

國立交通大學

工業工程與管理學系

碩士論文

應用禁忌搜尋法求解供應鏈中在具車容限制下

整合產品製造與成品配送兩階段問題

Applying Tabu Search Algorithm to Solving Production and Distribution

Problems with Vehicle Capacity Constraints in Supply Chains

研究生：呂學君

指導教授：張永佳 博士

中華民國九十六年六月

應用禁忌搜尋法求解供應鏈中在具車容限制下整合產品製造與成品配送兩階段問題

學生：呂學君

指導教授：張永佳

國立交通大學工業工程與管理學系碩士班

摘要

由於及時化(just-in-time)觀念的普及，使得越來越多供應鏈中的企業採用接單式生產(make-to-order)的方式滿足顧客需求，同時專注於降低存貨水準以提升其競爭力。此種現象造成了產品製造與成品配送作業的密切互動，同時增加了研究此類問題的必要性。傳統方法分別最佳化產品製造與成品配送階段的作業並無法最佳化供應鏈的整體利益。因此本研究探討一整合供應鏈中產品製造與成品配送之訂單排程問題，以非等效平行機台(unrelated parallel machine)模擬產品製造階段，並將成品配送階段以考慮車容限制的車輛途程問題(vehicle routing problem, VRP)模擬。訂單之完成時間定義為產品從開始製造至送達顧客手上所花費的時間，以總加權後訂單完成時間為系統的績效指標。在此類績效指標之下，整合產品製造與成品配送兩階段之問題的複雜度為未定多項式難度(NP-Hard)，故本研究設計一禁忌搜尋法(tabu search algorithm)求解整合產品製造與成品配送之兩階段問題，並以電腦模擬方式產生一系列測試問題，由測試結果得知，本研究所設計之演算法可在合理的時間內求得品質佳的近似解，可供相關研究與產業參考。

關鍵字：禁忌搜尋法、非等效平行機台排程、車輛途程問題、製造與配送

Applying Tabu Search Algorithm to Solving Production and Distribution Problems with Vehicle Capacity Constraints in Supply Chains

Department of Industrial Engineering and Management

National Chiao Tung University

Student : Hsuen-Chun Lu

Advisor : Yung-Chia Chang

Abstracts

Due to widespread just-in-time concept, more and more enterprises have chosen make-to-order business model to satisfy customers' demand and focus on reducing their inventory level in order to be competitive. This phenomenon has not only increased the interaction between production and distribution but also raised the practicability of integrating these two stages. Traditional approaches consider production and distribution separately and sequentially may not maximize the total profits in supply chains. Therefore, this research studies an integrated production and distribution scheduling problems at the individual job level. The production stage is modeled by an unrelated parallel machines scheduling problem and distribution stage is modeled by a capacitated vehicle routing problem. The job completion time is defined as the time when it is delivered to its customer. The objective is to minimize total job weighted completion times. The complexity of this problem is NP-hard. We designed a tabu search algorithm to find near-optimal solutions to this problem. Computational analysis based on computer generated problems shows the stability and effectiveness of the proposed approach.

Key words: tabu search algorithm, unrelated parallel machine scheduling , vehicle routing problem, production and distribution

誌謝

在交大工工所兩年的日子轉眼間即將結束，論文之所以能付梓，首先感謝我的指導教授張永佳博士，在研究與論文撰寫的過程中，老師不時指點我正確的方向，使我從中獲益匪淺。同時也感謝唐麗英老師與梁高榮老師在論文計劃書與口試時所提出的意見，使得本論文的內容更加完備。

兩年來無論在生活上或是學術上都受到許多人的幫助，博班的惠誠學長、93級的昱皓、佩君與柏詠都對我十分照顧。同甘共苦的實驗室同學國維、天威、丞舜，很高興我們一齊渡過這兩年的研究生生活並順利畢業。95級的理維、登凱、佩均、佳儒，有你們加入 005，為實驗室帶了許多歡樂，同時也感謝你們對我的幫助。

最後，要感謝我的雙親，沒有你們細心培養就沒有現在的我。還有我兩個優秀的弟弟學增與其錚，謝謝你們的支持，謹以此文獻給所有幫助過我的人。



謹誌於

交通大學工業工程與管理學系

中華民國九十六年七月

目錄

第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	2
1.3 研究目的.....	4
1.4 研究方法.....	4
1.5 研究架構.....	5
第二章 文獻探討.....	6
2.1 供應鏈整合之相關文獻.....	6
2.2 非等效平行機台與車輛途程問題之相關文獻.....	9
2.2.1 非等效平行機台問題之相關文獻.....	9
2.2.2 車輛途程問題之相關文獻.....	10
2.3 巨集式啟發式演算法.....	14
2.3.1 巨集式啟發式演算法.....	14
2.3.2 禁忌搜尋法.....	16
2.3.3 應用禁忌搜尋法之相關文獻.....	18
第三章 問題定義與研究方法.....	21
3.1 問題定義.....	21
3.2 Tabu_PDP 演算法設計.....	24
3.2.1 Tabu_PDP 之起始解.....	24
3.2.2 鄰域定義與交換法則.....	29
3.2.3 Tabu_PDP 之結構與流程.....	30
3.3 範例說明.....	32
第四章 電腦測試與結果分析.....	39
4.1 Tabu_PDP 參數設計.....	39
4.2.1 禁忌表列長度.....	39
4.2.2 停止準則.....	40

4.2 測試 Tabu_PDP 之求解品質	41
4.3 測試問題設計	42
4.4 測試結果及分析	44
4.4.1 Tabu_PDP 運算時間複迴歸分析	46
4.4.2 Tabu_PDP 改善率複迴歸分析	47
4.5 小結	49
第五章 結論與未來發展方向	50
5.1 結論	50
5.2 未來發展方向	50
5.2.2 演算法方面	50
5.2.1 整合產品製造與成品配送兩階段問題方面	51



圖目錄

圖 1 節省法使用前路徑	12
圖 2 節省法使用後路徑	12
圖 3 順時針掃描法之示意圖	13
圖 4 最鄰近解法求解之示意圖	13
圖 5 T_Start 演算法之流程	28
圖 6 Tabu_PDP 演算法流程	30
圖 7 B_1 於機台上加工之甘特圖 (假設開始時間為 0)	35
圖 8 B_2 機台排程之甘特圖	35
圖 9 各機台加工流程圖	36
圖 10 禁忌表列長度與目標函數關係	40
圖 11 迭代次數與目標函數值、運算時間關係	41
圖 12 Tabu_PDP 訂單數、機台數與運算時間關係	47



表目錄

表 1 訂單權重值以及訂單於各機台加工時間.....	32
表 2 各訂單與製造工廠間的運送時間矩陣	32
表 3 權重運送時間矩陣	33
表 4 批次運送順序以及其對應之訂單到達時間.....	34
表 5 B_1 的各機台加工時間	34
表 6 訂單 4 加工後剩餘訂單於各機台加工完成時間	34
表 7 訂單 6 加工後剩餘訂單於各機台加工完成時間	34
表 8 權重最大完工時間	36
表 9 機台加工順序與完工時間表	36
表 10 起始解訂單花費時間與目標函數值表.....	37
表 11 最佳移步之機台加工順序與完工時間.....	38
表 12 最佳移步之訂單運送順序與到達時間.....	38
表 13 最佳移步之訂單花費時間與目標函數值.....	38
表 14 真實理想解與 T_{Start} 、 $Tabu_PDP$ 最佳解之相對誤差	42
表 15 測試問題表	43
表 16 $Tabu_PDP$ 測試結果	45
表 17 $Tabu_PDP$ 運算時間複迴歸分析	46
表 18 $Tabu_PDP$ 改善率複迴歸分析.....	48

第一章 緒論

1.1 研究背景

隨著時代的變遷，供應鏈管理已成為產業中廣為討論的議題，一條供應鏈包含從原物料轉化成最終產品到顧客手中會經過的所有階段。供應鏈管理的主要目的是有效地利用資源來滿足顧客的需求，而近十年來顧客的需求從單純的價格導向變得多元化，產品品質或交貨是否準時皆為滿足顧客需求的因素，因此對企業而言傳統的降低成本策略已不敷使用，取而代之的是如何提升顧客滿意度。由於資訊科技的日新月異，使得電子商務蓬勃發展，讓企業與企業間溝通交流愈加密切，使得供應鏈中不同成員間合作與營運的程度增加，而共同提升顧客滿意度為供應鏈成員可共同追求之目標。

供應鏈管理之目標，在策略面上為滿足顧客需求，然而將策略面(strategic)目標轉化成在操作面(operational)上，供應鏈所需達成的目標是提高產能、減少庫存、降低成本、準時交貨等，這些操作面上的目標，可以透過數學模式表示成最佳化組合之問題。在供應鏈中產品從原料至顧客手中經過的所有企業，如：原料供應商、製造商、物流業者、零售商等都各自負責供應鏈中一個階段的工作，例如在一個典型的供應鏈中，原料供應商供應原物料給製造商，製造商將原物料加工製造成品後，由物流業者運送產品給零售商負責銷售。在每一個階段，企業希望可以追求本身利益最大化，然而各階段追求的利益往往出現互償(trade-off)的情況；舉例來說，假設在供應鏈中有三張訂單需經過產品製造加工與成品配送階段送至顧客手上，對產品製造階段而言，為了減少存貨的觀念並以最大完工時間之和為目標，工廠傾向將一張訂單加工完後，即由成品配送階段接手出貨給顧客；但成品配送階段來說，在未達車容量上限的條件下，一個車次運送

三個顧客的訂單可能會比使用三個車次運送三張訂單所需的發車成本與總運送時間來得少，產品製造階段與成品配送對車輛數量的選擇，就是一種互償的情況。若我們以供應鏈管理的角度切入，對整合產品製造與成品配送之兩階段問題，以總交期最短為目標，最佳化的組合可能會是前兩張訂單一齊配送、第三張獨自配送，只著重單一階段的最佳化反而造成供應鏈整體效益的降低，更進一步造成供應鏈整合上的困難度，這是供應鏈體系不樂見的情況。因此供應鏈不該只著重於單一階段，而是要思考如何強化供應鏈上游與下游間的合作關係，以期達到整體最大的利益，也就是供應鏈管理的精神所在。

1.2 研究動機

過去數十年學者們討論最佳化之問題，大都針對供應鏈中的單一階段，如傳統的排程問題、車輛途程問題以及存貨管理問題等，此種研究方法在供應鏈各階段之企業間以存貨作為緩衝的情況下是有效的，但不論是原物料、半成品或成品對企業來說需要付出一定的存貨成本，影響企業所能運用的資金。隨著及時化(just-in-time)觀念的普及，企業必須減少存貨以提升其競爭力，在為了減少存貨的條件下，只著重單一階段的最佳化組合問題已無法滿足最佳化供應鏈整體利益之目的；再者，現今顧客的需求變動迅速且趨向多樣化，今日倉庫裡堆放的產品未必能符合明日顧客的需求，又或者是產品無法長期保存，因此有越來越多企業選擇考慮整合產品製造與成品配送兩階段，並採用接單式生產(make-to-order)與直接銷售(direct-order)的商業模式。以生鮮食品配送業為例：由於產品無法長時間保存，其模式為業者在接受顧客訂單後，馬上開始生產食品，此為第一階段「產品製造」階段；食品製作完成後，將食品分配到運送工具上開始配送，此為第二階段「成品配送」階段。要如何在合理時間內，規劃食品的製造與外送，使業者能在合理的成本條件下滿足顧客需求，即為一整合兩階段之問題。怎樣能有效整合產品製造階段與成品配送階段，尋求兩階段整體之最大效益，避免各階段因獨自尋求各自最大利益，而使得部份人員或機

器閒置，造成不必要的浪費或損害顧客權益並不是一個容易解決的問題。因此在實務應用面上，如何以整體觀點同時考慮供應鏈上的各階段，的確有深入探討的必要性。

在本研究中探討整合供應鏈中產品製造(production)與成品配送(distribution)的兩階段問題。在產品製造階段，部分製造商在一個工廠可能會同時設置數條生產相同產品的生產線，而由於生產線由於人員或機台的效率不一，使得每條生產線上的產品加工速度皆不相同，因此本研究選定以非等效平行機台(unrelated parallel machine)來模擬產品製造階段；而在成品配送階段，假設成品是經由多部具有容量限制的車輛來運送至數個不同的顧客所在位置，所以可視為具有容量限制的車輛途程問題 (vehicle routing problem, VRP)。本研究所採用之績效衡量指標，為將訂單開始處理到顧客實際收到成品所花費的時間，乘上訂單的權重轉換為成本，以最小化兩階段之總成本作為目標函數。

在這樣的目標函數之下，非等效平行機台或車輛途程問題皆為未定多項式難度(non-deterministic polynomial-time hard, NP-Hard)的問題(Garey and Johnson, 1979)，因此本研究所探討的問題也至少 NP-Hard。目前已有不少學者研究整合供應鏈中產品製造與成品配送兩階段的問題，但早期的研究都是針對供應鏈策略層面性質作討論，Sarmiento and Nagi (1999)回顧了在策略層面上，學者所提出的幾種整合產品製造與成品配送的系統模型，並探討整合之系統對企業可能帶來的影響，然而這些模型都沒有討論到排程、配送等作業層面上的問題或利用數學模型求解的方法。而近期探討作業層面的文獻也只是透過數理證明找出兩階段問題在不同設定情況與目標函數下的複雜度，或利用動態規劃(dynamic programming)與啟發式演算法來求解簡化過的小規模問題或者是特殊個案（如 Lee and Chen 2001, Hall and Potts 2002, Chang and Lee 2003, Li *et al.* 2003, Chang and Lee 2004 等），極少有利用巨集式啟發式演算法(meta-heuristics)來求解問題複

雜度為 NP-hard 的較大規模問題之研究。巨集式啟發式演算法雖然無法求得問題之最佳解，但是能在合理的時間內提供品質佳的近似解，使其更具實用價值，故適合應用於求解像本研究所探討之複雜度高的問題。

禁忌搜尋法(tabu search algorithm)發展至今約有二十多年的歷史(Glover, 1989)，此演算法廣泛地被使用在求解組合最佳化(combinatorial optimization)之問題，對於本研究所探討的產品製造階段中的排程問題(scheduling problem)或是單單針對成品配送階段之車輛途程問題都為其應用之範圍。考量所探討之兩階段問題的複雜度，本研究選擇以禁忌搜尋法為基礎發展演算法來求解此問題，期望藉由禁忌搜尋法所具備之彈性記憶結構與避免重複求解等特性，增加跳脫出區域最佳解的機率以及快速搜尋的機制，以提升搜尋時的效率與近似解的品質。

1.3 研究目的

本研究所探討的問題為整合供應鏈中產品製造與具車輛容量限制的成品配送兩階段問題，主要目的為應用禁忌搜尋法發展出一套能兼具求解速度與近似解品質的演算法，期望能在合理的時間內求得品質佳的近似解。

1.4 研究方法

本研究之研究方法流程如下：

1. 問題定義：確定問題情境，縮小問題範圍，訂定假設條件以符合問題情境。
2. 演算法設計：應用禁忌搜尋法設計一套演算法，在 C++ 之環境下進行程式的撰寫，求解整合非等效平行機台與具車輛容量限制的車輛途程兩階段問題。
3. 演算法測試：探討在不同情境與參數設定下，禁忌搜尋法的求解速度與近似解品質。

4. 結論與建議：根據測試結果作出結論且建議未來可以再深入研究之方向。

1.5 研究架構

本研究共分成五個章節，第一章敘述研究背景動機以及研究目的；第二章回顧供應鏈整合、非等效平行機台、車輛途程問題以及禁忌搜尋法之相關文獻；第三章為研究方法，詳細定義本研究所探討的問題模型與基本假設，並以禁忌搜尋法為架構，設計一個演算法求解整合兩階段之問題；第四章為演算法測試與結果，以分析本研究所提出演算法的求解能力，第五章則是結論與未來發展方向，討論本研究所發現之現象並作出推論，最後針對本研究未盡完善與不足之處提出建議。



第二章 文獻探討

第二章為文獻探討，其中 2.1 小節回顧與供應鏈整合相關的文獻，2.2 小節分別回顧非等效平行機台與車輛途程問題之相關文獻，2.3 小節為巨集式啟發式演算法的介紹及其應用於相關問題之文獻回顧。

2.1 供應鏈整合之相關文獻

供應鏈管理的成功要素之一，在於有效整合企業間相關業務活動、使得採購、生產、排程、儲存與配送活動緊密連結，發揮整體供應鏈績效(Chiu, 1995)。在 1990 年代時期，學者們研究供應鏈整合的問題，大都是以供應鏈之架構、性質與案例來進行探討，Thomas and Griffin (1996)回顧供應鏈管理整合模式的文獻，而此類模式分成三大範疇：買方與賣方協調合作(buyer-vendor coordination)、生產與配銷整合(production-distribution coordination)以及存貨與配銷整合(inventory-distribution coordination)，並且在其結論中建議供應鏈整合之焦點，應該由策略面(strategic)轉至作業面(operational)，將模糊的問題變成數學模型來解決，以對供應鏈整體利益產生實質的影響。

以往針對產品製造與成品配送兩階段問題的解決方式，學者們以精確法(exact algorithm)或數理證明的方式居多，Lee and Chen (2001)探討的供應鏈整合問題，主要考慮了兩種運送模式，第一類模式考慮屬於機台與機台間的半成品運輸問題；第二類問題則探討為將完成的產品從製造工廠中運送至顧客手上的問題。他們所研究的第二類問題中，產品製造之排程部分為單一機台或雙機式流程式(flow shop)排程問題，成品則以批次的方式，被一台或多台具容量限制的車輛運送。作者釐清各類問題的複雜度後，針對特定問題，依據最佳化特性(optimality property)發展出可於多項式時間內求解的演算法。Hall and Potts (2003)首度提出供應鏈排程(supply chain

scheduling)的觀念，討論供應鏈管理中最小化生產及批次運送成本的問題，並分別以供應商、製造商以及供應鏈整合三種決策者觀點探討此類問題。在產品製造之排程部分，作者將單一階段之供應商或製造商視為單一或平行機台；在批次運送部分，在不限車輛數目，不考慮車容限制的條件下，以直接運送的方式送達顧客手中，因此該研究不需要討論車輛途程的選擇。作者除了針對各類問題找出其問題複雜度之外，亦針對在多項式時間內可求解的問題，提出以動態規劃為基礎的演算法；作者也提出了三個簡單的例子說明以整合的觀點來做決策，可以使得整合排程與配送之成本較分別以單一階段解決最佳化問題之成本總和為低，其成本的改善率在某些情況可達 20%，甚至是單一階段解決最佳化問題成本的一半。之後 Hall and Potts (2005)將問題延伸至考慮車容量限制，同樣針對各類問題找出其問題複雜度之外，亦針對在多項式時間內可求解的問題，提出以動態規劃為基礎的演算法。Chang and Lee(2003)探討 Lee and Chen (2001)所研究的兩類問題，作者將這兩個階段視為一個系統，利用兩種在實務上常被使用的概念來求解此整合問題：向前求解法(forward approach)與向後求解法(backward approach)。其中向前求解法是先求解系統問題中的第一階段的問題，將其結果當成是第二階段的輸入參數而再求解第二階段問題；而向後求解法(backward approach)則是先求解系統的第二階段問題，再依據求解第二階段所得的資訊來求解第一階段的問題。作者使用最劣情況分析(worst-case analysis)來衡量這兩種方法應用於不同問題上的表現。Chang and Lee (2004)探討 Lee and Chen (2001)所研究第二類問題，並考慮車輛容量的限制以及訂單體積的大小，將完工時間定義為訂單送至顧客手中的時間，而以最小化最大完工時間為目標。作者針對機台數量及顧客區域數設定了三種情況：單一機台運送至單一顧客區域、兩台等效機台運送至單一顧客區域及單一機台運送至兩個顧客區域，而以啟發式演算法並配合最劣

情況分析求解該類問題。Li *et al.* (2003) 針對 Chang and Lee (2004)之研究作了更深入的探討，將系統欲服務的顧客數從一個延伸到多個，探討單一機台排程整合單一車輛途程決策的問題，他們假設有多個已知的顧客，以最小化訂單交貨時間的總和為目標，並在特定條件下簡化問題後，使用動態規劃求得最佳解，並建議未來研究方向可以朝著將模型延伸至多部車輛，或是利用啟發式解法來求解任意顧客數目的問題。

Chen and Vairaktarakis (2005)同樣探討生產與運送的整合排程模型：一組訂單會在特定的設備上加工，接著不透過存貨直接送到顧客手中，並假設車輛數量無限制，目的為最小化加權後訂單送至時間與運送成本的總和。此研究在排程的部分考慮單一與等效平行機台兩種情況，並考慮以批次的方式運送成品，將成品送至不同顧客手上，同一批的貨品可能為不同客戶的訂單。陳昱皓 (2006)探討整合產品製造與成品配送兩階段問題，其製造階段屬於非等效平行機台，配送階段則為不具容量限制的車輛途程問題並假設系統中可供運送產品的車輛數目固定，其目標為最小化加權訂單送至時間之總和。作者利用禁忌搜尋法求解此兩階段問題。周碩鴻 (2006)利用基因演算法來求解與陳昱皓 (2006)相同的問題，與陳昱皓所設計之演算所求得之起始解作比較，比較兩者之研究結果，在同樣的輸入資料與參數設定下，所得最佳近似解值相近，與禁忌搜尋法起始解比較後的改善率平均約在 20%左右。

本研究探討 Lee and Chen (2001)研究分類中的第二類問題，探討機台排程包含成品從製造工廠送到顧客手上的情況。在現有的相關文獻中產品製造排程部分，學者們研究的類型大致上有單一機台排程、雙機式流程式排程、等效平行機台與非等效平行機台四類；而在成品配送部分，則是有訂單完成即運送、批次運送至同一顧客、或考慮車容量限制的問題。現今的文獻中，大多以動態規劃法為基礎發展演算法應用在可於多項式時間內

求解的問題，然而不同的限制條件會造成問題的難度不同，當問題規模增大到一定程度時，便無法在合理的時間內找到最佳解。而目前大部分之研究在探討產品製造與成品配送兩階段問題時，而除了陳昱皓 (2006)與周碩鴻(2006)外，幾乎沒有其它文獻著墨於本研究所探討的整合非等效平行機台與車輛途程之兩階段問題，而本研究與陳昱皓 (2006)與周碩鴻(2006)最大的不同差異，在於考慮了車容限制後，利用不同的演算法交換法則，希望在更接近實務的情況下求解問題，故接下來的小節我們將非等效平行機台以及車輛途程問題分開作基本介紹以及一些相關文獻之回顧。

2.2 非等效平行機台與車輛途程問題之相關文獻

本節分別回顧非等效平行機台與車輛途程問題的相關文獻。

2.2.1 非等效平行機台問題之相關文獻

產品製造型態為決定排程的依據，常見的產品製造型態有單機生產(single machine)、流程式生產(flow shop)、零工式生產(job shop)、開放式生產(open shop)、平行機台生產(parallel machine)。所謂平行機台是指有若干個等待加工的工件，需經過某一加工程序，經由具有相似的功能的機台中任一機台進行處理，此類機台即是平行機台。而平行機台又分成完全相同平行機台(identical parallel machine)、等效平行機台(uniform parallel machine)與非等效平行機台(unrelated parallel machine)三類，其中完全相同平行機台表示所有機台之功能與處理速度皆完全相同，相同的工件在不同機台上的加工時間是相同的；等效平行機台是指機台加工速度不同，但相同工件在不同機台上的加工時間呈比例關係；非等效平行機台即表示相同工件在任一台平行機台加工時間皆不一樣，且任兩機台間加工時間無相關性(Allahverdi and Mittenthal, 1994)。以往針對平行機台所做的排程研究，大部分都針對最大完工時間(makespan)、交期延遲(tardiness)以及最小總完工時間(total completion time)三個目標討論，而在非等效平行機台的相關文獻

中，最主要是在討論最小化最大完工時間。Horn (1973)和 Bruno *et al.*(1974)以最小化最大完工時間為目標，將小規模問題轉化成為指派問題，求解機台的排程最佳化。Herrmann *et al.* (1997)利用啟發式演算法求解較大規模之最小化最大完工時間問題，Mokotoff and Jimeno (2002)以混合式整數規畫法(mixed integer linear programming ,MIP)為架構發展了一套啟發式演算法，自國際題庫中取得數據，以最小化最大完工時間為目標，求解非等效平行機台問題，並分析其測試結果與傳統的解法的比較，以建議該演算法在適用的問題範圍。Liaw *et al.*(2003)利用上下界逼近法求解非等效平行機台之總加權延遲時間最小化問題，並與傳統分枝界限法比較求解速度與近似解品質，Ghirardi and Potts (2005)利用改良式分枝界限法解決最小化最大完工時間的問題，得到品質佳的近似解。針對最小化總加權完工時間問題，過去在非等效平行機台上的研究並不多見，Azizoglu and Kirca (1999)則針對非等效平行機台之問題，以最小總加權完工時間為目標，利用分支界限法進行小規模問題之排程最佳化。Weng *et al.*(2001) 探討非等效平行機台問題，在研究中使用了兩種不同的排程策略，第一種策略為將工件分配至擁有最小加工時間或最小成本的機台，第二種策略則是優先指派擁有最小加工時間加上整備時間除以權重的訂單，以最小化加權後平均完工時間為目標，並考慮了整備時間後利用 7 個不同的啟發式演算法求解問題。接下來本研究回顧車輛途程問題之相關文獻。

2.2.2 車輛途程問題之相關文獻

成品配送是供應鏈系統中不可或缺的一環，而其所對應的數學模型大都使用車輛途程問題(vehicle routing problems, VRP)，故在數十年來備受學者及企業的重視。Dantzig and Ramser 在 1959 年首度提出車輛途程問題，基本定義為單一物流中心、由一組車隊將貨品運送至已知顧客的所在位置，每部車輛所裝載之貨品不能超過該車輛之容量上限，每個顧客都會被

服務一次且需求量皆為已知，車輛在服務完被分配之顧客後返回物流中心，而車輛途程問題的目標為最佳化車輛所行走的路徑。近數十年來，學者們陸續研究了不同種類的車輛途程問題，而目前主要是將車輛途程問題的模型歸類成車輛途程問題與時窗性車輛途程問題(VRP problem with time windows)兩大類。所謂時窗性車輛途程問題，即在車輛途程問題加入時窗限制，其問題定義為單一物流中心、由一組車隊將貨品運送至已知顧客的所在位置，每部車輛所裝載之貨品不能超過該車輛之容量上限，每個顧客都會被服務一次且需求量皆為已知，在不違反車容量及時窗限制下，以物流中心為起點與終點，且需在路徑最大時限內完成服務。本研究所探討為一般性的車輛途程問題，並不考慮時窗限制，已有許多學者針對此類問題做了一系列的整理回顧(如 Bodin *et al.* 1983, Laporte 1992)。早期在求解傳統車輛途程問題時，以較簡單的精確法針對規模較小的問題來求解，而 Christofides *et al.*(1981)與 Fisher and Jaikumar (1978, 1981)等提出利用數學規劃方法(mathematical programming approach)求解較大規模之問題，大約可處理 50 個顧客的問題。然而車輛途程問題屬於 NP-Hard 的問題(Garey and Johnson,1979)，當問題規模較小時，可用最佳解法求得最佳解或使用啟發式演算法得到品質佳的近似解；但隨著問題規模變大時，這些方法的求解時間將隨著顧客位置的增加呈指數性成長，因此要在合理的時間內求得最佳解會變得非常困難。應用於車輛途程問題的啟發式演算法包括了節省法(savings algorithm)、掃描法(sweep algorithm)、最鄰近解法(nearest neighborhood algorithm)等。

以下為各類作法的基本介紹(夏萬春, 2000)：

1. 節省法

此方法需先產生一個起始解後，再將各顧客點插入任一路徑內，計算該顧客位置插入後的路徑節省值，選取其中節省值最大者，進而產生一個

新的可行路徑解。圖 1 與圖 2 以兩個顧客與一個物流中心為例，說明節省法前後之比較，其中節點 0 表示物流中心，而節點 1 與節點 2 分別表示顧客 1 與顧客 2。在使用節省法前，顧客 1 與顧客 2 各自用一部車輛從節點 0 出發，運送完畢之後回到原點；使用節省法後，運送模式變成一台車輛一次運送兩位顧客之訂單，運送順序為車輛自節點 0 出發後，先前往顧客 1 之位置交貨，緊接著直接由顧客 1 的位置直接往顧客 2 的位置移動，若 d_{ij} 為從節點 i 移動至節點 j 之路徑長度，則 $S_{12} = d_{10} + d_{20} - d_{12}$ ， S_{ij} 為使用節省法將路徑改變後與未改變前之節省量，顧客節點愈多，可選擇節省之組合愈多，同時節省值數量愈多。

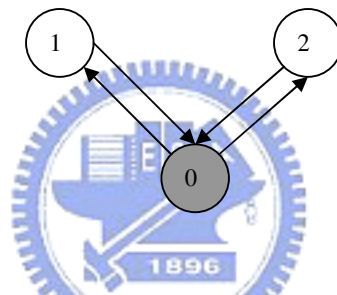


圖 1 節省法使用前路徑

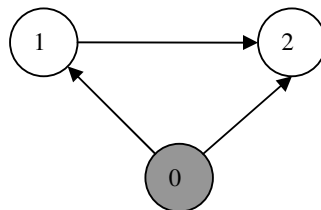


圖 2 節省法使用後路徑

2. 掃描法

此法為 Gillett 和 Miller 於 1974 年所提出的方法，利用極座標來表示各顧客點之位置，以物流中心為原點，選取任一顧客位置為起始點，在基本限制條件下(如車輛容量限制或行走距離限制)，從起始點依順時針(或

逆時針)方向，以掃描的方式尋找下一個顧客位置，所遇到的下個顧客點即為第二個運送的順序。此方法之優點是可任取起始點作求解，再比較所得之起始解何者較佳。圖 3 為順時針掃描法之示意圖。

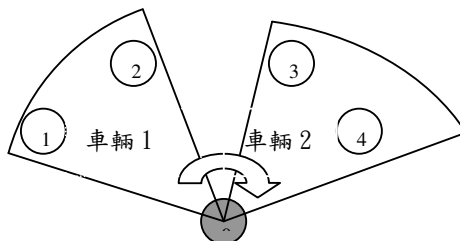


圖 3 順時針掃描法之示意圖

3. 最鄰近解法

最鄰近解法主要是先選取距離物流中心最近的一點為第一個運送順位，在同一條路徑只能走一次的條件下，第二順位則是距離第一個運送之顧客位置最近的顧客點，以此類推，在不違反基本限制條件下，所能服務最多的顧客數，之後再由物流中心開始第二部運送車輛之途程選擇，至到所有顧客皆被服務為止。如圖 4 為一使用最鄰近解法求解的示意圖。

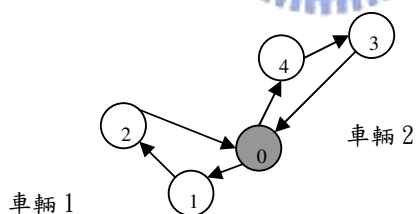


圖 4 最鄰近解法求解之示意圖

使用啟發式演算法可節省許多繁複的搜尋程序，並在有限的時間內找出近似解，對於求解大規模問題而言更顯得其效率性。儘管如此，啟發式演算法常有陷入區域最佳解或近似解品質不佳的問題，因此學者們發展了

巨集式啟發式演算法解決此類問題，加上現今電腦運算速度不斷增快的影響，近幾年來巨集式啟發式(meta-heuristic)演算法的發展變得十分迅速，學者們針對不同問題發明新的演算法並持續改良，由於本研究所討論之兩階段問題複雜度高，應用巨集式啟發式演算法能夠求解較大規模的問題，較能符合實務上之需求。因此下一個小節則回顧與巨集式啟發法演算法相關之文獻。

2.3 巨集式啟發式演算法

本節先介紹巨集式啟發式演算法的發展經歷與基本特性，接著詳細說本研究所採用之禁忌搜尋法，第三小節則是回顧應用禁忌搜尋法於製造或配送單一階段之相關文獻。

2.3.1 巨集式啟發式演算法

根據 2.2.1 以及 2.2.2 小節之文獻回顧可知，早期在研究非等效平行機台或車輛途程問題，通常都是利用數學規劃法求解規模較小的問題或是利用傳統式啟發式演算法針對研究問題有效率地求得一個品質尚不錯的近似解。但本研究所考慮的供應鏈整合問題複雜度為 NP-hard；當問題規模增大時，求解時間會與問題規模呈指數分配的成長。近年來電腦運算速度不斷提升，巨集式啟發式演算法在解決非等效平行機台或車輛途程這種問題上有長足的進步。目前常被研究者所使用的巨集啟發式演算法有基因演算法(genetic algorithms)、模擬退火法(simulated annealing)、禁忌搜尋法(tabu search) 以及螞蟻演算法(ants colony optimization)等，以下為禁忌搜尋法以外各演算法的基本介紹，禁忌搜尋法將於下一節做更詳細之說明：

(1) 基因演算法：

此法最早是在 1970 年代由 Holland 提出(Booker 1989, Holland 1992)，主要利用生物學遺傳優選的原理，以達爾文的物競天擇論為基礎，強調物競天擇、適者生存，並利用個體間的配對與刺激產生突變，來產生更好的

下一代，經由如此不斷地演化，適應力較差之子代會被淘汰；而適應力佳的子代則用以產生更佳的下一代。遵循達爾文進化論的精神，經由重複地演化，以產生近似最佳解甚至為最佳解，基因演算法即是將此種生命科學現象應用於求解組合最佳化的問題。

(2) 模擬退火法：

是由 Metropolis *et al.* (1953) 提出一個模擬金屬降溫的演算法，應用物理冶金方面的觀念，將金屬快速升溫後，再慢慢降低溫度，進而達到金屬結構之最佳狀態，以得到最佳的冶金效果。而三十年後 Kirkpatrick *et al.* (1983) 發現要將溫度達到物質能量最低狀態，猶如在問題中尋求目標函數的最小值，此降溫過程與尋優步驟觀念是可以相通的，因此他們提出類似於 Metropolis *et al.* (1953) 的模擬方式，應用於求解一些困難的組合最佳化問題。

(3) 蟻群演算法

蟻群演算法初始概念最早是由 Colormi *et al.* (1991) 提出，後來由 Dorigo and Gambardella (1997) 完成其演算法之架構。主要是起源於發明者在觀察螞蟻之移動行為時發現，螞蟻在移動時會分泌一種費洛蒙(pheromone)，螞蟻行經一條路徑之機會與該路徑曾遺留之費洛蒙成正比。一個路程遺留之費洛蒙愈多表示之前經過這條路徑的螞蟻愈多，費洛蒙濃度愈高又會吸引更多螞蟻行走該路徑。因此，當螞蟻面臨兩條路以上的抉擇時，其行走某一路徑之機率與其遺留的費洛蒙濃度有關。由於愈短之路徑其螞蟻通過時間短，以致愈短的路徑上所遺留之費洛蒙濃度愈高，進而誘使更多螞蟻行經該路徑，使得所有螞蟻行走最短路徑的機會最大。螞蟻演算法即在模擬螞蟻移動之行為，進行最佳化組合之搜尋工作。螞蟻演算法可應用在最短路徑、車輛途程問題、生產排程等最佳化問題上，目標為求得各問題之最適解 (蔡志強，2003)。

巨集式啟發式演算法最大的優點在於解決問題時，能夠跳脫出區域最佳解的限制，快速地找到品質佳的近似解。因此希望在較短處理時間、耗用較少電腦資源的條件下獲得品質佳的近似解時，使用巨集式啟發式演算法是很好的選擇。各種演算法都有其優缺點，並非單一方法適用於所有的問題，必須視問題類型來決定使用何種演算法，因此演算法的設計與搜尋策略是近似解品質與求解效率好壞的關鍵。而使用禁忌搜尋法求解本研究中之單一階段的非等效平行機台或車輛途程問題時，已有相關文獻之研究可以得到不錯的效果（如 Osman 1993, Reeves 1993, Class *et al.*1994, Piersam and Dijk 1996, Nowicki and Smutnicki 1998）。因此本研究推論應用禁忌搜尋法於求解整合產品製造與成品配送之問題，可以求得品質佳的近似解。因此本研究接下來將介紹禁忌搜法的發展過程與基本元素，並回顧一些使用禁忌搜尋求解非等效平行機台以及車輛途程問題之相關文獻。

2.3.2 禁忌搜尋法

禁忌搜尋法屬於巨集式啟發式演算法的一種。此演算法廣泛地被應用在求解組合最佳化之問題，在工業工程領域常用於求解關於排程、車輛途程、設施規劃等問題上。禁忌搜尋法最早是由 Fred Glover 於 1977 年提出，用於求解整數規劃問題；一直到了 1990 年代 Glover 才完整說明禁忌搜尋法之基本原理與使用(Glover 1989, Glover 1990)。禁忌搜尋法需要生成一個起始解，然後根據起始解產生一個鄰域，由其鄰域之中找一個最佳鄰近解作為下次迭代搜尋的起始解並產生新的鄰域，如此不停重覆下去，直到演算法達到所設定停止搜尋準則。禁忌搜尋法具備彈性記憶結構之特性，可記錄曾經搜尋過之最佳鄰近解，避免下次迭代重複搜尋的情況發生，增加跳出區域最佳解的機會。被記錄禁止再搜尋之解集合稱之為禁忌表列(tabu list)。設計者需要決定禁忌表列的長度，並且使用先進先出(first in first out, FIFO)的原則處理表列內之各個解，意即迭代次數超過表列長度時，第一

個被記錄禁止再搜尋的解會被移出禁忌表列中，以此類推。因此進入禁忌表列的解經過一定搜尋次數後會被解禁，演算法可再往該區域搜尋。以下簡單介紹禁忌搜尋法之基本組成元素(Ho and Haugland 2004, Homberger and Gehring 2005)

(1) 起始解(start solution)

較為常用的方法是利用啟發式演算法，快速地求得一個可行解作為禁忌搜尋法的起始解。

(2) 鄰近解(neighborhood solution)

指鄰近起始解的可行解，不過鄰近之定義是由設計者依問題性質或需求而決定。考慮一個有 i, j, k 三位不同顧客的產品由一台車輛運送的例子，假設起始解運送順序為 $i \rightarrow j \rightarrow k$ ，若設計者將鄰近解定義為運送順序互換，則鄰近解有 $j \rightarrow i \rightarrow k$ 、 $i \rightarrow k \rightarrow j$ 或是 $k \rightarrow j \rightarrow i$ 兩種，禁忌搜尋將會選擇較佳之鄰近解作為下次搜尋的起點。

(3) 移步(move)

從起始解移動至鄰域中之解的動作稱為移步，上述例子中的一次運送順序互換即為一次移步。

(3) 禁忌表列

由於傳統搜尋法在移步過程中，會有陷入循環解的情況，為了避免重覆搜尋，禁忌搜尋法設立了禁忌表列，記錄前幾次移步，經過一定次數的迭代之後，最早進入禁忌表列中的移動路徑會被解禁。故當有一新鄰近解進入禁忌表列的同時，一個本來在禁忌表列中的解會離開表列。

(4) 期望水準(aspiration criteria)

由於禁忌表列的限制，在搜尋的過程中無法移至禁忌表列所記錄之移步，但由於這些移步中，可能會有不錯的可行解出現，故需要設定期望水準來為這類好的可行解去除禁忌限制，讓搜尋可進行該移步。一般而言，

期望水準之設定方式為移步之可行解優於現在的最佳解，在這設定下若在搜尋中求得一個鄰近解在禁忌表列中，但移步所產生之目標值優於搜尋至今的最佳解(即滿足期望水準)，則將這移步解開禁忌禁制，並更新成為目前之最佳解。

(5) 深度化搜尋(intensification)

禁忌搜尋法的彈性記憶結構主要分成短期記憶與長期記憶兩類，深度化搜尋即為短期記憶之部分，主要是針對特定區域做深入的搜尋。

(6) 廣度化搜尋(diversification)

深度化搜尋則為彈性記憶結構中的長期記憶部份。長期記憶記錄進行搜尋的所有移步，當搜尋法陷入區域最佳解時，經由改變鄰近解定義或重新生成一個起始解，對先前較少經過之區域再做深度化的搜尋。

(7) 停止準則(stopping criterion)

指禁忌搜尋法可以停止搜尋所需要達到的條件，一般而言，停止準則有下列三種情況(陳正雄，2000)：

- (i) 搜尋迭代次數達到設計者一開始所設定的最大次數或是目標值達到預設的值。
- (ii) 超過某一設定搜尋代次數所求得之最佳目標值仍是無法優於現有的最佳目標值。
- (iii) 運算之鄰近解集合扣掉違反禁忌搜尋規則之解集合為空集合。

以下本研究回顧禁忌搜尋法應用於非等效平行機台問題，車輛途程問題以及整合前述兩階段問題之相關文獻。

2.3.3 應用禁忌搜尋法之相關文獻

在應用於非等效平行機台問題部份，Suresh and Chaudhuri (1996)利用禁忌搜尋法求解不含前置時間之多目標非等效平行機台問題，該問題主要

有兩個目標：最大完工時間與最大延遲時間。作者將演算法測試之結果與他們先前發展的 GAP-EDD 演算法(Suresh and Chaudhuri, 1994) 所產生的結果比較而發現在研究問題所定義範圍下，禁忌搜尋法在近似解的品質與求解之速度均表現 GAP-EDD 演算法來得好。Hsieh *et al.* (2003)則是研究主機板產業中印刷電路板加工廠之鑽孔作業問題，由於製程中同一功能之不同工作站的機台效率不一且互相獨立，故可視為非等效平行機台問題。該研究以機台成本、使用之機台數以及機台利用率作為目標函數，利用禁忌搜尋法在合理的時間內求得一品質不錯之近似解並與人工排程的結果作比較。研究結果指出雖然人工排程所花時間較少，但禁忌搜尋對近似解的品質較人工排程優異許多。Chen (2006)則是針對考慮在加工限制與整備時間之情境下，將最大延遲時間作為目標函數，研究結果顯示禁忌搜尋法優於模擬退火法之電腦測試結果；而針對小規模之非等效平行機台問題，此演算法亦可在合理的時間內求得最佳解。

禁忌搜尋法在車輛途程問題上亦有不少的應用，大部份的使用方式為改良或結合其它演算法一起使用。Gendreau (1994)針對一般性的車輛途程問題以禁忌搜尋法為主體，提出一個演算法並命名為 TABUROUTE。此演算法的最大特點，在於可將不可行解作為起始解與鄰近解，然後檢視符合期望水準之解是否為不可行解，若在移動後產生可行解則更新為搜尋法之最佳解。作者根據國際標竿題庫所提供之數據 TABUROUTE 演算法，結果發現 TABUROUTE 所得之解優於當時其它啟發式演算法所得之解。

Renaud *et al.* (1996) 則以禁忌搜尋法為基礎設計了 FIND 演算法，以快速改善、廣度化搜尋與深度化搜尋三個階段，對多物流中心之車輛途程問題，在考慮車輛容量及路徑長度限制之條件下，以最短路徑為目標函數求解，該演算法測試 11 組國際標竿問題之數據，比較先前其它學者的測試結果比較，指出 FIND 優於先前標竿之測試結果。

針對供應鏈整合問題，Lee (2001)利用巨集式啟發式演算法在印刷電路板(printed circuit board)製造系統中，求解包含機台成本、存貨成本和時窗懲罰之多機台兩階段問題。其中比較了禁忌搜尋法、模擬退火法、基因演算法與鄰近搜尋法，結果顯示禁忌搜尋法在求解此類多機台兩階段問題時，所得到之解的品質優於模擬退火法、基因演算法與鄰近搜尋法，而求解的時間僅略遜於鄰近搜尋法，而較模擬退火法和基因演算法為短。在使用巨集式啟發式演算法求解非等效平行機台與車輛途程之兩階段整合問題，並鮮少學者作深入之研究與探討。僅陳昱皓 (2006)在不考慮車容限制的條件下，將車輛數假設為已知，利用禁忌搜尋法求解整合非等效平行機台與車輛途程之兩階段問題；周碩鴻 (2006)利用基因演算法來求解與陳昱皓 (2006)相同的問題，與陳昱皓所設計之演算所求得之起始解作比較，比較兩者之研究結果，在同樣的輸入資料與參數設定下，所得最佳近似解值相近，與禁忌搜尋法起始解比較後的改善率平均約在 20%左右。

在一系列的文獻回顧之後，本研究所探討之問題模型與陳昱皓 (2006)與周碩鴻 (2006)所研究之問題類似，但不同之處在於兩者之研究皆不考慮車容限制，並在研究假設時限制車輛數目，為了更符合實務應該情況，本研究加入車容限制且車輛數目為一變數隨著批次組合不同而有所變化。此外使用巨集式啟發式演算法對於求解 NP-hard 問題有很好的表現，而禁忌搜尋法在針對求解單一階段之非等效平行機台或車輛途程問題，相較於其它巨集式啟發式演算法，在大部份的情況下皆可獲得品質較佳的近似解，故本研究推論針對求解非等效平行機台與車輛途程之兩階段整合問題，使用禁忌搜尋法會是一個很好的選擇。

第三章 問題定義與研究方法

本研究探討整合產品製造與成品配送兩階段問題，分別應用非等效平行機台排程問題與具車容限制之車輛途程問題模擬此兩階段。本研究將訂單之完成時間定義為產品從開始製造至送達顧客手上所花費的時間，每個訂單由決策者依據訂單重要性決定其權重，以總加權後訂單完成時間為所探討問題的目標函數，並以最小化此目標函數為目的。針對此類問題，本研究設計了一個禁忌搜尋法（以下稱為 Tabu_PDP）求解，期望可以在合理的時間內找到品質佳的近似解。

3.1 問題定義

本研究之問題屬於組合最佳化問題，可用下列簡單模式表示：

$$\text{Min } Z = f(x)$$

$$\text{s.t. : } x \in X$$

x ：表示訂單機台的加工順序、批次編號以及訂單運送順序的一種組合

X ：表示訂單機台的加工順序、批次編號以及訂單運送順序所有的組合

$f(x)$ ：表示 x 所對應的目標值函數

本研究討論的問題敘述如下：系統內有 n 張訂單，每個訂單的體積均為 1 個單位，需經過 k 個非等效平行機台中的任何一部機台加工，為了方便敘述，本研究顧客編號為 i 之訂單即為訂單 i 。訂單 i 在機台 m 的加工時間為 p_{mi} ，而訂單 i 於機台加工完畢之後的時間定義為在機台上之完成時間 z_i ，訂單在機台上加工完畢後之訂單需經過編號 r 之車輛送到顧客手上，而編號為 r 之車輛即為編號為 r 之批次，為方便敘述稱以下為 B_r 。將 n_r 定義為 B_r 中的訂單個數，系統內車輛數沒有限制，但每輛每次最多只能裝載 v 張訂單，車輛自開始運送到訂單 i 顧客手上所花費的時間為 δ_i ，為車輛從

製造工廠出發，依訂單運送順序運送成品至顧客 i 前各點間所花費時間之總和。 c_i 為訂單 i 從開始加工到送達顧客手上所花費之時間，定義如式(1)，其中 γ_r 為 B_r 之發車時間，為該批次所配送之訂單中於機台完成時間的最大者，如式(2)。假設製造工廠為原點，而訂單 0 為一虛擬訂單，其製造與配送時間均為 0， d_{0i} 為製造工廠至訂單 i 的配送時間，而 d_{ji} 為訂單 j 至訂單 i 的配送時間。若訂單 j 為訂單 i 運送順序之前一站，則 δ_i 由式(3)表示之。

$$c_i = \gamma_r + \delta_i \quad i \in B_r \quad (1)$$

$$\gamma_r = \max\{z_i\} \quad i \in B_r \quad (2)$$

$$\delta_i = \delta_j + d_{ji} \quad i \neq j \quad i, j \in B_r \quad (3)$$

本研究所探討問題之目標函數如式(4)所示、而 w_i 為訂單 i 之權重，本問題的決策變數包括決定訂單在哪部機台上加工及其加工順序、訂單被分配於哪個批次以及訂單在批次內的運送順序。

$$\text{Min } Z = \sum_{i=1}^n w_i c_i \quad (4)$$

為求在合理的範圍中簡化問題，本研究就產品製造階段與成品配送階段所設定之基本假設如下：

一、產品製造階段

1. 每一工件在各機台上的加工時間皆為已知；
2. 所有訂單起始時間皆 0；
3. 機台在時間為 0 時皆可供訂單加工；
4. 機台加工無整備成本；
5. 不考慮訂單重工或機台當機；
6. 任一機台同一時間只能加工一個訂單；
7. 每一顧客只訂購一筆訂單。

二、成品配送階段

1. 單一物流中心（即為製造工廠）；
2. 可使用之車輛數目無上限；
3. 每部車皆從製造工廠出發；
4. 所有車輛在時間為 0 時皆可供使用；
5. 所有顧客位置已知；
6. 每一部車的容量限制與速度皆相同；
7. 車輛行走距離無限制；
8. 每部車只運送一次；

此外本研究所探討的問題，具備以下之最佳化性質(optimality property)：

定理 1：本研究所探討的問題，在下列情況下存在一最佳排程。

- (i) 本研究所探討的問題，在產品製造排程部份中，若在每部機台上加工之訂單間機台無閒置時間(idle time)，則存在一個最佳排程。
- (ii) 本研究所探討的問題，當每一部車輛的發車時間為該批次內中訂單在機台上最大完工時間時，則存在一個最佳排程。
- (iii) 本研究所探討的問題，當同一批次之訂單在所有機台上均為連續完工，即同一批次之訂單在所有機台之完工時間的區間內沒有不同批次的訂單交互完工時，則存在一個最佳排程。

證明：

- (i) 假設存在一排程，其中訂單 a 與訂單 b 為在同一機台上兩相鄰被加工的訂單，其加工順序為 $a-b$ 且訂單 a 與訂單 b 間有機台的閒置時間，則可將訂單 b 的加工起始時間提前至訂單 a 之完工時間點，而不會增加此排程之目標函數值。
- (ii) 假設一個特定批次的發車時間大於該批次的訂單最大完工時間時，則可將此批次的發車時間提前至該批次所有訂單中的最大完工時間，此可行解所對應的目標函數值不會因此而增加。

(iii) 假設存在一個排程，其中有 $a、b、c、d、e、f$ ，共 5 個訂單，而其完工順序依序為 $a-b-c-d-e-f$ ，即 $z_a \leq z_b \leq z_c \leq z_d \leq z_e \leq z_f$ 。若目前的分批方式為訂單 $a、b、e$ 屬於 B_1 ，訂單 $c、d、f$ 屬於 B_2 ，其中 B_1 的發車時間 $\gamma_1 = \max\{z_a, z_b, z_e\} = z_e$ 而 B_2 的發車時間 $\gamma_2 = \max\{z_c, z_d, z_f\} = z_f$ 。若我們將 B_1 內的訂單 c 與 B_2 的訂單 e 交換，使得 B_1 包含訂單 $a、b、c$ ，而由(ii)得知其交換後之發車時間應調整為 $\gamma_1' = \max\{z_a, z_b, z_c\} = z_c \leq \gamma_1$ ；同理，交換後之 B_2 包含訂單 $d、e、f$ ，且其發車時間應調整為 $\gamma_2' = \max\{z_d, z_e, z_f\} = z_f = \gamma_2$ 。經過此訂單所屬批次交換之步驟，此排程的目標函數值不會因此而增加。

3.2 Tabu_PDP 演算法設計

本小節介紹 Tabu_PDP 起始解之求法，接著說明鄰近解之定義方式與交換法則，最後則是 Tabu_PDP 演算法的步驟。

3.2.1 Tabu_PDP 之起始解

禁忌搜尋法需要生成一起始解後，再從起始解產生一個鄰域，由鄰域之中找一個最佳鄰近解作為下次迭代搜尋的起始解。在起始解生成的部份，有隨機生成或使用啟發式演算法產生兩種方法。為了提高演算法之搜尋效率，本研究設計一個啟發式演算法（以下稱為 T_Start）產生一可行解，以此解做為 Tabu_PDP 的起始解。

T_Start 首先考慮配送階段在優先滿足車輛利用率的假設下，依各訂單權重與顧客的位置決定各車輛的運送批次與順序；再回到產品製造的階段，決定各批次在機台加工的順序，最後依批次順序將訂單排入機台，進而得到一組可行解。以下詳細說明 T_Start 的步驟：

(i) 決定批次與車輛途程之運送順序

本研究先以最鄰近解法(nearest neighborhood)為基礎，發展了一個啟發式演算法（以下稱為演算法 A）來決定運送批次以及批次內的運送順序。

最鄰近解法是車輛途程問題中的一種啟發式演算法，在選擇運送順序時僅考慮顧客間之相對距離。然而在本研究所探討之問題中，需考慮權重對目標函數之影響，故在選擇顧客與運送順序時，以運送時間除以該訂單之權重為衡量標準。演算法 A 的各個步驟說明如下：

演算法 A

步驟一. 令 N 為所有訂單之集合即 $N = \{1, 2, \dots, n\}$ ；

$$B_r = \phi, \text{ for } r = 1, 2, \dots, n; \quad n_r = 0, \text{ for } r = 1, 2, \dots, n;$$

$$r = 1; c = 0; \delta_i = 0; t = 0$$

步驟二. 利用權重最鄰近解法選取訂單為運送之對象

$$i = \underset{j}{\text{aug min}} \left\{ \frac{d_{ij}}{w_j}, j \in N \right\};$$

步驟三. 將利用演算法所選取之訂單放入批次 B_r 中並記錄訂單編號與

運送時間

$$N := N \setminus \{i\}$$

$$B_r := B_r \cup \{i\}$$

$$n_r := n_r + 1$$

$$\delta_i := \delta_i + d_{ii}$$

$$t := i$$

步驟四. 檢查是否所有訂單均被分配完畢

1. 若 $N = \phi$ ，令 $a = r$ ，為起始解之批次數，結束演算法 A。
2. 若 $N \neq \phi$ ，繼續步驟五。

步驟五. 檢查運送批次體積是否達到車容限制

1. 若 $n_r = v$ ， $r := r + 1$ ，回到步驟二。
2. 若 $n_r \neq v$ ，回到步驟二。

(ii) 試算各批次於機台清空時之最大完工時間

在使用演算法 A 決定了運送批次數(a)、各批次內訂單之組成以及各



批次中訂單之運送順序後，接著針對各個批次的起始機台負荷皆為 0 的情況下，透過最短加工時間(shortest processing time, SPT)的原則，試算各批次的最大完工時間 P^r 。以下將此部份之演算法稱為演算法 B，其流程如下：

演算法 B

步驟一. 令 M 為所有機台之集合即 $K = \{1, 2, \dots, k\}$

步驟二. 假設所有機台之始起負荷量為 0

$$\text{令 } l_k = 0, m = 1, 2, \dots, k; r = 1; t = 0$$

步驟三. 利用 SPT 原則選取訂單與被加工之機台

$$\text{令 } i = \underset{j}{\text{aug min}} \{l_k + p_{kj}, j \in B_r, m \in K\}$$

$$x = \underset{k}{\text{aug min}} \{l_k + p_{kj}, j \in B_r, m \in K\}$$

步驟四. 將所選取之訂單放入機台並更新機台負荷量

$$B_r := B_r \setminus \{i\};$$

$$l_x := l_x + p_{xi};$$

步驟五. 檢查批次中的訂單是否都已被排入機台；

1. 若 $B_r = \phi$ ， $P^r = l_x$ ，繼續步驟五

2. 若 $B_r \neq \phi$ ，回到步驟二。

步驟六. 檢查所有批次是否已被分配完畢

1. 若 $r = a$ ，終止演算法 B。

2. 若 $r \neq a$ ， $r := r + 1$ ，回到步驟二。

(iii) 決定批次在機台上的加工前後順序

演算法 B 試算了在機台尚未處理任何訂單時（即機器負荷為 0 時），各批次的最大完工時間 P^r 。由 3-1 小節所提到的最佳化特性(iii)可知，若 B_1, B_2 為兩連續批次，其發車時間 B_1 必定小於 B_2 ，此排程才有可能最佳排程。因此批次在機台上的加工前後順序，相當於批次之發車時間之順

序。故本研究另外設計了一決定批次於機台上的加工先後順序的演算法（以下稱為演算法 C），根據式(5)計算出批次之最大完工時間(P^r)加上利用最鄰解法所求得之批次總運送時間除以該批次中所有訂單的權重和，

$$H_r = \frac{P^r + \max\{\delta_j\}}{\sum W_j} \quad j \in B_r, r \in R \quad (5)$$

再將各訂單依批次之先後順序重新編號，其後用 SPT 原則將訂單排入機台加工順序，求得一組利用本研究所設計之演算法所求得之起始解。以下說明演算法 C 之各個步驟。

演算法 C

步驟一. 令 $R = \{1, 2, \dots, r\}$,

步驟二. 利用式(3-3)決定被排入機台的批次

$$\text{令 } i = \underset{r}{\text{aug min}} \left\{ \frac{P^r + \max\{\delta_j, j \in B_r\}}{\sum_{j \in B_r} w_i}, r \in R \right\}$$

$R := R \setminus \{i\}$ ，至所有批次皆被排序完畢。

步驟三. 將批次依照機台上的加工順序重新編號，使得

$$H_1 = \frac{P^1 + \max\{\delta_j, j \in B_1\}}{\sum_{j \in B_1} w_i} \leq H_2 = \frac{P^2 + \max\{\delta_j, j \in B_2\}}{\sum_{j \in B_2} w_i} \leq \dots \leq H_a = \frac{P^a + \max\{\delta_j, j \in B_a\}}{\sum_{j \in B_{ar}} w_i}$$

;

令 $r = 1; l_k = 0; k = 1, 2, \dots, m$;

步驟四. 再次使用 SPT 原則選取訂單與被加工之機台

$$\text{令 } i = \underset{j}{\text{aug min}} \{l_k + p_{kj}, j \in B_r, m \in K\}$$

$$x = \underset{k}{\text{aug min}} \{l_k + p_{kj}, j \in B_r, m \in K\}$$

步驟五. 記錄目前批次在機台加工之屬性

$$B_r := B_r \setminus \{i\}$$

$$l_x := l_x + p_{xi};$$

步驟六. 檢查批次內訂單是否分配完畢

1. 若 $B_r = \phi$, $\gamma_r = \max\{l_k, k \in M\}$, 繼續步驟七
2. 若 $B_r \neq \phi$, 回到步驟四。

步驟七. 檢查所有批次數是否分配完畢

1. 若 $r = a$, 結束演算法 C。
2. 若 $r \neq a$, $r := r+1$, 回到步驟四。

(iv) 計算目標函數值

經由(i)、(ii)以及(iii)之演算法 A、B 以及 C，求得一組可行解，作為 Tabu_PDP 之起始解，計算並記錄其目標函數。T_Start 演算法的流程整理於圖 5。

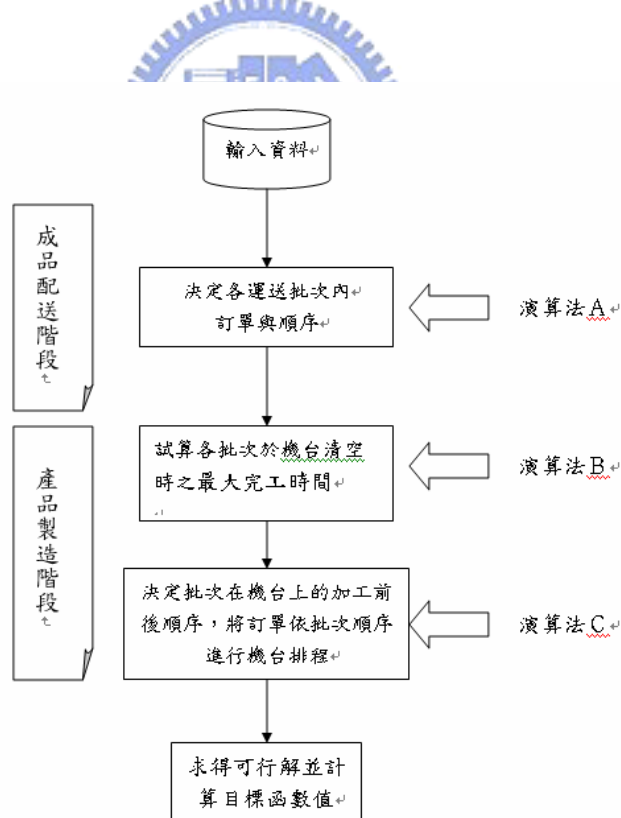


圖 5 T_Start 演算法之流程

3.2.2 鄰域定義與交換法則

本研究的三個決策點為決定訂單於機台選擇與加工順序、訂單被分配之批次以及訂單在批次中的訂單運送順序，故鄰近解的定義基於決策點來設定。本研究設計了以下三種交換法則：

(1) 批次內運送順序交換

第一種類型之鄰近解，為批次內運送順序之兩兩交換。若 B_r 內有 n_r 組訂單，則該批次這類型之鄰近解有 $C_2^{n_r}$ 種組合，若有 y 個批次則總

共有 $\sum_{r=1}^y C_2^{n_r}$ 種組合

(2) 批次於機台加工順序之交換

第二類型之鄰近解為批次於機台加工順序之交換，批次的個數與批次數相同，當 $y \geq 2$ 時，批次於機台加工順序交換之組合共有 C_2^y 種。

(3) 不同批次中訂單之交換

第三類型之鄰近解為不同批次中訂單之交換，選擇任兩批次中的訂單利用兩兩交換法產生鄰近解，本研究假設當不同批次中的訂單交換時，其在指派之機台與加工順序以及車輛運送順序亦同時被交換，否則將違反最佳化性質(i)或者是(ii)，此類鄰近解的組合共有 $C_2^y C_1^{n_1} C_1^{n_2}$ 種。

本研究之鄰近解定義，為三種類別解之集合，即目前解之鄰域，而鄰域中目標函數值最低且不在禁忌表列中的解，即為最佳移步。

3.2.3 Tabu_PDP 之結構與流程

本研究 Tabu_PDP 基本搜尋流程如圖 6，以下說明各步驟之詳細內容。

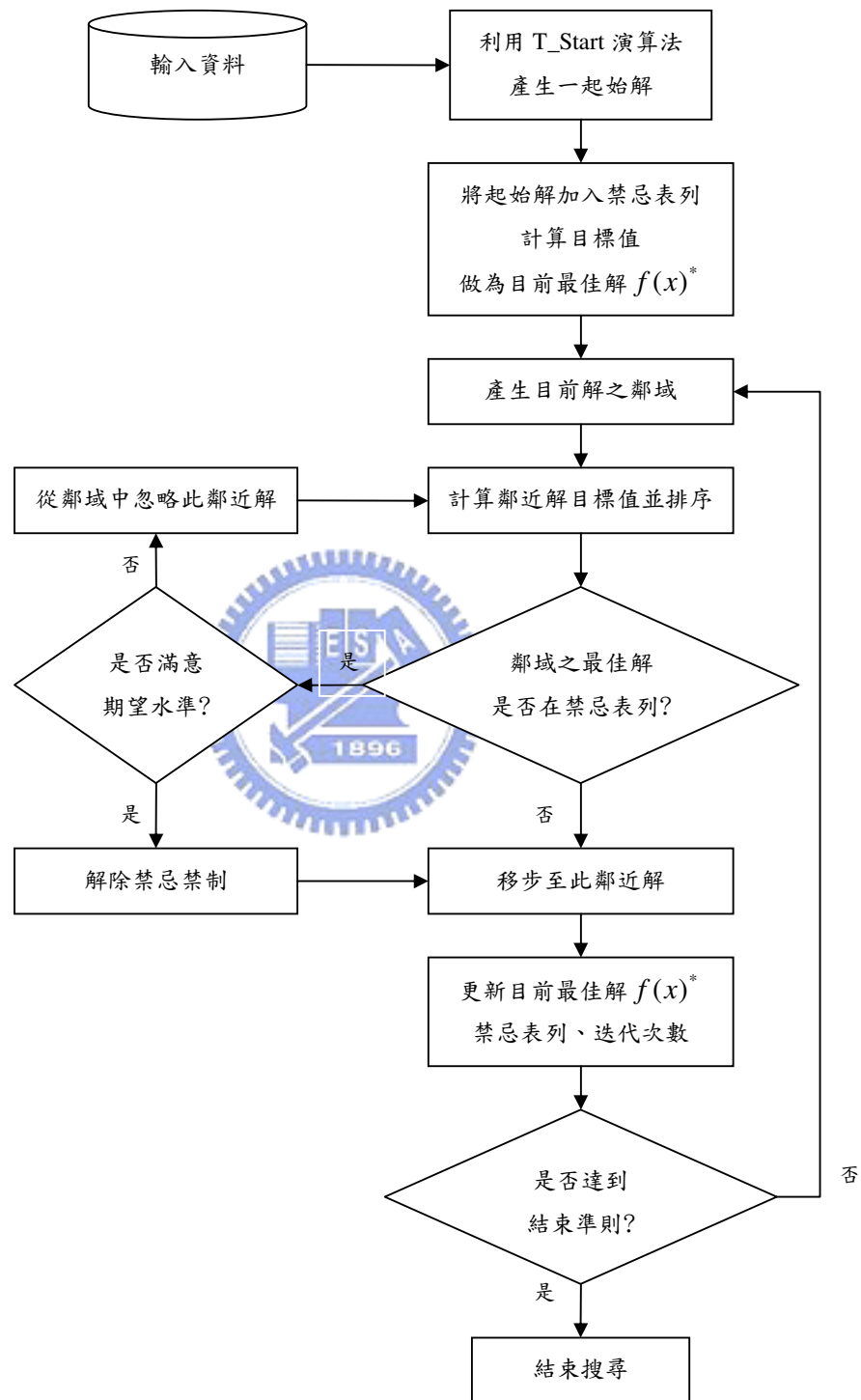


圖 6 Tabu_PDP 演算法流程

步驟一. 設定禁忌表列、期望水準與迭代次數

1. 禁忌表列: 設定為 t 即是須經過 t 次的搜尋才能重複搜尋過去所搜尋過的解。
2. 期望水準: 本研究設定搜尋到新的解若要滿足期望水準必須同時符合下列三種情況:
 - (i) 禁忌限制。
 - (ii) 對應目標值優於目前所得之最佳解 $f(x)^*$ 。
 - (iii) 禁忌限制內最佳解。
3. 結束準則: 預設當迭代次數達到指定之次數或最佳近似解連續指定次數未更新時即停止。

步驟二. 利用 T_Start 產生 Tabu_PDP 的起始解

步驟三. 更新禁忌表列、計算目標函數值更新為 $f(x)^*$

在求得 Tabu_PDP 之起始解後，將此解加入禁忌表列中，以避免在第二次迭代時的鄰域中，將本研究所求得之起始解作為第二次迭代中鄰域的最佳解，而發生重覆搜尋的情況。在第一次迭代中，目前演算法之 $f(x)^*$ 為起始解之目標函數值。

步驟四. 移步的選擇

本研究檢視在這次迭代所生成的鄰域，計算所有鄰近解所對應之目標函數值，並找出所有鄰近解中目標函數值最小的一個。若此解不在禁忌表列中，則移步到此一新的解，此解為本次迭代的最佳移步，同時更新禁忌表列並檢視最佳移步所對應之目標函數值是否優於目前最佳解 $f(x)^*$ ，當此解優於目前最佳解，更新此解成為最佳解 $f(x)^*$ 。若此鄰近解位於禁忌表列中，則視其是否符合期望水準，本研究設定之期望水準為在禁忌限制中其對應是目標函數值優於目前最佳解 $f(x)^*$ ，若符合期望水準，則移步至此鄰近解，更新禁忌表列與目前最佳解 $f(x)^*$ 。當所有鄰近解中目標函數值最

小的一個解位於禁忌表列，而其目標函數值又小於目前最佳解 $f(x)^*$ 時，則在鄰域中排除此解後，重新搜尋鄰近解中目標函數值最小的可行解，並重覆上述動作，直到找到最佳移步並更新禁忌表列、最佳解以及迭代次數。

步驟五. 檢視結束準則

本研究所設定之結束準則為迭代次數達設定之次數即停止。

3.3 範例說明

本小節以本研究所設計之 T 架構為例，說明 T_Start 的計算步驟與 Tabu_PDP 第一次搜尋之結果。考慮 6 個訂單(即 $n = 6$)、2 部機台(即 $k = 2$)、車輛容量上限為 3 個訂單(即 $v = 3$)的問題。為方便表示，假設訂單 i 由顧客 i 所訂購。各訂單權重值以及訂單於各機台加工時間如表 1 所示，各訂單與製造工廠間的運送時間矩陣如表 2 所示。

表 1 訂單權重值以及訂單於各機台加工時間

	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
w_i	3.1	2.5	2.9	4.7	1.8	6.2
p_{i1}	14	42	35	16	27	50
p_{i2}	20	19	40	48	20	30

表 2 各訂單與製造工廠間的運送時間矩陣

d_{ij}	製造工廠	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
製造工廠	0	137	174	89	78	54	96
訂單 1	137	0	115	110	161	134	183
訂單 2	174	115	0	69	91	122	118
訂單 3	89	110	69	0	36	76	99
訂單 4	78	161	91	36	0	102	51
訂單 5	54	134	122	76	102	0	63
訂單 6	96	183	118	99	51	63	0

步驟一. 設定禁忌表列、期望水準與迭代次數

由於此範例為說明生成起始解與第一次搜尋之結果，故不設定禁忌表列長度以及結束準則。

步驟二. 利用 T_Start 產生一起始解

(i) 決定批次與車輛途程之運送順序

首先本研究將各訂單與製造工廠間的運送時間矩陣轉換成權重運送時間矩陣如表 3，利用在 3.2.1 節介紹的演算法 A 來決定訂單的運送批次以及其運送所花費的時間，可得從製造工廠至各訂單之最小值為訂單 6 的 15.5，因此訂單 6 為 B_1 中第一個被運送之訂單。當車輛到達將貨品送達訂單 6 之位置後，需選擇接下來被運送之訂單；訂單 6 可以前往的顧客位置中，其權重運送時間值最小的為訂單 4 的 10.9，故訂單 4 為批次中第 2 個運送點。再來檢視訂單 4 可達位置中最小值為訂單 3 的 12.4，為第 3 個運送點。此時 B_1 訂單體積已達車輛容量上限 3， B_1 可運送訂單與訂單運送順序為製造工廠-訂單 6-訂單 4-訂單 3。接著開始新的批次 B_2 自製造工廠出發，求得其運送訂單與順序為製造工廠-訂單 5-訂單 1-訂單 2。求得 B_1 與 B_2 之運送順序後，計算其從開始發車到成品送達顧客手上所花費之時間如表 4 所示。

表 3 權重運送時間矩陣

d_{ij}	製造工廠	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
製造工廠	0	44.2	69.6	30.7	16.6	30.0	15.5
訂單 1	0	0.0	46.0	37.9	34.3	74.4	29.5
訂單 2	0	37.1	0.0	23.8	19.4	67.8	19.0
訂單 3	0	35.5	27.6	0.0	7.7	42.2	16.0
訂單 4	0	51.9	36.4	12.4	0.0	56.7	8.2
訂單 5	0	43.2	48.8	26.2	21.7	0.0	10.2
訂單 6	0	59.0	47.2	34.1	10.9	35.0	0.0

表 4 批次運送順序以及其對應之訂單到達時間

B_1	訂單 6	訂單 4	訂單 3
到達時間	96	147	183
B_2	訂單 5	訂單 1	訂單 2
到達時間	54	188	303

(ii) 計算各批次分別在機台負荷量為 0 時之最大完工時間

在第一階段中，本研究求得起始解中批次的分配方式以及運送順序，接著以批次方式決定其在機台上加工的先後順序，已知 B_1 的訂單在各機台加工時間如表 5。針對各批次利用演算法 B 求解，先求得各批次分別在所有機台負荷為 0 時的最大完工時間。訂單 4 於機台 1 加工時間為 16，為所有機台加工時間中最小的值，故第 1 個加入機台排程。此時更新機台 1 的負荷量為 16。訂單在考慮機台負荷量後的機台完工時間如表 6，選擇最小值之訂單 6 於機台 2 加工，其完成時間為 30，接著再次更新機台負荷量。表 3-7 顯示最後一張訂單 3 需在機台 1 加工，該批次在起始時間為 0 的最大完工時間即為訂單 3 之 51。 B_1 於機台上之排程如圖 3-3， B_1 中機台 1 的加工順序依序為訂單 4-訂單 3，機台 2 加工訂單 6， B_1 在機台上的完工時間為 51。

表 5 B_1 的各機台加工時間

	訂單 3	訂單 4	訂單 6
p_{i1}	35	16	50
p_{i2}	40	48	30

表 6 訂單 4 加工後剩餘訂單於各機台加工完成時間

	訂單 3	訂單 6
p_{i1}	51	66
p_{i2}	40	30

表 7 訂單 6 加工後剩餘訂單於各機台加工完成時間

	訂單 3
p_{i1}	51
p_{i2}	70

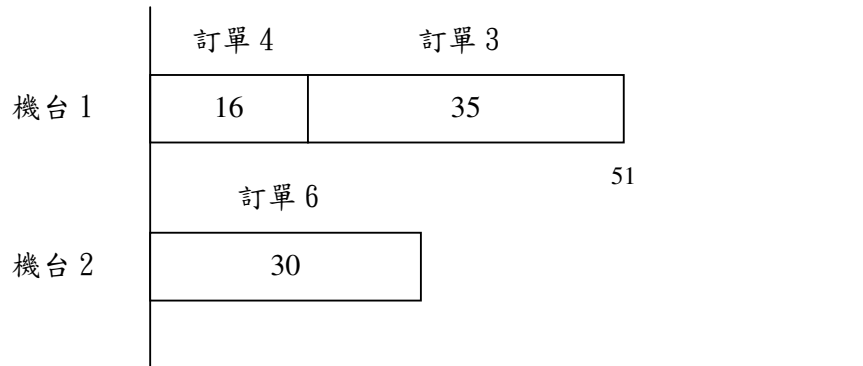


圖 7 B_1 於機台上加工之甘特圖 (假設開始時間為 0)

同理，利用演算法 B 求得 B_2 機台排程之結果如圖 7， B_1 中機台 1 的加工順序依次訂單 4-訂單 3，機台 2 加工訂單 6，批次之最大完工時間為 39。

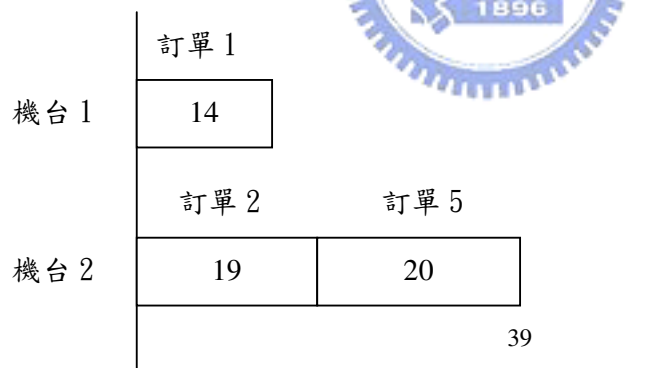


圖 8 B_2 機台排程之甘特圖

(iii) 決定批次在機台上加工的順序

由演算法 A 與 B，我們求得兩批次之最大完工時間以及運送時間，以演算法 C 求得該批次權重最大完工時間如表 8，並將其依遞增的方式排序。由於 $H_1=17.0$ 小於 $H_2=46.2$ ，故批次的加工順序為 B_1-B_2 ， B_1 之發車時間即為 51，重新將訂單依照批次之順序，最後將訂單排入機台中；如圖 9 所示，各機台加工流程中 B_2 的最大完工時間往前移動至 69。

表 8 權重最大完工時間

	H_r
B_1	17.0
B_2	46.2

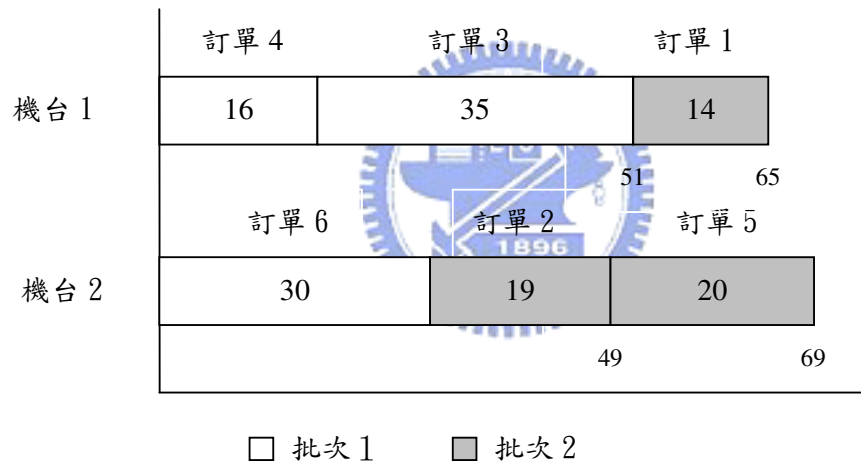


圖 9 各機台加工流程圖

(iv) 一個起始解組合與計算目標函數值

由 (i)、(ii) 以及 (iii)，本研究求得一起始解之組合如表 4 以及表 9 所示。在此組合下其對應之訂單完成時間與加權後訂單總完成時間列於表 10，4469 即為本研究起始解的目標函數值。

表 9 機台加工順序與完工時間表

機台 1	訂單 4	訂單 3	訂單 1
完工時間	16	51	65
機台 2	訂單 6	訂單 2	訂單 5
完工時間	30	49	69

表 10 起始解訂單花費時間與目標函數值表

	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
w_j	3.1	2.5	2.9	4.7	1.8	6.2
c_j	257	372	234	198	123	147
$w_j c_j$	797	930	679	931	221	911
$\Sigma w_j c_j$	4469					

步驟三. 更新禁忌表列、計算目標函數值更新為 $f(x)^*$

在求得此範例之起始解後，將此解加入禁忌表列中，將目標函數值更新成為目前搜尋法之最佳解 $f(x)^*$ 。

步驟四. 產生目前解之鄰域

本研究鄰域共有三種鄰近解的類別：

(i) 批次內運送順序之兩兩交換

此例起始解之批次內皆為 3 張訂單，如圖所示，訂單與訂單交換其運送順序成為一鄰近解，範例中此類鄰近解數共 6 個。

(ii) 批次於機台加工順序之交換

起始解只 2 個批次，故鄰近解僅 2 個。

(iii) 不同批次中訂單之兩兩交換

第三類型之鄰近解為不同批次中訂單之兩兩交換，在交換批次時，同時交換兩張訂單在機台加工與運送時的順序，其組合共有 9 種。

步驟五. 移步的選擇

本研究檢視此範例之鄰近解並計算其目標函數值，在鄰域中以 B_1 中的訂單 6 與訂單 4 兩張訂單交換其運送順序，為所有鄰近解中成本最低的一組，其解之組合如表 11 與表 12，目標函數值之計算過程列於表 13，

此解之目標函數值為 4220，與起始解之目標函數值比較後，改善率約為 3%。移步至此解並將其記入禁忌表列中；同時因為此目標函數值低於起始解，故更新此解為目前之最佳解。至此搜尋法之第一次迭代結束，搜尋法將以此解做為第二次迭代中鄰域產生的基準，目前禁忌表列記錄起始解以及目前移步之解，禁忌表列確保搜尋法在第二次迭代時不會選取起始解為移步，防止搜尋法陷入重覆搜尋的情況。

表 11 最佳移步之機台加工順序與完工時間

B_1	訂單 4	訂單 6	訂單 3
運送時間	78	129	165
B_2	訂單 5	訂單 1	訂單 2
運送時間	54	188	303

表 12 最佳移步之訂單運送順序與到達時間

機台 1	訂單 4	訂單 3	訂單 1
完工時間	16	51	65
機台 2	訂單 6	訂單 2	訂單 5
完工時間	30	49	69

表 13 最佳移步之訂單花費時間與目標函數值

	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
w_i	3.1	2.5	2.9	4.7	1.8	6.2
c_j	257	372	216	180	123	129
$w_j c_j$	797	930	626	846	221	800
$\Sigma w_j c_j$	4220					

第四章 電腦測試與結果分析

本章以模擬實驗的方法來評估不同因子下對本研究所發展 Tabu_PDP 所找出之最佳近似解的影響，並測試 Tabu_PDP 的求解品質與效率。其中 4.1 探討 Tabu_PDP 參數的最佳組合，4.2 說明模擬問題的設計方式，而 4.3 節則分析是電腦測試結果與改善率。本研究利用 Dev C++ 軟體將演算法以 C 語言撰寫成為程式並在 AMD 64 Sempron 2800+ CPU 1G RAM 環境進行電腦模擬與測試。

4.1 Tabu_PDP 參數設計

本節中探討 Tabu_PDP 中禁忌表列長度與停止準則兩個重要參數的設定值。在測試參數設定值時，希望在問題複雜度適中與程式運算時間不要太長的情況下進行。故本研究選定問題之訂單數為 80 張、訂單在機台的加工時間由均勻分配 $U(20,80)$ 隨機產生、運送時間自 $U(150,250)$ 產生、機台數目為 4 台而車容限制為 5 個單位。本研究針對需要測試的每種參數組合進行 10 次的隨機實驗，並以此 10 次實驗之平均目標函數值為主要的評估標準。

4.2.1 禁忌表列長度

以往在使用禁忌搜尋法求解問題時，禁忌表列長度為影響最佳近似解品質的一個重要參數。目前文獻中大多利用 Glover(1990)在其研究中所指出的數字 7 作為禁忌表列之長度；然而本研究所探討之問題與目前文獻中所探討的問題模型有所不同，故需測試禁忌表列長度對目標函數值的影響。因此本研究針對預設問題，在 50 次迭代下測試 3、5、7、9、11、13 以及 15 這 7 種不同的禁忌表列長度與目標函數值之間的關係，所得結果如圖 10。由圖 10 中可知，當禁忌表列長度為 7 時演算所求得之最佳近似解品質較佳，禁忌表列長度大於 7 後目標函數值的差異甚小。故本研究採用 Glover (1990)所提出之建議，在測試演算法時將禁忌表列長度設定為 7。

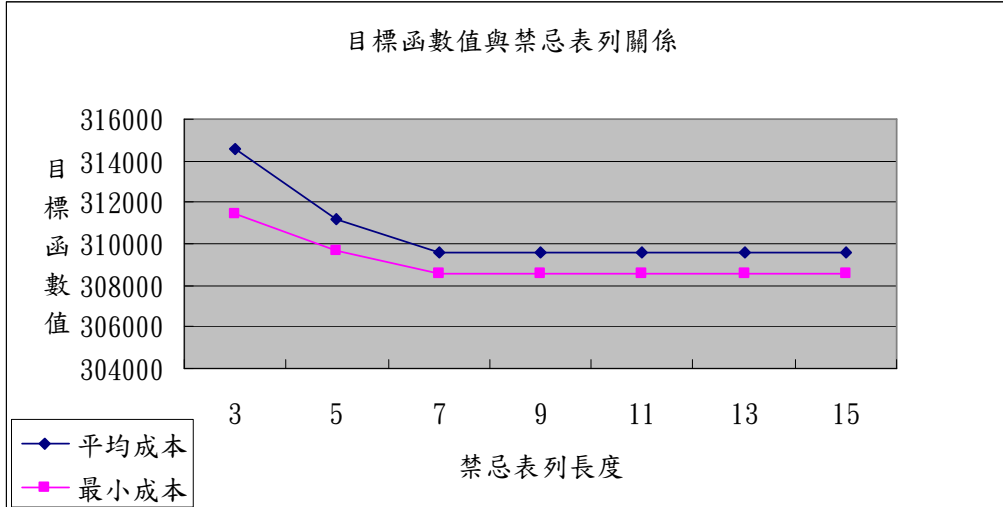


圖 10 禁忌表列長度與目標函數關係

4.2.2 停止準則

在確定禁忌表列長度為 7 後，本小節討論迭代次數對最佳近似解收斂程度以及運算時間的影響。由於禁忌搜尋法產生鄰近解之特性，在訂單數目增加時，會使得一次迭代中可行解的數量增加，進而增加了一次迭代所需要的運算時間。故在相同迭代數下訂單數愈大，運算時間愈長。本研究利用預設問題的 10 組樣本測試 0 到 200 次迭代數與目標函數值以及運算時間之關係後之結果如圖 11，其中橫軸為迭代次數、圖左之縱軸為目標函數值、圖右之縱軸為程式運算時間。由圖 11 中可看出，在目標函數值隨著迭代數的增加而下降。當迭代次數在 100 以前，問題之平均目標函數值有較大幅度的降低，然而在 100 次迭代以後，目標函數值的改善率即趨於平緩。

本研究亦測試了 500 之迭代數，發現目標函數值與在迭代數為 200 僅有非常小的差異。故對目標函數值而言，迭代次數為 100 時最佳。此外，本研究希望 Tabu_PDP 的運算時間具效率性，因此需探討迭代次數對運算時間之影響。由圖 11 中亦可看出，在預設問題中迭代次數與平均運算時間呈線性成長的現象；在 200 張訂單大小的情況下，100 次迭代所需的運算

時間在 1.5 個小時左右，而使用 500 次迭代則需花費約 7 個小時。考量 Tabu_PDP 在 100 次迭代後目標函數改善幅度趨於平緩的特性，並假設其運算時間在合理的範圍內，本研究將演算法之停止準則設為達 100 次迭代即停止搜尋。

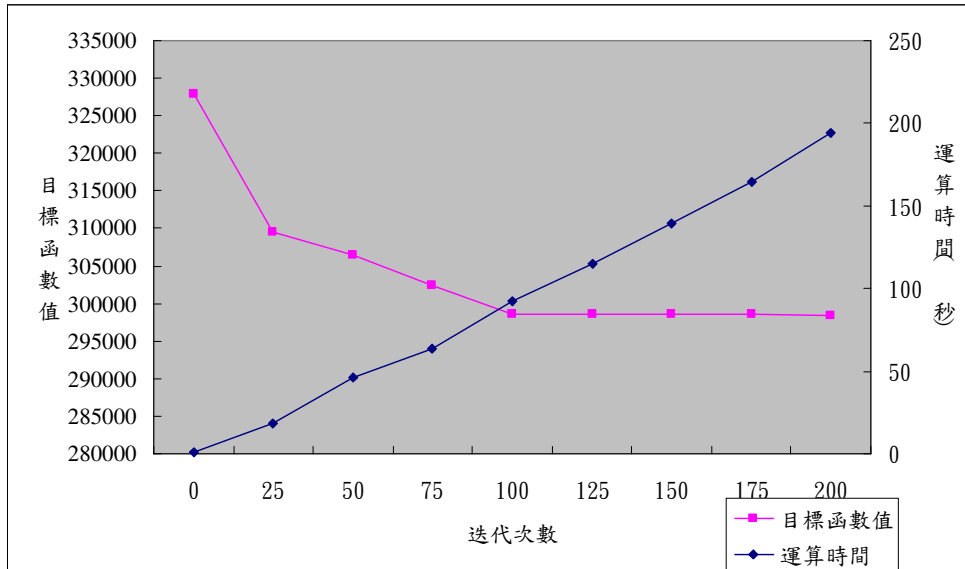


圖 11 迭代次數與目標函數值、運算時間關係

4.2 測試 Tabu_PDP 之求解品質

在確定了 Tabu_PDP 的最佳參數設定後，本研究進一步測試 Tabu_PDP 求解問題時所能找到的最佳解與真實理想解之差距（即求解之品質）。由於目前尚無研究能找出本研究所探討問題的真實最佳解，故本研究考慮一個單一機台且所有訂單權重皆為 1 的問題（此問題為本研究所探討問題的一個特例），利用窮舉法求得其真實理想解並與分別與 T_Start 和 Tabu_PDP 所得之結果比較。在禁忌表列長度為 7 與迭代次數為 100 次的參數設定下，測試 10、20 以及 30 三種不同訂單數量、加工時間與運送時間皆服從 $U(100,400)$ 以及車容限制為 5 單位與 10 單位的 6 組問題。結果如表 14 所示，其中 T_Start 解與 Tabu_PDP 最佳解分別表示利用 T_Start 與 Tabu_PDP 所找出

的解，而T_Start相對誤差的計算方式為 T_Start 相對誤差=(T_Start解 - 真實最佳解)/ 真實最佳解*100%；Tabu_PDP相對誤差的計算方式為Tabu_PDP相對誤差=(Tabu_PDP最佳解 - 真實最佳解)/ 真實最佳解*100%。由表 14 中可看出，T_Start相對誤差介於6.65%與9.5%之間，平均值為7.44%；而Tabu_PDP相對誤差之平均值為0.72%，最大值不超過1.1%。除此之外，Tabu_PDP針對問題編號1（訂單數10與車容限制為5）可求得與真實理想解相同的目標函數值（為20744），而對於其它問題亦可找出與真實理想解非常接近的解。由此測試結果可看出T_Start能找出與真實理想解差距在10%內的解，而更進一步以Tabu_PDP搜尋的話，可達到與真實理想解非常接近的效果。

表 14 真實理想解與 T_Start、Tabu_PDP 最佳解之相對誤差

編號	訂單數	車容限制	真實理想解	T_Start 解	T_Start 相對誤差 (%)	Tabu_PDP 最佳解	Tabu_PDP 相對誤差 (%)
1	10	5	20744	22124	6.65	20744	0.00
2	10	10	31347	34324	9.50	31670	1.02
3	20	5	50101	53436	6.66	50369	0.53
4	20	10	67408	72751	7.93	67890	0.71
5	30	5	131956	141020	6.87	133400	1.08
6	30	10	162791	174322	7.08	164439	1.00

4.3 測試問題設計

由於本研究所探討問題並無適合之標準問題可供測試，故為了測試Tabu_PDP的求解品質與效率，本研究以電腦模擬的方式產生一系列的測試問題。模擬的資料主要分成兩個部分，分別為各訂單在不同機台上的加工時間以及代表顧客與物流中心相對距離的顧客運送時間矩陣。在產品製造階段，本研究以U(20,80)、U(150,250)、U(700,900)三種不同大小的均勻分配隨機產生訂單於各機台加工之時間，此三種分配的平均數呈等比級數增加，變異數近似於等比級數。在成品配送階段，本研究產生U(150,250)

的數據作為測試所需的運送時間矩陣。由於加工時間與運送時間組合的不同，會有訂單加工時間大於運送時間（情境 L）、訂單加工時間近似運送時間（情境 M）以及訂單加工時間小於運送時間（情境 S）三種不同的情境，以測試演算法與研究問題目標函數值間之關係。

測試問題將訂單數量這個因子分為 10、80 與 200 三種水準，目前文獻中求解此類兩階段問題的最大訂單數為 120 (陳昱皓，2006)，本研究將其增加至 200。此外，機台數與車容上限亦為影響目標函數值的因子，在假設當訂單數增加時工廠可選擇加設機台的條件下，本研究在 10 張訂單時會有 2 個機台可供訂單加工，在 80 張訂單時可選擇 2 部或 4 部機台加工，在 200 張訂單時則有 2 部、4 部或 8 部機台三種選擇；在車容上限則有 5 個單位以及 20 個單位兩種情況作為這個因子的水準。由上述不同情境的組合，在排除車容限制大於訂單數的組合後可得 33 組模擬實驗，列於表 15。為方便識別測試問題的性質，本研究依測試問題的內容為其命名，其命名規則為：情境+訂單數_機台數_車容限制。如測試問題 1 之訂單數為 10、機台數為 2、車容限制為 5，而加工時間為 U(20,80)屬情境 S，故將測試問題 1 命名為 S10_2_5。

表 15 測試問題表

編號	測試問題	機台加工時間	訂單數	機台數	車容限制
1	S10_2_5	U(20,80)	10	2	5
2	S80_2_5		80	2	5
3	S80_2_20				20
4	S80_4_5			4	5
5	S80_4_20				20
6	S200_2_5		200	2	5
7	S200_2_20				20
8	S200_4_5			4	5
9	S200_4_20				20
10	S200_8_5			8	5
11	S200_8_20				20

12	M10_2_5	U(150,250)	10	2	5
13	M80_2_5		80	2	5
14	M80_2_20				20
15	M80_4_5			4	5
16	M80_4_20				20
17	M200_2_5		200	2	5
18	M200_2_20				20
19	M200_4_5			4	5
20	M200_4_20				20
21	M200_8_5			8	5
22	M200_8_20				20
23	L10_2_5		U(700,900)	10	2
24	L80_2_5	80		2	5
25	L80_2_20				20
26	L80_4_5			4	5
27	L80_4_20				20
28	L200_2_5	200		2	5
29	L200_2_20				20
30	L200_4_5			4	5
31	L200_4_20				20
32	L200_8_5			8	5
33	L200_8_20				20

4.4 測試結果及分析

本研究所探討整合產品製造與成品配送問題，由於目前文獻並無可供比較求解品質的標竿，因此本研究將 Tabu_PDP 所得之最佳近似解與 T_Start 所產生的可行解比較。本研究在禁忌表列長度為 7 以及迭代次數為 100 的參數設定下，針對於 4.3 節中所設計之 33 種測試問題產生隨機樣本進行電腦模擬測試，測試結果列於表 16。表 16 中記錄各問題起始解(以下稱為 T_Start 解)之目標函數值、最佳近似解(以下稱為 Tabu_PDP 最佳解)之目標函數、改善率以及運算時間。由於 T_Start 的運算時間皆在 1 秒內完成，故表 15 直接被計算在運算時間內。而改善率的計算方式如下：

$$\text{改善率} = (\text{T_Start 解} - \text{Tabu_PDP 最佳解}) / \text{T_Start 解} * 100\%$$

表 16 Tabu_PDP 測試結果

編號	測試問題	T_Start 解	Tabu_PDP 最佳解	改善率	運算時間(秒)
1	S10_2_5	45062	39573	12.18	5
2	S80_2_5	501518	454841	9.31	59
3	S80_2_20	1025328	946240	7.71	53
4	S80_4_5	327954	298524	8.97	85
5	S80_4_20	826090	734209	11.12	77
6	S200_2_5	2197425	2007077	8.66	2182
7	S200_2_20	3585214	3391388	5.41	2006
8	S200_4_5	1289594	1178337	8.63	3108
9	S200_4_20	2566480	2346394	8.58	3005
10	S200_8_5	859428	776710	9.62	5650
11	S200_8_20	2126453	1866694	12.22	5450
12	M10_2_5	79255	77024	2.81	5
13	M80_2_5	1463655	1407954	3.81	65
14	M80_2_20	2236041	2188612	2.12	60
15	M80_4_5	836195	772868	7.57	100
16	M80_4_20	1410857	1366754	3.13	78
17	M200_2_5	8216493	7813960	4.90	2036
18	M200_2_20	10160290	10005719	1.52	2047
19	M200_4_5	4279577	4084874	4.55	2756
20	M200_4_20	5783207	5697236	1.49	2958
21	M200_8_5	2358079	2247973	4.67	5675
22	M200_8_20	3738536	3623616	3.07	5473
23	L10_2_5	230049	221122	3.88	5
24	L80_2_5	5504361	5214798	5.26	59
25	L80_2_20	7164618	7074030	1.26	53
26	L80_4_5	2880642	2751850	4.47	84
27	L80_4_20	3899944	3815557	2.16	77
28	L200_2_5	32465652	31305407	3.57	2157
29	L200_2_20	36590868	35544369	2.86	2124
30	L200_4_5	16272272	15660434	3.76	2988
31	L200_4_20	18886198	18408377	2.53	3009
32	L200_8_5	8526116	8205672	3.76	5672
33	L200_8_20	10348580	10186398	1.57	5467

為了進一步探討影響運算時間與改善率的相關變項，本研究對表 16 所得測試結果使用複迴歸分析(Analysis of Variance, AVOVA)，討論各因子對本研究所設計演算法之運算時間與改善率的影響，以下探討各因子與運算時間與改善率的問題作更深一步的探討。

4.4.1 Tabu_PDP 運算時間複迴歸分析

以運算時間為因變數的複迴歸分析結果如表 17 所示，其中 $R^2=0.962$ 、調整後的 $R^2=0.924$ ，代表表示四個自變數可解釋 92% 運算時間的變異。在 $\alpha=0.05$ 時，無論在情境 L、情境 M 以情境 S 三種不同的情境的因子水準下，加工時間對運算時間並無顯著影響，即 Tabu_PDP 的求解速度不受加工時間與配送時間之相對關係改變而有所影響。除此之外，車容限制對運算時間亦並無顯著影響。

表 17 Tabu_PDP 運算時間複迴歸分析

Regression Summary for Dependent Variable: 運算時間 R= .96165988 R2= .92478973 Adjusted R2= .91404540 F(4,28)=86.072 p<.00000 Std.Error of estimate: 610.24						
	Beta	Std.Err. of Beta	B	Std.Err. of B	t(28)	p-level
Intercept			-2235.04	390.9791	-5.71651	0.000004
加工時間分配	0.000272	0.051827	0.68	130.1044	0.00524	0.995856
訂單數	0.576690	0.058610	17.05	1.7325	9.83946	0.000000
機台數	0.562590	0.057928	532.23	54.8012	9.71197	0.000000
車容限制	-0.065075	0.052592	-17.86	14.4328	-1.23735	0.226235

註：顯著水準為 0.05

訂單數為一個影響運算時間的顯著因子，推論其原因為增加訂單數會使得 Tabu_PDP 一次迭代所能產生之鄰近解數目增加，因此運算時間亦增加。而另一個對運算時間有顯著影響之因子為機台數，如同訂單數一樣，增加訂單數亦會使得 Tabu_PDP 一次迭代所能產生之鄰近解數目增加，因此運算時間亦增加。

由於訂單數與機台數皆對於運算時間有顯著影響，圖 12 為 Tabu_PDP

運算時間表，橫軸為訂單數與機台數，縱軸為以秒為單位之運算時間，由圖中可看出 Tabu_PDP 解 S10_2_5 問題僅需要 5 秒左右，而解 S200_8_20 的問題則需花費 5450 秒（約 91 分鐘），這是由於本研究所探討之問題為 NP-hard，故當問題規模增加時，Tabu_PDP 於求解問題的運算時間會大幅成長。當運算時間為決策中之重要考量時，建議使用 Tabu_PDP 於訂單數（同時也表示顧客數）小於 80 的問題，在此類的問題中，Tabu_PDP 可快速於 60 秒內找到近似解。

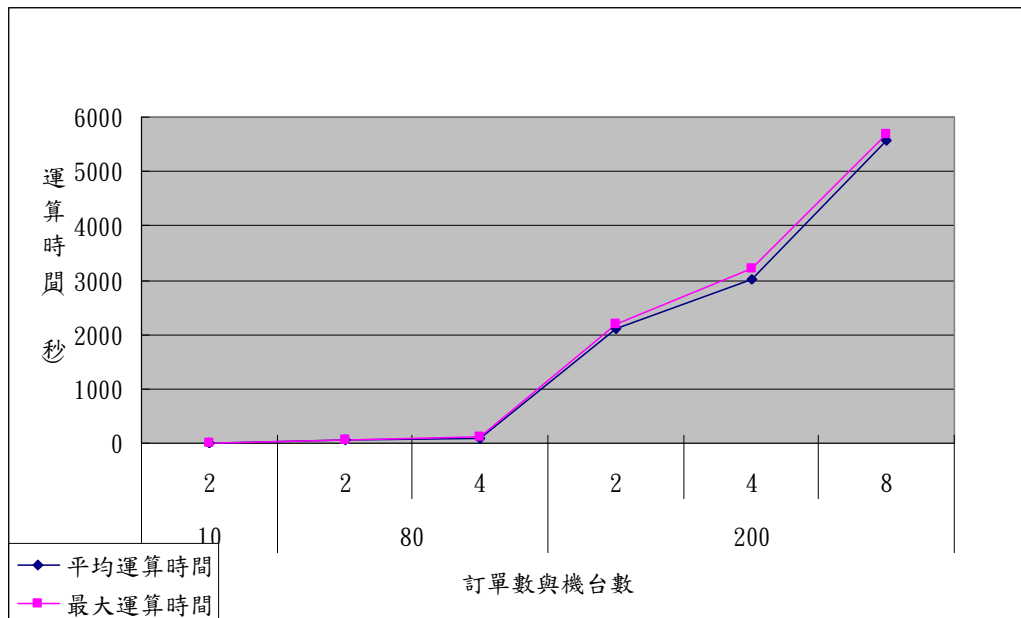


圖 12 Tabu_PDP 訂單數、機台數與運算時間關係

4.4.2 Tabu_PDP 改善率複迴歸分析

以改善率為自變數的複迴歸分析結果列於表 18，其中 $R^2=0.697$ 、調整後的 $R^2=0.654$ ，代表表示四個自變數可解釋 65% 改善率的變異。由表中可看出，在 $\alpha=0.05$ 時，Tabu_PDP 的求解品質不受訂單數與機台數影響。除此之外，複迴歸分析結果顯示加工時間對改善率有顯著影響，而加工時間不同其實表示了加工時間與配送時間之相對關係。當因子水準落於 U(20, 80)（即加工時間小於配送時間，情境 S）時之平均改善率為 9.31%，而在

U(150, 250)與 U(700, 900)兩因子水準下的改善率皆在 4%左右。U(20, 80)高出其它兩因子水準的改善率 2 倍。

表 18 Tabu_PDP 改善率複迴歸分析

Regression Summary for Dependent Variable: 改善率 R= .83495789 R2= .69715468 Adjusted R2= .65389106 F(4,28)=16.114 p<.00000 Std.Error of estimate: 1.9277						
	Beta	Std.Err. of Beta	B	Std.Err. of B	t(28)	p-level
Intercept			12.70779	1.236048	10.28931	0.000000
加工時間分配	-0.774355	0.104000	-3.06006	0.410981	-7.44575	0.000000
訂單數	-0.149145	0.117610	-0.00694	0.005473	-1.26814	0.215193
機台數	0.177948	0.116240	0.26501	0.173109	1.53086	0.137025
車容限制	-0.248029	0.105535	-0.10715	0.045591	-2.35021	0.026040

註：顯著水準為 0.05

本研究推論這種現象是由於 Tabu_PDP 在車輛途程部分，其產生的可行解數目較多，使得一次迭代中的鄰近最佳解出現在交換運送順序或批次間訂單的情況機會較大；且該因子水準下之目標函數值較情境 M 或情境 L 下低許多，在改善率 = $(T_Start \text{ 解} - \text{Tabu_PDP 最佳解}) / T_Start \text{ 解} * 100\%$ ，T_Start 解與 Tabu_PDP 解之改善量相同時，分母愈小所得改善率愈大。因此在加工時間小於運送時間時，其改善率會明顯高於加工時間相當於運送時間的情況，而造成 Tabu_PDP 求解品質受加工時間與運送時間相對關係影響。

除了加工時間之外，車容限制對改善率亦有顯著影響。推論其發生原因為車容限制增加相當於批次數減少，會使得透過批次內運送順序的交換所產生之可行解增加，批次間的訂單交換之可行解減少。而 Tabu_PDP 鄰近最佳解常為批次間訂單之交換，其可行解會有同時減少批次在機台上完工時間與訂單運送時間的機會，使得 Tabu_PDP 改善量較大。故由於批次間訂單交換的可行解減少關係，車容限制大會使得改善率下降。

4.5 小結

整體而言，Tabu_PDP 的平均改善率在 5%。對於研究問題而言，在所測試的 33 個問題中，Tabu_PDP 對於求解加工時間小於配送時間而車容限制為 5 的問題 S10_2_5、S80_2_5、S80_4_5、S200_2_5、S200_4_5 以及 S200_8_5 表現較佳，平均改善率在 9.56%。其中 S200_2_5、S200_4_5 以及 S200_8_5 因為訂單數影響，運算時間皆在 30 分鐘以上。Tabu_PDP 對於問題 M200_2_20、M200_4_20、L80_2_20 以及 L200_8_20 之改善率皆低於 2%，其中 M200_2_20、M200_4_20、L200_8_20 的運算時間因受訂單數影響，運算時間皆在 30 分鐘以上。

故本研究建議針對訂單數大於 80 而車容限制為 20 的這些改善率較低而運算時間長的問題，以 T_Start 所找出的可行解作為問題之最佳近似解為較佳，訂單數小於 80 而車容限制為 5 的問題採用 Tabu_PDP 求解為較佳選擇。而訂單數大於 80 但車容限制為 5 的問題，若主要需求為運算時間短，則選用 T_Start 求解；若需求為求解品質佳，則使用 Tabu_PDP 求解。

第五章 結論與未來發展方向

5.1 結論

由於越來越多企業選擇整合產品製造與成品配送兩階段，並採用接單式生產作為製造類型，使得產品直接在生產完畢後配送至顧客手上，因此最佳化整合產品製造與成品配送兩階段問題逐漸成為供應鏈中的重要議題。本研究以總加權後訂單完成時間為系統的績效指標，在考慮車容限制的情況下，設計一禁忌搜尋法求解整合供應鏈中產品製造及成品配送兩階段問題，期望研究結果能作為相關產業參考之依據，讓符合問題模型之業者不在依賴經驗法則，而是用科學化的方法能夠在合理的時間內做出較佳的決策以降低製造與配送的成本，進而達到整體利潤最大化。

本研究所發展出的生成起始解的演算法 T_Start 為一簡單的啟發式演算法，電腦測試結果顯示此方法可於數秒內找到品質不錯的可行解，適用於運算資源受限或需要快速反應的情況；而本研究所發展出的以禁忌搜尋法為基礎的 Tabu_PDP，可以在合理的時間內平均改善 5% T_Start 所找出的解，適用於對成本比較敏感且可接受較長規劃時間的情況。

5.2 未來發展方向

本研究成功地發展一套演算法於求解產品製造與成品配送兩階段問題，但在進行研究的過程中發現仍有許多不足之處，故最後一個小節提出以下的建議與未來能夠發展之方向：

5.2.2 演算法方面

1. 就改善率而言，可嘗試在交換法則上設定優先順序或使用其它可行的方法，以改善當加工時間與車容限制在不同水準時，求解品質不穩定的現象。
2. 對運算時間部分，針對禁忌搜尋法之交換法則再作更深一步之討

論，發展出更具效率性的鄰近解定義方法，以加快演算法的運算速度。

3. 比較並結合其他巨集式啟發演算法，如蟻群演算法或基因演算法。擷取其長處，以期能求得具有有效性與穩健性兼備的最佳近似解。
4. 發展可找出真實最佳解的演算法或是找出問題的目標下限值 (lower bound)，以用於比較啟發式解法的真實效果。

5.2.1 整合產品製造與成品配送兩階段問題方面

1. 考慮加入適用於其他情境的假設，如加入時窗限制、整備時間或卸貨時間的限制等，以期問題能更符合真實情境。
2. 探討不同的目標函數，如加入與運送相關之操作成本或是與滿足交貨期限要求的目標，以更符合實際應用的需求。



參考文獻

- 周碩鴻 (2006)。應用基因演算法求解供應鏈中生產排程與物流配送兩階段總成本最小化問題。國立交通大學工業工程與管理學系，碩士論文。
- 陳昱皓 (2006)。塔布搜尋法於求解供應鏈中整合生產排程與成品配送兩階段問題，國立交通大學工業工程與管理學系，碩士論文。
- 陳正雄 (2000)。塔布搜尋法在塑化排程之應用-以 BOPP FILM 為例。元智大學工業工程研究所，碩士論文。
- 夏萬春 (2000)。塔布搜尋法於車輛排班之探討。國立高雄第一科技大學機械工程研究所，碩士論文。
- 蔡志強 (2003)。以蟻群系統建立物流宅配最佳化配送路徑規劃。國立屏東科技大學工業管理系，碩士論文。
- Allahverdi, A. and J. Mittenhal (1994). Scheduling on M parallel machines subject to random breakdowns to minimize expected mean flow time. Naval Research Logistics, 41(5), 677-682.
- Azizoglu, M. and O. Kirka (1999). Scheduling jobs on unrelated parallel machines to minimize regular total cost functions. IIE Transactions, 31, 153-159.
- Bodin, L.D., B.L. Golden, A.A. Assad and M.O. Ball (1983). Routing and scheduling of vehicle and crews-the state of the art. Computer & Operations Research, 10, 69-211.
- Booker, L. B., D. E. Goldberg and J. H. Holland (1989). Classifier systems and genetic algorithms. Artificial Intelligence, 40(1-3), 235-282.
- Bruno, L.G., E.G. Coffman and R. Sethi (1974). Scheduling independent tasks to reduce mean finishing time. Communications of the ACM, 17, 382-387.

- Chang, Y. C. , C. Y. Lee (2003). Logistics scheduling: analysis of two-stage problems. Journal of System Science And Systems Engineering, 12(4), 371-393.
- Chang, Y. C. , C. Y. Lee (2004). Machine scheduling with job delivery coordination. European Journal of Operation Research, 158(2), 470-487 .
- Chen, J.F. (2006). Minimization of maximum tardiness on unrelated parallel machines with process restrictions and setups. International Journal of Advanced Manufacturing Technology, 29(5), 557-563.
- Chen, Z.L. and G.L. Vairaktarakis (2005). Integrated scheduling of production and distribution operations. Management Science, 51(4), 614-628.
- Chiu, H.N. (1995). The integrated logistics management system: a framework and case study. International Journal of Physical Distribution & Logistics Management, 25(6), 4-22.
- Christofides, N., A. Mingozzi, and P. Toth (1981). Exact algorithm for the vehicle routing problem, based on Spanning Tree and shortest path relaxations. Mathematical Programming, 20, 255-282.
- Class, C.A., C.N. Potts and P. Shade (1994). Unrelated parallel machine scheduling using local search. Mathematical and Computer Modeling, 24(9), 11-19.
- Colomi, A., M. Dorigo, and V. Maniezzo (1991). Distributed optimization by ant colonies. F. J. Varela and P. Bourguine (Ed.), In Proceedings of the First European Conference on Artificial Life (pp. 134-142). Cambridge, Massachusetts: MIP Press.
- Dorigo, M. and L. M. Gambardella (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1), 53-66.

- Fisher, M.L. and R. Jaikumar. (1978). A decomposition algorithm for large-scale vehicle routing. working paper 78-11-05, Department of Decision Sciences, University of Pennsylvania.
- Fisher, M.L. and R. Jaikumar (1981). A generalized assignment heuristic for vehicle routing. Networks, 11, 109-124.
- Garey, M.R. and D.S. Johnson (1979). Computers and intractability: A guide to the theory of NP-completeness. New York: W.H. Freeman and Company.
- Gendreau, M., A. Hertz and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. Management Science, 12, 76-90.
- Ghirardi, M. and C.N. Potts (2005). Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach. European Journal of Operational Research, 165(2), 457-467.
- Glover, F. (1989). Tabu search: Part I. ORSA Journal on Computing, 1, 190-206.
- Glover, F. (1990). Tabu search: Part II. ORSA Journal on Computing, 2, 4-32.
- Hall, N.G. and C.N. Potts (2003). Supply chain scheduling : batching and delivery. Operation Research, 51(4), 566-584.
- Hall, N.G. and C.N. Potts (2005). The coordination of scheduling and batch deliveries. Annals of Operations Research, 135, 41-64.
- Herrmann, J., J.M. Proth and N. Sauer (1997). Heuristics for unrelated machine scheduling with precedence constraints. European Journal of Operational Research, 102,(3): 528-537.
- Ho, S.C. and D.Haugland (2004). A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. Computers & Operations Research, 31, 1947–1964.
- Holland, J.H. (1992). Genetic algorithms. Scientific American, 267(1), 66-72.

- Homberger, J. and H. Gehring (2005). A two-phase hybrid meta-heuristic for the vehicle routing problem with time windows. European Journal of Operational Research, 162, 220–238.
- Horn, W.A. (1973). Minimizing average flow time with parallel machines. Operations Research, 21, 846–847.
- Hsieh, J.C., P.C. Chang and L.C. Hsu (2003). Scheduling of drilling operations in printed circuit board factory. Computers & Industrial Engineering, 44, 461-473.
- Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi (1983). Optimization by simulated annealing. Science, 220(4598), 671-680.
- Lee, C.Y. and Z.L. Chen (2001). Machine scheduling with transportation considerations. Journal of Scheduling, 4, 3-24.
- Lee, I. (2001). Artificial intelligence search methods for multi-machine two-stage scheduling with due date penalty, inventory, and machining costs. Computers & Operations Research, 28, 835-852.
- Li, C. L., G. Vairaktarakis, C. Y. Lee. (2003). Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research, 164, 39–51.
- Liaw, C.F., Y.K. Lin and C.Y. Cheng (2003). Scheduling unrelated parallel machines to minimize total weighted tardiness. Computers & Operations Research, 30(12), 1777-1789.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research, 59, 345-358.
- Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, (1953). Equations of calculation by fast computing machines, Journal of Chemical Physics, 21(6), 1087-1092, 1953.

- Mokotoff, E. and J.L. Jimeno (2002). Heuristics based on partial enumeration for the unrelated parallel processor scheduling problem. Annals Journal of Operational Research, 117(1-4), 133-150.
- Nowicki E., and C. Smutnicki (1998). The flow shop with parallel machines: A tabu search approach. European Journal of Operational Research, 106(2-3):, 226-253
- Osman, I. H. (1993). Meta-strategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, 41, 421-451.
- Piersma, N. and W. Van Dijk (1996). A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search. Math Computer Modeling, 24(9), 11-19.
- Pirlot, M (1996). General local search methods. European Journal of Operational Research, 92(3), 493-511.
- Reeves, C.R. (1993). Improving the efficiency of tabu search for machine sequencing problems. Journal of the Operational Research Society, 44(4), 375-382.
- Renaud, J., G. Laporte and F.F. Boctor (1996). A tabu search heuristics for the multi-depot vehicle routing problems. Computer & Operations Research, 23(3), 229-235.
- Sarmiento, A. M., R. Nagi (1999). A review of integrated analysis of production-distribution systems. IIE Transactions 31 (11), 1061-1074.
- Suresh, V. and D. Chaudhuri (1994). Minimizing maximum tardiness for unrelated parallel machines. International Journal of Production Economics, 34(2), 223-239.
- Suresh, V. and D. Chaudhuri (1996). Bicriteria scheduling problem for unrelated parallel machines. Computer & Industrial Engineering, 30(1), 77-82.

Thomas, D. J. and P. M. Griffin (1996). Coordinated supply chain management.
European Journal of Operational Research, 94, 1-15.

Weng, M. , J. Lu and H. Ren (2001). Unrelated parallel machine scheduling
with setup consideration and a total weighted completion time objective.
International Journal of Production Economics, 70, 215–226.

