, 2019)文獻中GA⁺法求解L群問題的30次的平均最大完工時間以及求得最佳的最大完工時間。表4-7下半部是本研究研擬的CP模型在不同錯誤次數的設定下求解L群問題求得的最大完工時間。表中括號內的數值是不同求解法CPU的求解時間。表中的*代表求解的結果是問題的全域最佳解。

表 4-6 L 群問題的變數和限制式數量

CP模型參數	標竿問題		
	L_{8-8-5}	L_{15-8-6}	$L_{15-10-6}$
變數數量	4,810	8,532	27,109
限制式數量	4,689	8,381	26,661

表 4-7 不同求解法求解 L 群問題的結果

求解法		標竿問題		
		L ₈₋₈₋₅	L ₁₅₋₈₋₆	L ₁₅₋₁₀₋₆
GA ⁺ 法	30次平均	194	2,864	231
	最佳解	184 (33s)	2,864 (4min25s)	219 (3min52s)
CP求解模型 (錯誤次數)	1,000,000次	183 (3min14s)	2,864* (28.01s)	212 (4min52s)
	2,000,000次	183 (5min22s)	2,864* (28.01s)	207 (9min29s)
	3,000,000次	183 (8min)	2,864* (28.01s)	206 (14min55s)
	4,000,000次	183 (10min47s)	2,864* (28.01s)	201 (20min57s)

L 群大型問題的 CP 模型複雜度較高，因此將求解條件提高，錯誤次數最高設定到 4,000,000 次。如表 4-7 顯示，本研究研擬的 CP 求解模型求解 L 群問題結果皆優於或是等於(李佳陽, 2019)文獻展示的最佳解。求解L₈₋₈₋₅問題時，本 CP 模型在錯誤次數 1,000,000 次的條件下，就能求出比GA⁺演算法更好的結果。

L₁₅₋₈₋₆問題的變數和限制式數量接近L₈₋₈₋₅的兩倍，但本 CP 模型能在短時間內求出L₁₅₋₈₋₆的全域最佳解，其最大完工時間是 2,864，卻無法確認L₈₋₈₋₅的最大完工時間 183 是否為全域最佳解。因此限制規劃求解模型是否能求出最佳解不一定與問題複雜度有關。求解結果顯示限制規劃模型求解L₁₅₋₈₋₆問題時，在錯誤次數 1,000,000 次內就能求得全域最佳解，結果與GA⁺法的結果相同，但限制規劃模型的求解時間只需 28 秒，比GA⁺法的 4 分 25 秒大幅降低。

L₁₅₋₁₀₋₆問題是複雜度最高的標竿問題，變數數量高達 27,109 個，且限制式數量也高達 26,661 個，見表 4-6。求解結果顯示在錯誤次數 1,000,000 次下，限制規劃法的求解結果 212 就比GA⁺法的 30 次的平均求解結果 219 佳；而在錯誤次數 4,000,000 次時，本 CP 模型求得目前最佳解其最大完工時間是 201。圖 4-11 至圖 4-13 是本研究求解 L 群問題的最佳排程結果的甘特圖。

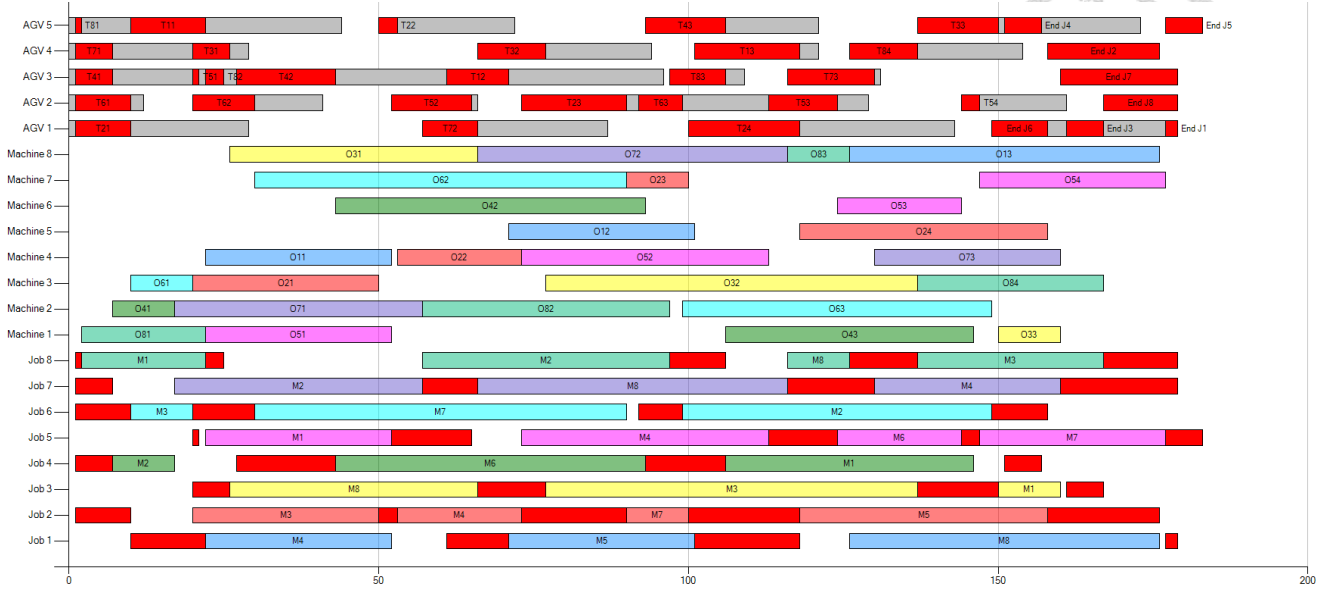


圖 4-11 L₈₋₈₋₅排程結果甘特圖

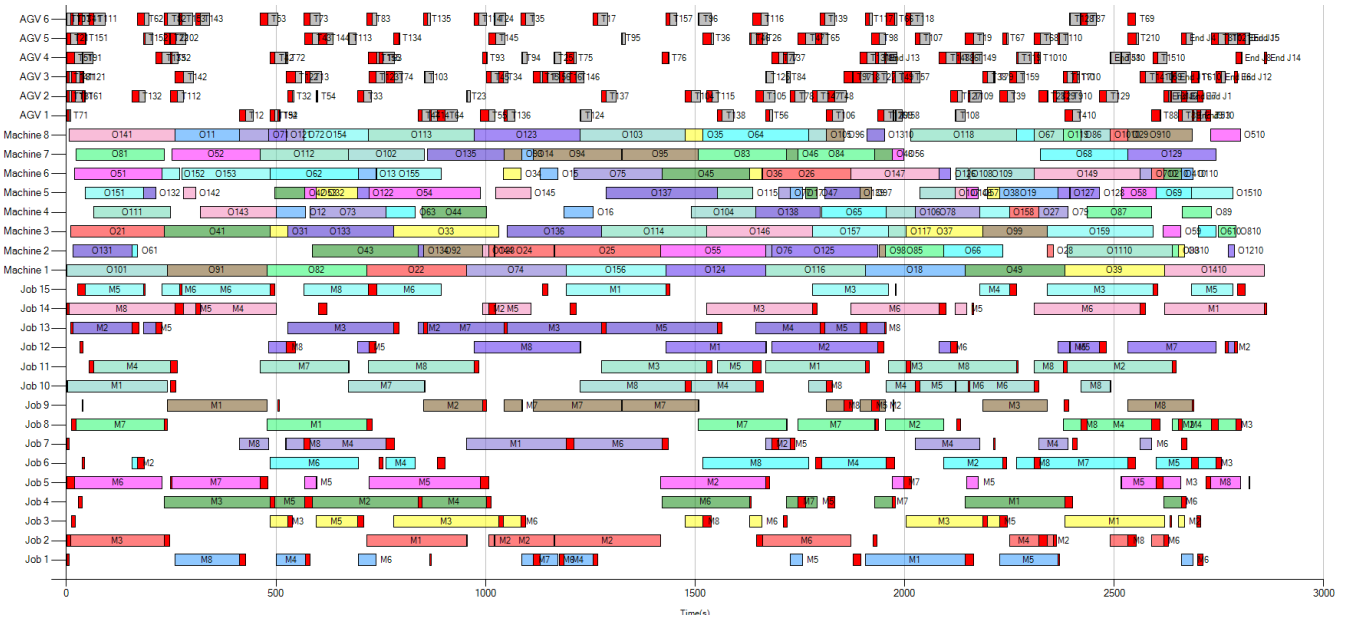


圖 4-12 L₁₅₋₈₋₆排程結果甘特圖

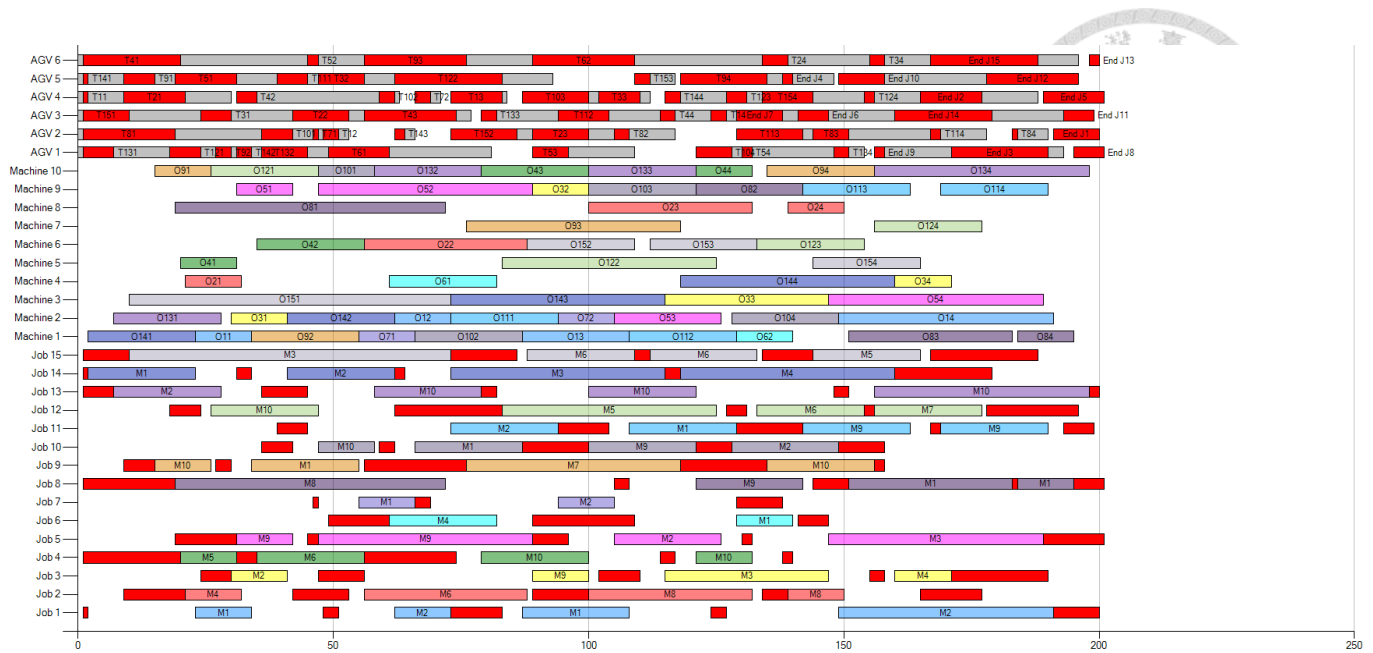


圖 4-13 $L_{15-10-6}$ 排程結果甘特圖

限制規劃模型使用不同目標函式的情境測試

實務上彈性製造系統生產排程的目標可能不同，因此本測試在探討限制規劃模型分別使用 Minimize C_{max} 及 Minimize C_{total} 兩種不同目標下，各標竿問題求解結果，其排程的產品最大完工時間、產品總完工時間、AGV 平均稼動率、和機台平均稼動率是否有明顯差異，求解得到的數值再與(李佳陽, 2019)文獻中展示的 GA^+ 法比較。本測試的 L 型標竿問題皆採用直線型機台布置，是(李佳陽, 2019)中測試的 \bar{L}_{8-8-5} 、 $\bar{L}_{15-10-6}$ 、和 \bar{L}_{15-8-6} 三個問題。

求解 \bar{L}_{8-8-5} 和 \bar{L}_{15-8-6} 問題，CP 模型的錯誤次數設定 2,000,000 次。因 $\bar{L}_{15-10-6}$ 的複雜度較高，CP 模型的錯誤次數設定成 4,000,000 次。 GA^+ 的加工作業段染色體採 Order 交配和 Reciprocal 突變，AGV 段和機台段皆採 SVD 交配及泛用型隨機基因值重設法。 GA^+ 求解 L 群問題的染色體數量設定為 100，交配率 0.8，突變率 0.8，初始貪婪解占比設為 0.01。求解 \bar{L}_{8-8-5} 和 \bar{L}_{15-8-6} 問題時停止條件是目標函式呼叫達上限 2,000,000；而 $\bar{L}_{15-10-6}$ 則設成 4,000,000。

表 4-8 是求解 \bar{L}_{8-8-5} 、 $\bar{L}_{15-10-6}$ 、和 \bar{L}_{15-8-6} 三個問題時 CP 模型的變數和限制式數量。表 4-9 列出本 CP 模型求解這三個問題得到的最佳排程的值。表中展

示限制規劃法的求解結果，並記錄(李佳陽, 2019)文獻中GA⁺演算法的最佳解以及機台和 AGV 的平均稼動率。表中的*代表求解的結果是問題的最佳解。

表 4-8 \bar{L}_{8-8-5} 、 $\bar{L}_{15-10-6}$ 、和 \bar{L}_{15-8-6} 問題的變數和限制式數量

CP模型參數	標竿問題		
	\bar{L}_{8-8-5}	\bar{L}_{15-8-6}	$\bar{L}_{15-10-6}$
變數數量	4,810	8,532	2,7109
限制式數量	4,689	8,381	2,6661

表 4-9 \bar{L}_{8-8-5} 、 $\bar{L}_{15-10-6}$ 、和 \bar{L}_{15-8-6} 問題的限制規劃法及GA⁺法求解結果

標竿問題	各指標值	GA ⁺ 法	CP模型	
			Minimize C_{max}	Minimize C_{total}
\bar{L}_{8-8-5}	C_{max}	223	209	285
	C_{total}	-	1,521	1,500
	機台平均稼動率	0.36	0.6	0.62
	AGV平均稼動率	0.54	0.79	0.77
\bar{L}_{15-8-6}	C_{max}	2,864	2,864	3,435
	C_{total}	-	40,418	27,947
	機台平均稼動率	0.65	0.8	0.56
	AGV平均稼動率	0.29	0.32	0.28
$\bar{L}_{15-10-6}$	C_{max}	278	268	304
	C_{total}	-	3,500	3,148
	機台平均稼動率	0.51	0.65	0.66
	AGV平均稼動率	0.62	0.92	0.88

測試結果顯示針對直線型機台布置的 \bar{L}_{8-8-5} 、 \bar{L}_{15-8-6} 、和 $\bar{L}_{15-10-6}$ 問題，CP模型使用Minimize C_{max} 目標函式，雖然產品的總完工時間較長，但 AGV 的平均

稼動率都優於使用 Minimize C_{total} 目標函式。就機台的平均稼動率而言，兩種目標函式沒有一定較佳的選擇。

不論目標函式為何，CP 模型求解 \bar{L}_{8-8-5} 和 $\bar{L}_{15-10-6}$ 問題的結果都優於 GA⁺ 演算法，且機台和 AGV 的平均稼動率也明顯提升。使用限制規劃法求解 \bar{L}_{15-8-6} 問題時，只有使用 Minimize C_{max} 目標函式時，求解結果才會較 GA⁺ 演算法佳。

圖 4-14 和圖 4-15 分別是 CP 模型使用 Minimize C_{max} 及 Minimize C_{total} 目標函式求解 $\bar{L}_{15-10-6}$ 問題的排程結果甘特圖。比較兩張圖可看出 CP 模型使用目標函式 Minimize C_{max} 時，同一產品的加工作業排程間有較多產品的等待時間。但機台的使用效率很高，少有閒置時間，機台平均稼動率高達 0.8。當 CP 模型使用 Minimize C_{total} 目標函式時，同一產品的加工作業較緊密相連，因此求得的總完工時間 27,947 顯著低於另一目標函式求得的 40,418。由排程結果也可得知，當目標函式設定為 Minimize C_{total} 時，CP 求解引擎會先排完同一產品的加工作業再排其他產品的加工作業。如圖 4-15 所示，產品 10 的第一個加工作業從 2,382 秒才開始執行。此現象會造成機台在到後來才開始處理那些加工作業，延後機台執行完所有加工作業的時間，造成機台稼動率降低。但從另一方向來看，該機台閒置時段完整且長，實務上可安排其他批的產品製造；意即支援他批製造的彈性較佳。若產品訂單來自多個客戶，則選擇 Minimize C_{total} 目標函式會是較佳的選擇，因產品的加工作業排程較集中，可快速加工完成一產品的訂單，縮短產品能交付給客戶的時間，系統即可排定其他產品訂單。

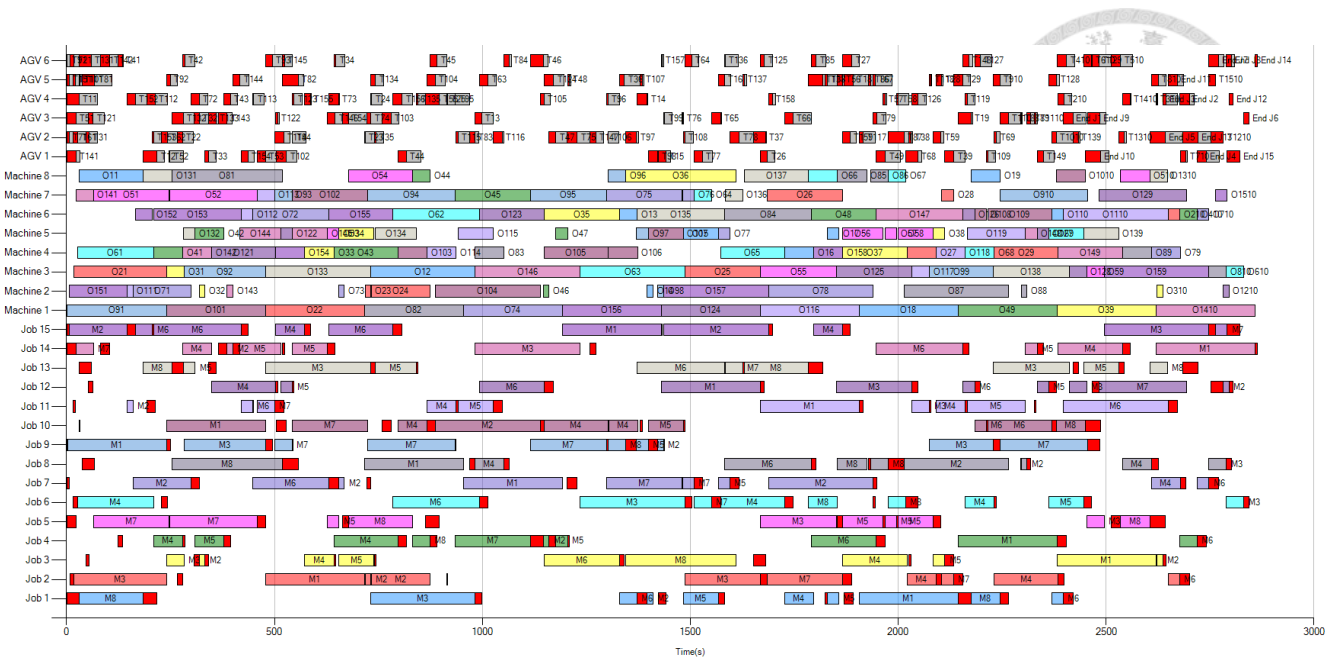


圖 4-14 CP 模型使用 Minimize C_{max} 目標函式求解 L_{15-8-6} 的排程結果甘特圖

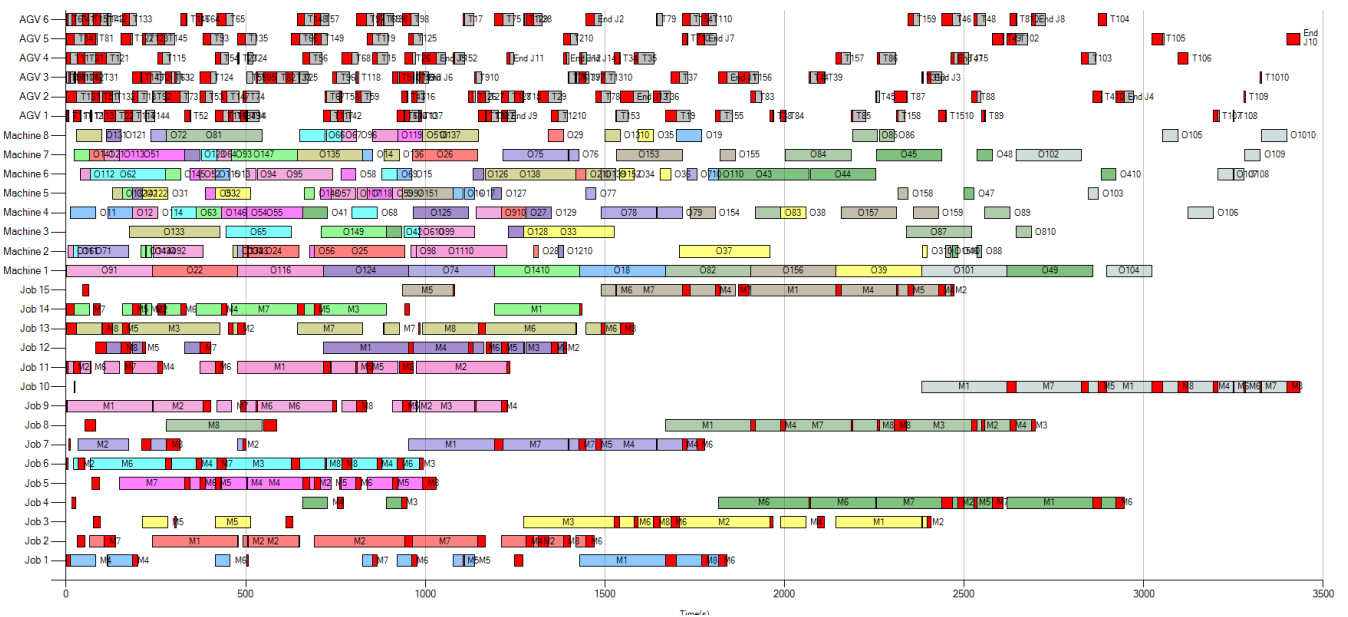


圖 4-15 CP 模型使用 Minimize C_{total} 目標函式求解 L_{15-8-6} 的排程結果甘特圖

4.4 小結

數值測試結果顯示本限制規劃求解模型在各型標竿問題求解結果都較(Liang et al., 2012)文獻中的 RKPSO 法及(李佳陽, 2019)文獻中的 GA^+ 演算法求解結果佳。而本研究研擬的限制規劃求解模型能在錯誤次數較低時，求得比 GA^+ 演算法的平

均最大完工時間更好的解。當放寬解搜尋條件時，意即提高 CP 模型設定的錯誤次數，限制規劃法的求解結果均優於GA⁺演算法。

本 CP 求解模型提供兩種優化目標函式，能讓決策者選擇不同的求解目標。CP 模型使用 Minimize C_{max} 目標函式能求得最小的最大完工時間，因此若希望提升搬運設備的稼動率，或是產品訂單來自於同一個客戶有相同交期時，使用 Minimize C_{max} 目標函式會是較佳的選擇。而 CP 模型使用 Minimize C_{total} 目標函式讓系統能在短時間內生產更多樣的產品。

本限制規劃求解模型求解 L 群大型標竿問題時，不論是採用 U 型或是直線型的機台布置，本 CP 模型求得的最大完工時間，數值都小於或等於GA⁺演算法的最佳解，驗證本研究研擬的 CP 求解模型求解大型問題的效能。

第5章 結論與未來研究建議




本研究研擬了彈性零工生產與搬運排程問題的限制規劃求解模型，求解目標是找到一個最佳的排程使產品的最大完工時間最小化或完工時間加總最小化。除了研擬限制規劃模型外，本研究使用 C# 程式語言在 Microsoft .Net Framework 4.6.1 平台下，以 IBM ILOG® CPLEX 的 CP Optimizer 建模套件實作此求解模型的求解系統。為了驗證限制規劃法的求解效能，本研究也以文獻中的二群問題為標竿問題，進行不同情境的範例測試。成果比較顯示本限制規劃求解法較文獻使用的 GA⁺ 及 RKPSO 啟發式演算法佳。本章節說明結論和未來建議。

5.1 結論

本研究已研擬出適切可行的彈性零工生產與搬運排程問題的限制規劃求解模型，並以最小化產品的最大完工時間產品的總完工時間和，供排程目標選用。經標竿問題試驗和比較，歸納出以下結論：

1. Y 群中小型標竿問題中，本研究研擬的限制規劃求解模型皆能求得比 (Liang et al., 2012) 文獻更佳的解，且找到的解與 (李佳陽, 2019) 文獻中展示的最佳解的最大完工時間相同。驗證了 GA⁺ 演算法有能力求得彈性零工生產與搬運排程問題的最佳解，但 (Liang et al., 2012) 文獻的 RKPSO 法求解中小型問題效能和成果都較差。
2. L 群大型標竿問題中，本限制規劃模型求得的最大完工時間，數值都小於或等於 GA⁺ 演算法的最佳解，效能確比啟發式求解法佳。
3. 本限制規劃法能在短時間內找到比 GA⁺ 演算法更好的解。若依照問題複雜度增高，放寬搜尋條件時(提高錯誤次數的設定)，限制規劃法求解結果都會優於或等於 GA⁺ 演算法。
4. 提供最小化最大完工時間以及最小化總完工時間兩種目標函式的限制規劃求解模型讓使用者依據顧客需求選擇使用的模型，分析不同目標函式對最大完工時間、總完工時間、和機台與 AGV 的平均稼動率的影響。

- 
5. 本問題是考量工件搬運的彈性零工生產問題，因複雜度過高，難以建構數學規劃模型。啟發式演算法則須針對問題設定調整系統參數，否則容易陷入區域最佳解。因限制規劃提供多種變數和運算子，加上 IBM ILOG 的 CP 求解引擎特殊的搜尋法，適合用來求解排程問題，且都能求得不錯的解，解決了上述其他求解法的問題。

5.2 未來研究建議

本研究提出以下建議，提供未來相關研究可繼續研究的方向：

1. 本研究的 AGV 以固定的速度運行在串接的軌道上，不考慮交通阻塞的情形。但實務上 AGV 在相同的軌道上行走可能會發生碰撞，未來研究可將此狀況納入考量。
2. 本研究假設倉儲中心和機台的空間夠大且無容量限制，可供無限多的 AGV 和工件停駐或儲存。實務上倉儲中心和機台的容量有限，會有發生系統鎖死的情況。未來研究可加入空間容量限制，更進一步迎合工廠實務。
3. 本研究是靜態產品排程規劃，也就是在 CP 求解引擎求解限制規劃模型前，預先確定所有加工作業的資訊，再進行一次性的排程安排。實務上工廠生產時可能會有許多突發狀況，例如：重新加工、緊急插單、取消訂單等。未來研究可設計成動態排程規劃系統，可即時接收產品插單，立即進行重排程。
4. 限制規劃模型的限制條件不只有一種表達方式，未來可利用 CP Optimizer 的其他運算子組成限制式，分析不同限制式對模型求解效率的影響，嘗試建構效率更高的限制規劃模型。

參考文獻



Bilge, Ü., & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, 43(6), 1058-1070. doi:10.1287/opre.43.6.1058

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3), 557-581. doi:Doi 10.1016/S0377-2217(98)00364-6

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations research*, 41(3), 157-183.

Demir, Y., & Kürşat İşleyen, S. (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, 37(3), 977-988. doi:<https://doi.org/10.1016/j.apm.2012.03.020>

Dincbas, M., Hentenryck, P. V., Simonis, H., Aggoun, A., Graf, T., & Berthier, F. (1988). *The Constraint Logic Programming Language CHIP*.

Ham, A. (2020). Transfer-robot task scheduling in job shop. *International Journal of Production Research*, 1-11. doi:10.1080/00207543.2019.1709671

Haralick, R. M., & Elliott, G. L. (1980). Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3), 263-313. doi:[https://doi.org/10.1016/0004-3702\(80\)90051-X](https://doi.org/10.1016/0004-3702(80)90051-X)

Jaffar, J., & Lassez, J.-L. (1987). Constraint logic programming. Paper presented at the Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, Munich, West Germany. <https://doi.org/10.1145/41625.41635>

Kacem, I., Hammadi, S., & Pierre, B. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(1), 1-13.

Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A general algorithm for short-term scheduling of batch operations—I. MILP formulation. *Computers & Chemical Engineering*, 17(2), 211-227. doi:[https://doi.org/10.1016/0098-1354\(93\)80015-F](https://doi.org/10.1016/0098-1354(93)80015-F)

Ku, W. Y., & Beck, J. C. (2016). Mixed Integer Programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73, 165-173. doi:10.1016/j.cor.2016.04.006

Laborie, P. (2009). *IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems*, Berlin, Heidelberg.

Laborie, P., Rogerie, J., Shaw, P., & Vilím, P. (2018). IBM ILOG CP optimizer for scheduling. *Constraints*, 23(2), 210-250. doi:10.1007/s10601-018-9281-x

Liang, Y., Lin, L., Gen, M., & Chien, C.-F. (2012). A hybrid evolutionary algorithm for FMS optimization with AGV dispatching. In *Proceedings of the 42nd international conference on computers and industrial engineering*, 296.291-296.214.

Liao, C. J., & You, C. T. (1992). An Improved Formulation for the Job-Shop Scheduling Problem. *Journal of the Operational Research Society*, 43(11), 1047-1054. Retrieved from <Go to ISI>://WOS:A1992JX29300004

Lin, L., & Gen, M. (2009). A random key-based genetic algorithm for AGV dispatching in FMS. *IJMTM*, 16, 58-75. doi:10.1504/IJMTM.2009.021504

Manne, A. S. (1960). On the Job-Shop Scheduling Problem. *Operations Research*, 8(2), 219-223. doi:DOI 10.1287/opre.8.2.219

Ozguven, C., Ozbakir, L., & Yavuz, Y. (2010). Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling*, 34(6), 1539-1548. doi:10.1016/j.apm.2009.09.002

Sabin, D., & Freuder, E. C. (1997). Understanding and improving the MAC algorithm. *Principles and Practice of Constraint Programming - Cp 97*, 1330, 167-181. Retrieved from <Go to ISI>://WOS:000077602600013

Sawik, T. (1996). A multilevel machine and vehicle scheduling in a flexible manufacturing system. *Mathematical and Computer Modelling*, 23(7), 45-57. doi:Doi 10.1016/0895-7177(96)00028-3

Simonis, H. (2008). A Problem Classification Scheme for Finite Domain Constraint Solving.

Tsang, E. (1995). Foundations of constraint satisfaction. *Journal of the Operational Research Society*, 46(5), 666.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2), 131-140. doi:10.1002/nav.3800060205

Xiao, H., Wu, X., Zeng, Y., & Zhai, J. (2020). A CEGA-Based Optimization Approach for Integrated Designing of a Unidirectional Guide-Path Network and Scheduling of AGVs. *Mathematical Problems in Engineering*, 2020, 3961409. doi:10.1155/2020/3961409

Xie, C., & Allen, T. T. (2015). Simulation and experimental design methods for job shop scheduling with material handling: a survey. *International Journal of Advanced Manufacturing Technology*, 80(1-4), 233-243. doi:10.1007/s00170-015-6981-x

李佳陽. (2019). 彈性零工生產與搬運排程問題及啟發式求解法. 國立臺灣大學, Available from Airiti AiritiLibrary database. (2019 年)

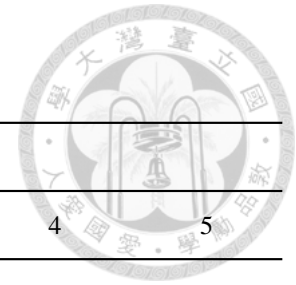
附錄一



Y₉₋₅₋₄、Y₉₋₅₋₅標竿問題各加工作業對照不同機台加工時間

產品	加工作業	機台				
		1	2	3	4	5
1	$o_{1,1}$	20	-	60	-	120
	$o_{1,2}$	100	120	-	90	40
	$o_{1,3}$	-	-	80	100	70
2	$o_{2,1}$	20	-	60	-	120
	$o_{2,2}$	-	-	80	100	70
3	$o_{3,1}$	20	-	60	-	120
	$o_{3,2}$	-	90	80	70	-
	$o_{3,3}$	70	70	100	-	80
4	$o_{4,1}$	100	120	-	90	40
	$o_{4,2}$	-	90	80	70	-
	$o_{4,3}$	20	-	60	-	120
	$o_{4,4}$	70	70	100	-	80
5	$o_{5,1}$	100	120	-	90	40
	$o_{5,2}$	70	70	100	-	80
6	$o_{6,1}$	100	120	-	90	40
	$o_{6,2}$	-	90	80	70	-
	$o_{6,3}$	-	-	80	100	70
	$o_{6,4}$	70	70	100	-	80
7	$o_{7,1}$	20	-	60	-	120
	$o_{7,2}$	70	70	100	-	80
8	$o_{8,1}$	20	-	60	-	120
	$o_{8,2}$	-	90	80	70	-
	$o_{8,3}$	100	120	-	90	40
	$o_{8,4}$	-	-	80	100	70
9	$o_{9,1}$	20	-	60	-	120
	$o_{9,2}$	-	90	80	70	-
	$o_{9,3}$	-	-	80	100	70

Y₉₋₅₋₄、Y₉₋₅₋₅標竿問題各機台與倉儲中心的搬運時間



D到P / P到D	倉儲中心			機台			
	0	1	2	3	4	5	
倉儲中心	0	1/1	1/7	8/13	14/18	16/23	18/20
	1	13/18	3/3	2/9	8/14	10/19	13/18
	2	18/22	22/28	2/2	2/7	4/12	12/18
機台	3	13/11	17/22	24/29	1/1	1/6	7/11
	4	8/14	12/20	18/26	24/29	3/3	2/10
	5	5/7	9/12	15/18	19/23	23/28	2/2

附錄二



Kacem 8x8 部分彈性零工生產標竿問題

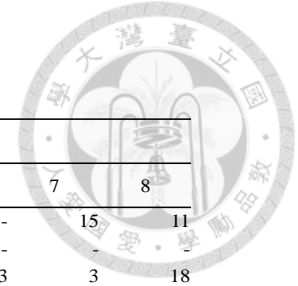
產品	加工作業	機台							
		1	2	3	4	5	6	7	8
1	$o_{1,1}$	5	3	5	3	3	-	10	9
	$o_{1,2}$	10	-	5	8	3	9	9	6
	$o_{1,3}$	-	10	-	5	6	2	4	5
2	$o_{2,1}$	5	7	3	9	8	-	9	-
	$o_{2,2}$	-	8	5	2	6	7	10	9
	$o_{2,3}$	-	10	-	5	6	4	1	7
	$o_{2,4}$	10	8	9	6	4	7	-	-
3	$o_{3,1}$	10	-	-	7	6	5	2	4
	$o_{3,2}$	-	10	6	4	8	9	10	-
	$o_{3,3}$	1	4	5	6	-	10	-	7
4	$o_{4,1}$	3	1	6	5	9	7	8	4
	$o_{4,2}$	12	11	7	8	10	5	6	9
	$o_{4,3}$	4	6	2	10	3	9	5	7
5	$o_{5,1}$	3	6	7	8	9	-	10	-
	$o_{5,2}$	10	-	7	4	9	8	6	-
	$o_{5,3}$	-	9	8	7	4	2	7	-
	$o_{5,4}$	11	9	-	6	7	5	3	6
6	$o_{6,1}$	6	7	1	4	6	9	-	10
	$o_{6,2}$	11	-	9	9	9	7	6	4
	$o_{6,3}$	10	5	9	10	11	-	10	-
7	$o_{7,1}$	5	4	2	6	7	-	10	-
	$o_{7,2}$	-	9	-	9	11	9	10	5
	$o_{7,3}$	-	8	9	3	8	6	-	10
8	$o_{8,1}$	2	8	5	9	-	4	-	10
	$o_{8,2}$	7	4	7	8	9	-	10	-
	$o_{8,3}$	9	9	-	8	5	6	7	1
	$o_{8,4}$	9	-	3	7	1	5	8	-

Kacem 15x10 完全彈性零工生產標竿問題



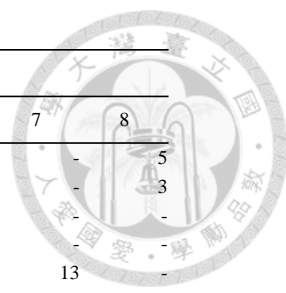
產品	加工作業	機台									
		1	2	3	4	5	6	7	8	9	10
1	$o_{1,1}$	1	4	6	9	3	5	2	8	9	4
	$o_{1,2}$	1	1	3	4	8	10	4	11	4	3
	$o_{1,3}$	2	5	1	5	6	9	5	10	3	2
	$o_{1,4}$	10	4	5	9	8	4	15	8	4	4
2	$o_{2,1}$	4	8	7	1	9	6	1	10	7	1
	$o_{2,2}$	6	11	2	7	5	3	5	14	9	2
	$o_{2,3}$	8	5	8	9	4	3	5	3	8	1
	$o_{2,4}$	9	3	6	1	2	6	4	1	7	2
3	$o_{3,1}$	7	1	8	5	4	9	1	2	3	4
	$o_{3,2}$	5	10	6	4	9	5	1	7	1	6
	$o_{3,3}$	4	2	3	8	7	4	6	9	8	4
	$o_{3,4}$	7	3	12	1	6	5	8	3	5	2
4	$o_{4,1}$	6	2	5	4	1	2	3	6	5	4
	$o_{4,2}$	8	5	7	4	1	2	36	5	8	5
	$o_{4,3}$	9	6	2	4	5	1	3	6	5	2
	$o_{4,4}$	11	4	5	6	2	7	5	4	2	1
5	$o_{5,1}$	6	9	2	3	5	8	7	4	1	2
	$o_{5,2}$	5	4	6	3	5	2	28	7	4	5
	$o_{5,3}$	6	2	4	3	6	5	2	4	7	9
	$o_{5,4}$	6	5	4	2	3	2	5	4	7	5
6	$o_{6,1}$	4	1	3	2	6	9	8	5	4	2
	$o_{6,2}$	1	3	6	5	4	7	5	4	6	5
7	$o_{7,1}$	1	4	2	5	3	6	9	8	5	4
	$o_{7,2}$	2	1	4	5	2	3	5	4	2	5
8	$o_{8,1}$	2	3	6	2	5	4	1	5	8	7
	$o_{8,2}$	4	5	6	2	3	5	4	1	2	5
	$o_{8,3}$	3	5	4	2	5	49	8	5	4	5
	$o_{8,4}$	1	2	36	5	2	3	6	4	11	2
9	$o_{9,1}$	6	3	2	22	44	11	10	23	5	1
	$o_{9,2}$	2	3	2	12	15	10	12	14	18	16
	$o_{9,3}$	20	17	12	5	9	6	4	7	5	6
	$o_{9,4}$	9	8	7	4	5	8	7	4	56	2
10	$o_{10,1}$	5	8	7	4	56	3	2	5	4	1
	$o_{10,2}$	2	5	6	9	8	5	4	2	5	4
	$o_{10,3}$	6	3	2	5	4	7	4	5	2	1
	$o_{10,4}$	3	2	5	6	5	8	7	4	5	2
11	$o_{11,1}$	1	2	3	6	5	2	1	4	2	1
	$o_{11,2}$	2	3	6	3	2	1	4	10	12	1
	$o_{11,3}$	3	6	2	5	8	4	6	3	2	5
	$o_{11,4}$	4	1	45	6	2	4	1	25	2	4
12	$o_{12,1}$	9	8	5	6	3	6	5	2	4	2
	$o_{12,2}$	5	8	9	5	4	75	63	6	5	21
	$o_{12,3}$	12	5	4	6	3	2	5	4	2	5
	$o_{12,4}$	8	7	9	5	6	3	2	5	8	4
13	$o_{13,1}$	4	2	5	6	8	5	6	4	6	2
	$o_{13,2}$	3	5	4	7	5	8	6	6	3	2
	$o_{13,3}$	5	4	5	8	5	4	6	5	4	2
	$o_{13,4}$	3	2	5	6	5	4	8	5	6	4
14	$o_{14,1}$	2	3	5	4	6	5	4	85	4	5
	$o_{14,2}$	6	2	4	5	8	6	5	4	2	6
	$o_{14,3}$	3	25	4	8	5	6	3	2	5	4
	$o_{14,4}$	8	5	6	4	2	3	6	8	5	4
15	$o_{15,1}$	2	5	6	8	5	6	3	2	5	4
	$o_{15,2}$	5	6	2	5	4	2	5	3	2	5
	$o_{15,3}$	4	5	2	3	5	2	8	4	7	5
	$o_{15,4}$	6	2	11	14	2	3	6	5	4	8

Brandimarte 15x8 部分彈性零工生產標竿問題

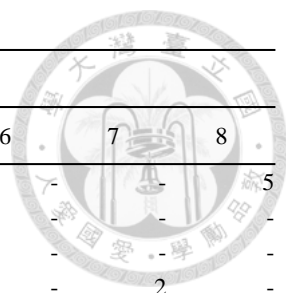


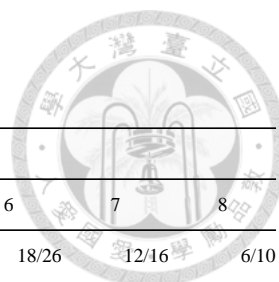
產品	加工作業	機台							
		1	2	3	4	5	6	7	8
1	$o_{1,1}$	-	-	-	5	19	-	15	11
	$o_{1,2}$	-	-	18	5	-	-	-	-
	$o_{1,3}$	-	-	16	-	-	3	3	18
	$o_{1,4}$	-	1	19	-	7	-	2	-
	$o_{1,5}$	-	-	-	-	6	3	-	-
	$o_{1,6}$	-	-	-	5	2	-	-	18
	$o_{1,7}$	-	-	-	-	2	-	-	-
	$o_{1,8}$	17	-	-	-	-	-	-	-
	$o_{1,9}$	12	10	14	-	10	-	-	5
	$o_{1,10}$	-	-	-	-	-	2	15	19
2	$o_{2,1}$	-	-	16	-	-	11	3	18
	$o_{2,2}$	17	-	-	-	-	-	-	-
	$o_{2,3}$	-	1	-	13	-	-	-	-
	$o_{2,4}$	12	10	14	-	10	-	-	5
	$o_{2,5}$	9	18	13	11	-	18	-	-
	$o_{2,6}$	-	-	-	-	-	15	13	-
	$o_{2,7}$	-	-	-	5	19	-	15	11
	$o_{2,8}$	-	1	19	-	7	-	2	-
	$o_{2,9}$	7	-	-	11	-	13	-	3
	$o_{2,10}$	-	-	-	-	-	2	15	19
3	$o_{3,1}$	-	-	3	-	5	-	-	-
	$o_{3,2}$	-	1	19	-	7	-	2	-
	$o_{3,3}$	-	-	18	5	-	-	-	-
	$o_{3,4}$	-	-	-	-	6	3	-	-
	$o_{3,5}$	7	-	-	11	-	13	-	3
	$o_{3,6}$	-	-	-	-	-	2	15	19
	$o_{3,7}$	9	18	13	11	-	18	-	-
	$o_{3,8}$	-	-	-	5	2	-	-	18
	$o_{3,9}$	17	-	-	-	-	-	-	-
	$o_{3,10}$	-	1	-	13	-	-	-	-
4	$o_{4,1}$	-	-	18	5	-	-	-	-
	$o_{4,2}$	-	-	3	-	5	-	-	-
	$o_{4,3}$	9	18	13	11	-	18	-	-
	$o_{4,4}$	7	-	-	11	-	13	-	3
	$o_{4,5}$	-	-	-	-	-	15	13	-
	$o_{4,6}$	-	1	19	-	7	-	2	-
	$o_{4,7}$	-	-	-	-	2	-	-	-
	$o_{4,8}$	-	-	16	-	-	11	3	18
	$o_{4,9}$	17	-	-	-	-	-	-	-
	$o_{4,10}$	-	-	-	-	6	3	-	-
5	$o_{5,1}$	-	-	-	-	-	15	13	-
	$o_{5,2}$	-	-	-	-	-	2	15	19
	$o_{5,3}$	-	-	-	-	2	-	-	-
	$o_{5,4}$	-	-	-	5	19	-	15	11
	$o_{5,5}$	9	18	13	11	-	18	-	-
	$o_{5,6}$	-	1	19	-	7	-	2	-
	$o_{5,7}$	-	-	-	5	2	-	-	18
	$o_{5,8}$	-	-	-	-	6	3	-	-
	$o_{5,9}$	-	-	3	-	5	-	-	-
	$o_{5,10}$	12	10	14	-	10	-	-	5
6	$o_{6,1}$	-	1	-	13	-	-	-	-
	$o_{6,2}$	-	-	-	-	-	15	13	-
	$o_{6,3}$	-	-	18	5	-	-	-	-
	$o_{6,4}$	-	-	16	-	-	11	3	18
	$o_{6,5}$	9	18	13	11	-	18	-	-
	$o_{6,6}$	12	10	14	-	10	-	-	5
	$o_{6,7}$	7	-	-	11	-	13	-	3
	$o_{6,8}$	-	-	-	5	19	-	15	11
	$o_{6,9}$	-	-	-	-	6	3	-	-
	$o_{6,10}$	-	-	3	-	5	-	-	-

產品	加工作業	機台							
		1	2	3	4	5	6	7	8
7	07,1	12	10	14	-	10	-	-	5
	07,2	7	-	-	11	-	13	-	3
	07,3	-	1	-	13	-	-	-	-
	07,4	17	-	-	-	-	-	-	-
	07,5	-	-	-	-	-	15	13	-
	07,6	-	1	19	-	7	-	2	-
	07,7	-	-	-	-	2	-	-	-
	07,8	9	18	13	11	-	18	-	-
	07,9	-	-	18	5	-	-	-	-
	07,10	-	-	-	-	-	2	15	19
8	08,1	-	-	-	-	-	2	15	19
	08,2	17	-	-	-	-	-	-	-
	08,3	-	-	-	5	19	-	15	11
	08,4	-	-	-	-	-	15	13	-
	08,5	12	10	14	-	10	-	-	5
	08,6	7	-	-	11	-	13	-	3
	08,7	9	18	13	11	-	18	-	-
	08,8	-	1	-	13	-	-	-	-
	08,9	-	-	18	5	-	-	-	-
	08,10	-	-	3	-	5	-	-	-
9	09,1	17	-	-	-	-	-	-	-
	09,2	12	10	14	-	10	-	-	5
	09,3	-	-	16	-	-	11	3	18
	09,4	-	-	-	-	-	2	15	19
	09,5	-	-	-	-	-	15	13	-
	09,6	7	-	-	11	-	13	-	3
	09,7	-	-	-	-	2	-	-	-
	09,8	-	1	-	13	-	-	-	-
	09,9	9	18	13	11	-	18	-	-
	09,10	-	-	-	5	19	-	15	11
10	010,1	17	-	-	-	-	-	-	-
	010,2	-	-	-	-	-	15	13	-
	010,3	-	-	-	5	2	-	-	18
	010,4	9	18	13	11	-	18	-	-
	010,5	7	-	-	11	-	13	-	3
	010,6	-	-	18	5	-	-	-	-
	010,7	-	-	-	-	6	3	-	-
	010,8	-	-	-	-	-	2	15	19
	010,9	-	-	16	-	-	11	3	18
	010,10	12	10	14	-	10	-	-	5
11	011,1	-	1	-	13	-	-	-	-
	011,2	-	-	-	-	-	2	15	19
	011,3	-	-	16	-	-	11	3	18
	011,4	-	-	18	5	-	-	-	-
	011,5	-	-	-	-	6	3	-	-
	011,6	17	-	-	-	-	-	-	-
	011,7	-	-	3	-	5	-	-	-
	011,8	-	-	-	5	2	-	-	18
	011,9	12	10	14	-	10	-	-	5
	011,10	9	18	13	11	-	18	-	-
12	012,1	7	-	-	11	-	13	-	3
	012,2	-	-	-	5	2	-	-	18
	012,3	-	-	16	-	-	11	3	18
	012,4	17	-	-	-	-	-	-	-
	012,5	9	18	13	11	-	18	-	-
	012,6	-	-	-	-	-	2	15	19
	012,7	-	-	-	-	2	-	-	-
	012,8	-	-	3	-	5	-	-	-
	012,9	-	-	-	5	19	-	15	11
	012,10	-	1	-	13	-	-	-	-



產品	加工作業	機台							
		1	2	3	4	5	6	7	8
13	<i>o</i> _{13,1}	12	10	14	-	10	-	-	5
	<i>o</i> _{13,2}	-	-	-	-	2	-	-	-
	<i>o</i> _{13,3}	-	-	18	5	-	-	-	-
	<i>o</i> _{13,4}	-	1	19	-	7	-	2	-
	<i>o</i> _{13,5}	-	-	-	-	-	15	13	-
	<i>o</i> _{13,6}	-	-	16	-	-	11	3	18
	<i>o</i> _{13,7}	-	-	-	5	19	-	15	11
	<i>o</i> _{13,8}	9	18	13	11	-	18	-	-
	<i>o</i> _{13,9}	-	-	-	-	6	3	-	-
	<i>o</i> _{13,10}	7	-	-	11	-	13	-	3
14	<i>o</i> _{14,1}	-	-	16	-	-	11	3	18
	<i>o</i> _{14,2}	-	-	-	5	2	-	-	18
	<i>o</i> _{14,3}	-	1	-	13	-	-	-	-
	<i>o</i> _{14,4}	-	1	19	-	7	-	2	-
	<i>o</i> _{14,5}	-	-	-	-	6	3	-	-
	<i>o</i> _{14,6}	-	-	18	5	-	-	-	-
	<i>o</i> _{14,7}	-	-	-	-	-	15	13	-
	<i>o</i> _{14,8}	-	-	-	-	2	-	-	-
	<i>o</i> _{14,9}	9	18	13	11	-	18	-	-
	<i>o</i> _{14,10}	17	-	-	-	-	-	-	-
15	<i>o</i> _{15,1}	12	10	14	-	10	-	-	5
	<i>o</i> _{15,2}	-	-	-	-	6	3	-	-
	<i>o</i> _{15,3}	-	-	-	-	-	15	13	-
	<i>o</i> _{15,4}	-	-	-	5	19	-	15	11
	<i>o</i> _{15,5}	-	-	16	-	-	11	3	18
	<i>o</i> _{15,6}	17	-	-	-	-	-	-	-
	<i>o</i> _{15,7}	9	18	13	11	-	18	-	-
	<i>o</i> _{15,8}	-	-	-	5	2	-	-	18
	<i>o</i> _{15,9}	-	-	18	5	-	-	-	-
	<i>o</i> _{15,10}	-	1	19	-	7	-	2	-



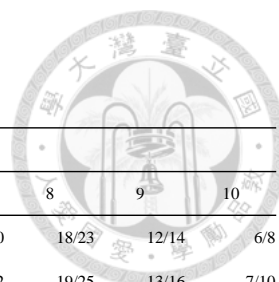


8 機台 U 型布置的搬運時間矩陣

D到P / P到D	倉儲中心				機台					
	0	1	2	3	4	5	6	7	8	
倉儲中心	0	1/1	1/3	6/9	9/13	12/15	19/23	18/26	12/16	6/10
	1	6/8	1/1	3/7	10/15	13/17	20/25	19/28	13/18	7/12
	2	9/13	10/15	2/2	2/6	9/14	17/22	16/25	9/15	9/15
	3	12/16	13/18	13/19	2/2	3/7	12/17	11/20	10/16	11/18
機台	4	19/23	20/25	20/26	17/22	1/1	10/14	11/19	17/23	18/25
	5	18/22	19/24	19/25	16/21	11/15	1/1	3/10	17/22	17/24
	6	12/19	13/21	15/22	9/18	10/18	17/26	5/5	3/11	11/21
	7	6/9	7/11	7/12	8/14	11/16	18/24	17/27	2/2	2/7
	8	2/5	3/8	8/14	11/18	14/20	21/28	20/31	14/21	3/3

8 機台直線型布置的搬運時間矩陣

D到P / P到D	倉儲中心				機台					
	0	1	2	3	4	5	6	7	8	
倉儲中心	0	1/1	1/7	5/11	9/14	11/20	18/26	21/26	22/31	29/42
	1	6/6	1/1	2/4	10/13	13/15	20/25	23/27	24/26	31/37
	2	9/10	9/12	2/2	2/5	9/12	17/22	19/24	20/23	27/34
	3	12/13	12/15	13/17	2/2	4/6	12/18	15/19	16/18	22/29
機台	4	19/19	19/21	20/23	17/19	1/1	5/9	13/17	14/16	20/26
	5	22/25	22/27	23/29	19/25	15/19	4/4	4/9	8/12	14/22
	6	23/24	23/27	24/29	20/25	16/19	14/20	3/3	2/5	11/18
	7	30/30	30/32	30/34	27/30	22/24	20/25	14/18	1/1	4/9
	8	37/41	37/43	38/45	34/41	29/38	27/36	20/28	17/23	5/5



10 機台 U 型布置的搬運時間矩陣

D到P / P到D	倉儲中心					機台						
	0	1	2	3	4	5	6	7	8	9	10	
倉儲中心	0	1/1	1/3	6/8	9/11	12/13	19/23	20/23	19/20	18/23	12/14	6/8
	1	6/7	1/1	3/6	10/13	13/15	20/25	21/25	20/22	19/25	13/16	7/10
	2	9/11	10/13	2/2	2/5	9/12	17/22	17/19	16/22	9/13	9/13	9/15
	3	12/14	13/16	13/17	2/2	3/6	12/17	13/17	12/15	11/17	10/14	11/16
	4	19/20	20/22	20/23	17/19	1/1	5/9	11/15	10/12	11/17	17/20	18/22
機台	5	21/25	22/27	22/28	18/24	14/18	4/4	4/10	7/11	13/21	19/25	20/27
	6	19/22	20/24	20/25	17/21	12/16	10/17	3/3	2/5	11/18	17/23	18/24
	7	18/19	19/21	19/22	16/218	11/13	11/16	12/16	1/1	3/9	17/19	17/21
	8	12/17	13/18	15/19	9/16	10/16	14/22	18/26	17/23	5/5	3/9	11/18
	9	6/8	7/9	7/10	8/12	11/14	18/24	19/24	18/21	17/24	2/2	2/6
	10	2/4	3/7	8/12	11/16	14/17	21/28	22/28	21/25	20/28	14/18	3/3

10 機台直線型布置的搬運時間矩陣

D到P / P到D	倉儲中心					機台						
	0	1	2	3	4	5	6	7	8	9	10	
倉儲中心	0	1/1	1/7	5/11	9/14	11/20	18/26	21/26	22/31	29/42	36/40	36/43
	1	6/6	1/1	2/4	10/13	13/15	20/25	23/27	24/26	31/37	38/41	39/43
	2	9/10	9/12	2/2	2/5	9/12	17/22	19/24	20/23	27/34	34/38	35/40
	3	12/13	12/15	13/17	2/2	4/6	12/18	15/19	16/18	22/29	39/33	30/35
	4	19/19	19/21	20/23	17/19	1/1	5/9	13/17	14/16	20/26	27/30	28/32
機台	5	22/25	22/27	23/29	19/25	15/19	4/4	4/9	8/12	14/22	20/26	21/28
	6	23/24	23/27	24/29	20/25	16/19	14/20	3/3	2/5	11/18	18/22	18/24
	7	30/30	30/32	30/34	27/30	22/24	20/25	14/18	1/1	4/9	17/19	18/21
	8	37/41	37/43	38/45	34/41	29/38	27/36	20/28	17/23	5/5	4/10	11/18
	9	38/39	38/41	39/43	35/39	30/33	28/34	21/26	18/21	17/24	2/2	2/6
	10	40/42	40/44	41/46	37/42	32/36	28/35	23/29	20/24	18/27	13/18	3/3

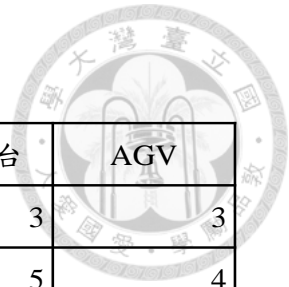
附錄三



Y₃₋₄₋₃最大完工時間 261 的解

產品	加工作業	開始時間	結束時間	加工機台	AGV
1	$o_{1,1}$	2	62	1	2
	$o_{1,2}$	89	109	4	1
	$o_{1,3}$	124	174	2	1
	$o_{1,4}$	189	256	4	3
2	$o_{2,1}$	62	122	1	3
	$o_{2,2}$	132	172	3	3
3	$o_{3,1}$	19	89	4	1
	$o_{3,2}$	109	129	4	3
	$o_{3,3}$	174	224	2	2

Y₉₋₅₋₄最大完工時間 362 的解



產品	加工作業	開始時間	結束時間	加工機台	AGV
1	$o_{1,1}$	40	100	3	3
	$o_{1,2}$	107	147	5	4
	$o_{1,3}$	149	219	5	2
2	$o_{2,1}$	82	102	1	4
	$o_{2,2}$	261	341	3	4
3	$o_{3,1}$	2	22	1	1
	$o_{3,2}$	24	114	2	4
	$o_{3,3}$	204	274	2	3
4	$o_{4,1}$	17	107	4	2
	$o_{4,2}$	110	180	4	1
	$o_{4,3}$	248	268	1	2
	$o_{4,4}$	274	344	2	4
5	$o_{5,1}$	59	99	5	2
	$o_{5,2}$	108	178	1	3
6	$o_{6,1}$	19	59	5	3
	$o_{6,2}$	100	180	3	3
	$o_{6,3}$	181	261	3	1
	$o_{6,4}$	278	348	1	2
7	$o_{7,1}$	62	82	1	4
	$o_{7,2}$	178	248	1	1
8	$o_{8,1}$	22	41	1	1
	$o_{8,2}$	114	204	2	4
	$o_{8,3}$	219	259	5	4
	$o_{8,4}$	261	331	5	1
9	$o_{9,1}$	42	62	1	1
	$o_{9,2}$	180	250	4	2
	$o_{9,3}$	253	353	4	3

Y_{9-5-5} 最大完工時間 362 的解



產品	加工作業	開始時間	結束時間	加工機台	AGV
1	$o_{1,1}$	32	52	1	3
	$o_{1,2}$	99	139	5	3
	$o_{1,3}$	249	319	5	1
2	$o_{2,1}$	72	92	1	4
	$o_{2,2}$	139	209	5	4
3	$o_{3,1}$	2	22	1	2
	$o_{3,2}$	24	114	2	1
	$o_{3,3}$	182	252	1	3
4	$o_{4,1}$	59	99	5	5
	$o_{4,2}$	114	204	2	2
	$o_{4,3}$	252	272	1	5
	$o_{4,4}$	274	344	2	2
5	$o_{5,1}$	17	107	4	3
	$o_{5,2}$	204	274	2	5
6	$o_{6,1}$	19	59	5	4
	$o_{6,2}$	78	158	3	2
	$o_{6,3}$	159	239	3	5
	$o_{6,4}$	272	342	1	1
7	$o_{7,1}$	52	72	1	2
	$o_{7,2}$	112	182	1	4
8	$o_{8,1}$	15	75	3	5
	$o_{8,2}$	107	177	4	3
	$o_{8,3}$	209	249	5	2
	$o_{8,4}$	268	348	3	3
9	$o_{9,1}$	92	112	1	1
	$o_{9,2}$	177	247	4	2
	$o_{9,3}$	250	350	4	5