

國立交通大學  
工業工程與管理學系

碩士論文

應用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

Solving A Two-Stage Problem with Scheduling and  
Delivery in Supply Chain by Tabu Search Algorithm

研究生：陳昱皓

指導教授：張永佳 博士

中華民國九十五年七月

# 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

研究生：陳昱皓

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

供應鏈強調整體效益之觀念日益受到重視，然而不同的階段追求的目標函數不同，會造成供應鏈彼此間上下游的利益衝突，因而無法達到整體效益的最佳化。本研究以供應鏈中最為重要的兩個階段產品製造與物流配送為問題主軸，利用非等效平行機台(unrelated parallel machine)模擬產品製造和車輛途程問題(vehicle routing problem, VRP)模擬物流配送，同時考慮此兩階段的目標函數，以整體最佳化為目的。但由於非等效平行機台與車輛途程問題皆被證明為未定多項式難度(NP-Hard)，因此整合此兩階段之問題的複雜度至少也為未定多項式難度，即當問題規模大到一定程度時，無法在合理時間內求得最佳解。本研究採用巨集式啟發式演算法(meta-heuristic)中的塔布搜尋法(tabu search)求解上述兩階段問題，以最小化總成本為目的。並使用國際題庫與模擬資料測試所開發出之塔布搜尋法的穩定度及有效性，期以有效率的方式找出供應鏈中整合產品製造與物流配送兩階段問題的最適解。

**【關鍵字】** 非等效平行機台、車輛途程問題、未定多項式難度、巨集式啟發式演算法、塔布搜尋法

# **Solving A Two-Stage Problem with Scheduling and Delivery in A Supply Chain by A Tabu Search Algorithm**

Department of Industrial Engineering and Management

National Chiao Tung University

Student: Yu-Hao Chen

Advisor: Yung-Chia Chang

## **Abstract**

The concept of emphasizing on profit earned across an entire system in a supply chain has been an important subject discussed in supply chain management. Since different stages often have different, sometimes conflicting, objectives, globally optimal integrated solution for an entire system is often difficult to achieve. This research studies an integrated problem that jointly considers two important stages in a supply chain: product manufacturing and finished good delivery. An unrelated parallel machine scheduling problem is used to simulate product manufacturing and a vehicle routing problem is applied to represent delivery of finished goods. The objective is to find a system-wide solution to minimize the total cost across the entire system. Both unrelated parallel machine scheduling problem and vehicle routing problem are NP-Hard, the complexity of the studied two-stage problem is also NP-Hard. It means the solution time will grow exponentially when problem size goes large. Therefore, a tabu search algorithm, one of the meta heuristics, is presented to solve this problem. Computational analysis based on international studied problems and simulated data are presented to test the effectiveness and stability of the proposed tabu search algorithm.

**【key word】** unrelated parallel machine scheduling 、 vehicle routing problem 、 NP-Hard 、 meta heuristic 、 tabu search algorithm

## 誌謝

歲月匆匆，兩年的研究所生涯即將要畫下句點了，能夠順利畢業，最要感謝的是我的指導教授張老師，多虧有老師的耐心指導才能順利完成本論文，其次也要感謝口試時給我許多寶貴建議的兩位口試委員唐老師、梁老師。當然還有陪伴我的實驗室夥伴，即將成為貴婦人的珮君、永遠都很忙的柏詠，還有實驗室學長學弟們，幫我很多的惠誠學長、梅竹英雄學君、辛苦報帳的國維、熱心助人的丞舜、渴望交女友的天威，還有其他實驗室的同學，嘴巴很機車的啟峰、超級鹹濕的正琪、創意無限的鈺堂、愛打嘴砲的修來、人脈超廣的長科、女人緣極佳的喬凱、說話有點激動的俊榮、眼神很犀利的莉安、學成歸國的幸穎以及努力做人中的盈杰。

最後，感謝老爸辛苦努力工作，使我在求學過程中無後顧之憂；感謝我媽時時叮嚀我熬夜身體不好；當然還有對我最好的哥哥，在我需要幫助時給我適時的援助，謝謝大家。



# 目錄

摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	vii
第一章 緒論.....	1
1.1 研究動機與背景.....	1
1.2 研究目的.....	3
1.3 研究方法流程.....	3
1.4 研究架構.....	4
第二章 文獻探討.....	5
2.1 非等效平行機台與車輛途程問題.....	5
2.1.1 巨集式啟發式演算法應用於非等效平行機台排程.....	5
2.1.2 巨集式啟發式演算法應用於車輛途程問題.....	10
2.2 供應鏈整合.....	15
2.3 塔布搜尋法.....	17
第三章 研究方法.....	21
3.1 問題描述.....	21
3.2 演算法之設計.....	22
3.2.1 塔布搜尋法.....	22
3.3.2 搜尋策略.....	23
第四章 電腦模擬與測試結果.....	31
4.1 產生測試問題.....	31
4.2 模擬結果.....	31
4.2.1 測試問題 D.....	32
4.2.2 測試問題 B.....	34
4.2.3 測試問題 P.....	36
4.2.4 測試問題 D、B、P 之比較.....	38
4.3 小結.....	38
第五章 結論與建議.....	40
5.1 結論.....	40

5.2 未來研究方向.....	40
參考文獻.....	41
附錄一 配送時間.....	417



## 圖目錄

圖 3-1	兩兩交換法之起始解示意圖 .....	28
圖 3-2	兩兩交換法之鄰近解示意圖 .....	28
圖 3-3	塔布搜尋法流程圖 .....	30
圖 4-1	訂單數與運算時間之關係圖(測試問題 D).....	33
圖 4-2	訂單數與改善率之關係圖(測試問題 D).....	34
圖 4-3	訂單數與運算時間之關係圖(測試問題 B).....	35
圖 4-4	訂單數與改善率之關係圖(測試問題 B).....	36
圖 4-5	訂單數與運算時間之關係圖(測試問題 P).....	37
圖 4-6	訂單數與改善率之關係圖(測試問題 P).....	38



## 表目錄

表 2-1 非等效平行機台使用的目標函數 .....	5
表 2-2 車輛途程問題型態劃分 .....	10
表 2-3 車輛途程問題的五個時期 .....	12
表 3-1 訂單之權重與加工時間表 .....	25
表 3-2 訂單之目的地距離矩陣 .....	25
表 3-3 訂單優先排序表 .....	25
表 3-4 機台負荷時間表 .....	26
表 3-5 第一階段訂單完成時間 .....	26
表 3-6 各訂單之總完成時間 .....	27
表 3-7 總成本之計算 .....	27
表 4-1 測試問題 D 之測試結果表格 .....	33
表 4-2 測試問題 B 之測試結果表格 .....	35
表 4-3 測試問題 P 之測試結果表格 .....	37





# 第一章 緒論

## 1.1 研究動機與背景

隨著資訊科技的蓬勃發展，人與人的距離不再是遙不可及，處在競爭日益激烈的環境裡，面對要求越來越高的客戶，企業必須改變其經營運作方式以求生存。許多企業為了降低生產成本或提高獲利契機，期望各類的策略或是合作方式降低生產成本或製造風險，甚至將風險轉移到供應鏈上其他廠商。長期下來此種方式反而造成整條供應鏈的潰散，連帶影響企業本身。因此我們不該只著重供應鏈的其中一層，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，這也就是供應鏈管理的精神所在。

一般來說供應鏈中包含許多角色，如：原料供應商、製造商等等，而產品製造與貨物配送是兩個能讓產品在供應鏈中形成與流動的重要功能。以現實生活中的郵局為例，當我們到郵局要寄送快遞郵件時，需經由窗口服務人員加以處理，再由郵務人員依照規劃之路線分送快遞郵件到各個目的地。上述的例子可看成一個簡單的供應鏈，第一階段就為郵件處理，其功能在於「產品製造」；第二階段為郵件投遞，其功能為「貨物配送」。郵件在第一階段被處理時，通常重視的是如何盡快的將郵件處理完成；而在第二階段運送郵件時，通常基於成本考量會希望使用最少的車輛以最小之行車距離完成配送工作。由郵局的例子就可以看出，在同一條供應鏈下，不同的階段使用的績效指標不一定相同，而造成各階段所追求的績效指標往往相互衝突，因而無法達成對供應鏈整體最佳化的目標。因此如何以整體觀點同時考慮供應鏈上的各階段，就實務應用面上來看，確實具有深入研究探討之必要性。目前探討供應鏈整合觀點的研究，大多以資訊系統的觀點，針對資訊共享的部分作探討居多，如：劉素雲[5]透過供應鏈的整合模式，達到企業上下遊資訊共享的目的，以適時提供顧客需求的

產品或服務。而少有研究以作業觀點探討供應鏈整合模式，因此本研究以作業觀點將供應鏈中較為重要的生產排程與物流配送作一整合。

本研究所探討的問題由兩個階段組成。第一階段考慮生產排程問題，生產排程為一種資源分配決策的問題，係將有限的產能、人力或空間分配到各個工作或顧客，讓系統內有限的資源作最有效的運用，以達到最大的效益。現實生活中的機台或是服務人員的效率通常不一，為合理地模擬實際生產情況，本研究選定以非等效平行機台問題(unrelated parallel machine problem) 來代表第一階段的生產環境。所謂平行機台是指有  $N$  個待加工的工件，需經過某一加工程序，而有  $M$  台機台具有相似的功能作此加工處理，此  $M$  台機台即是平行機台。而非等效的意義則是指機台的加工效率彼此獨立且沒有一定的關係；第二階段考慮產品的運輸配送，本研究將其模擬為車輛途程問題(vehicle routing problem, VRP)。所謂車輛途程問題為一組車隊，從一個物流中心運送貨品到已知顧客的手中，最後回到物流中心，且一個顧客只能被服務一次，載貨量不能超過車容量限制。

本研究以成本為主要考量，整合兩階段的績效指標。而在兩台完全相同的平行機台生產型態下，各工件完成時間分別乘上不同的權重，將時間轉換成金錢，以成本為績效衡量指標，此種問題被歸屬為未定多項式難度(NP-Hard) [45]，因此更為複雜的非等效平行機台排程問題如以成本為考慮目標，其複雜度至少為未定多項式難度；傳統車輛途程問題也經由 Savelsbergh[49]證明為未定多項式難度，故整合此兩階段的問題複雜度至少也為未定多項式難度。所謂未定多項式難度是指問題會因為問題規模的增大，求解時間會呈現指數成長。當求解此類未定多項式難度的問題時，規模較小時仍能利用傳統的數學規劃或是套裝軟體(Lingo 或是 Cplex)求解，但是當規模變大時，可能無法在合理時間內求得最佳解，甚至可能無法求得最佳解。由此可知此一兩階段問題，當規模變大到一定程度時想求得最佳解是不太容易的。

因此近年來為了解決此類未定多項式難度的問題，許多巨集式啟發式演算法(meta-heuristic)慢慢被發展應用，如：基因演算法(genetic algorithms, GA)、類神經網路(neural networks, NN)、模擬退火法(simulated annealing, SA)以及塔布搜尋法(tabu search, TS) 等等，是較為常用的巨集式啟發式演算法。巨集式啟發式演算法最大的優點在於解決問題時，能夠快速找到品質佳的近似解，且能夠跳脫出區域最佳解限制的機機會比較大，此外求解時間也在合理可接受的範圍內。因此在處理較為複雜的問題時，應用巨集式啟發式演算法是較為合理的解決方式，雖然並非保證是最佳解，但是使用較短的時間找出接近最佳解的近似解，在分秒必爭的時代是有其價值的。故本研究選用塔布搜尋法來求取近似解，因塔布搜尋法具有彈性記憶結構，可避免重複之前搜尋過的解，藉由此彈性記憶結構可增大跳脫出區域最佳解的機機會和擁有快速搜尋的機制，可提升搜尋時的效率與解的精準度。



## 1.2 研究目的

本研究所探討的問題由兩階段組成，第一階段為非等效平行機台排程問題，第二階段為車輛途程問題。本研究主要目的為應用塔布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，希望能在合理的時間內求得品質佳的近似解。

## 1.3 研究方法流程

本研究是以塔布搜尋法求解二階段問題，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現實情境。
2. 設計演算法：應用塔布搜尋法設計一套演算法，最後以 C++ 程式進行程

式的撰寫，求解非等效平行機台與車輛途程兩階段整合問題。

3. 實驗分析：探討在不同情境與參數設定下，塔布搜尋法求解的速度與近似解的品質。
4. 結論與建議：根據電腦求解結果作一結論且建議未來研究方向。

## 1.4 研究架構

本研究共由五個章節所組成，第一章敘述研究動機背景以及研究目的；第二章針對非等效平行機台、車輛途程問題以及塔布搜尋法等相關文獻作一系列的回顧；第三章為研究方法，應用塔布搜尋法設計一套演算法；第四章利用本研究所設計之演算法撰寫程式，求解非等效平行機台與車輛途程兩階段整合問題，分析比較在不同問題情境下，演算法的求解速度與近似解品質；第五章提出本研究之成果與結論，並建議未來之研究方向。



## 第二章 文獻探討

### 2.1 非等效平行機台與車輛途程問題

本小節將介紹本研究的兩階段問題，包含生產排程與物流配送。第一階段為生產排程的非等效平行機台排程問題，第二階段為物流配送的車輛途程問題。本小節分別針對這兩個階段作一文獻回顧。

#### 2.1.1 巨集式啟發式演算法應用於非等效平行機台排程

非等效平行機台(unrelated parallel machine)為平行機台加工型態的一種，所謂平行機台是指有  $N$  個待加工的工件，需經過某一加工程序，而有  $M$  台機台具有相似的功能作此加工處理，此  $M$  台機台即是平行機台。因此，非等效平行機台即是表示相同工件在每台平行機台加工時間皆不一樣，且機台間彼此間無任何相關性。

非等效平行機台排程問題種類甚多，但其中以考慮整備時間的相關問題居多，而在考慮整備時間時，會因為工件批次生產與否、加工順序不同等等的因素而使得整備時間有所不同，因此會衍生出許多類型的問題。而且每一個問題都有追求的目標函數，不同的目標函數也會造成最後的結果不同，使用的目標函數經由 Hart et al.[24]整理而得，如表 2-1 所示。而在這些目標函數中，又以  $C_{\max}$  和  $L_{\max}$  是最為常用的目標函數，最大完工時間最小化能夠提高機台的使用率，而最大延後時間最小化則是要讓工件盡量在交期前完成。

表 2-1 非等效平行機台使用的目標函數

目標函數符號	意義
$C_{\max}$	所有工件中最大完工時間
$\bar{C}$	一個工件的平均完工時間
$F_{\max}$	所有工件中最長流程時間

$\bar{F}$	加工一個工件的流程時間
$L_{\max}$	最大延後(lateness)時間,提早完工 L 值為負的
$\bar{L}$	平均延遲完工時間, 提早完工 L 值為負的
$T_{\max}$	最大延遲(tardiness)時間,提早完工 T 值為零
$\bar{T}$	平均延遲完工時間, 提早完工 T 值為零
NT	延遲的工件數
$E_{\max}$	最大提前完工時間

而在求解方法中，早期學者最常使用精確法(分支界限法、線性規畫法)解決非等效平行機台排程問題，如：在 1978 年 Lawler and Labetoulle[31] 使用線性規畫(linear programming)方法求解非等效平行機台雙目標( $C_{\max}$  與  $L_{\max}$ )的問題，將加工時間乘上權重轉換為成本，延遲時間當成懲罰成本，目標函數為總成本最小化。精確法之優點在於能求得最佳解，但求解過程費時費力，問題規模越大就越可能無法在合理的時間內求得最佳解。因此便有學者提出啟發式演算法來解決此類未定多項式難度問題，期能在短時間內找到近似解。但隨著時代的進步，問題趨於複雜多元化，規模也越變越大，在求解過程中容易陷入區域最佳解的困境。因此又有學者提出改良式的啟發式演算法，主要目的為增大在搜尋最適解過程中跳脫出區域最佳解的機會，我們將之稱為巨集式啟發式演算法，用來和之前的啟發式演算法作一區別。下面簡單介紹幾種較為常用的巨集式啟發式演算法及其應用在非等效平行機台排程的相關文獻。

#### (1) 模擬退火法：

是由 Metropolis 等人在 1953 年提出一個模擬降溫的演算法，應用冶金

方面的觀念，使溫度慢慢將低，達到最佳狀態而得到最佳的冶金效果。而三十年後 Kirkpatrick 等人發現要將溫度達到物質能量最低狀態猶如尋求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 的模擬方式，可以應用來解一些困難的組合尋優問題。而模擬退火法應用在非等效平行機台之相關文獻有：

Chinyao[14]以模擬退火法求解多階段流線型生產排程(multi-stage flow shop scheduling)問題，且包含了整備時間彼此獨立與卸料時間彼此相依相關假設，而每一個階段皆包含了非等效平行機台，追求目標函數為  $F_{max}$ 。結論中作者比較了自己本身提出的模擬退火法與 Pham and Karaboga 的模擬退火法搭配不同的起始解，從模擬結果中發現好的起始解對於改善過程是相當重要的。除此之外作者提出的模擬退火法可幫助廣度化與深度化的搜尋，對求解時間大有幫助。

Kim et al.[29]以模擬退火法求解非等效平行機台排程問題，且包含了批量生產與相依整備時間，其追求的目標函數為  $T_{max}$ 。文中作者認為針對本問題需提出新的參數設定方能做有效的搜尋，僅依靠 Kirkpatrick 等人所提出較為傳統的模擬退火法是不夠的，因此自己設計了一套模擬退火法，其中包含六種產生鄰近解的方法，然後與 Kirkpatrick 等人的模擬退火法比較，其結果不論是解的品質或是求解時間皆比 Kirkpatrick 等人的模擬退火法較佳。最後作者也建議其他巨集式啟發式演算法如塔布搜尋法和基因演算法也可能是理想的解決方法。其他有關模擬退火法應用於非等效平行機台的文獻還可參考 Chen[13]、Kim et al.[30]。

## (2) 基因演算法：

此法最早是在 1970 年代由 Holland 提出，主要利用生物學遺傳優選的原理，利用個體間的配對與刺激產生突變，來產生更好的下一代，然後將此種生命科學現象應用於求解最佳化的問題。以下介紹基因演算法應用於非等效平行機台之相關文獻。

Fatima [18]基因演算法求解零工式生產排程(job shop scheduling)的問題，包含非等效平行機台，以 $C_{max}$ 為目標函數，將結果與Ghedjati與Ghedjati and Pomerol的實驗相互比較，顯示基因演算法可改善近似解最多達到11%。

Ewa[17]以基因演算法和變數產生法(column generation technique)求解非等效平行機台問題，包含前置作業時間，以 $C_{max}$ 與切換機台(Changeover)成本為目標函數，將結果與作者提出之下界值作一比較，結果顯示以此一兩階段啟發式演算法求解之結果，其切換機台成本不會大於加工時間之成本的十分之一。

Ruiz and Maroto[48]以基因演算法求解流線型生產型態，且每一站皆包含非等效平行機台，並考慮相依整備時間，以 $C_{max}$ 為目標函數，結果顯示較之其他啟發式演算法優異許多。

(3) 塔布搜尋法：塔布搜尋法利用彈性記憶結構的機制，可記錄之前已搜尋過的解，避免重複搜尋，增加搜尋效率以及跳脫出區域最佳解的機會，其應用於非等效平行機台相關文獻有：

Suresh[51]以塔布搜尋法求解多目標非等效平行機台之排程問題，且不包含前置作業時間，本問題主要有兩個目標函數，分別為 $C_{max}$ 與 $T_{max}$ ，最後並將結果與GAP-EDD演算法比較，顯示塔布搜尋法在解的品質和運算時間均表現得較為優異，而未來可朝在動態情境下的問題深入研究。

Hsieh et al.[27]以塔布搜尋法求解印刷電路板(printed circuit board)工廠中的鑽孔作業問題，在製程中每一工作站的鑽孔機台效率不一，故可視為非等效平行機台排程問題，本研究以切換機台成本、使用機台數、利用率當作目標函數，最後與人工排程(manual approach)作一比較，結果顯示塔布搜尋法明顯優於人工排程。

除了上述三種較為常用之求解方法之外，其他也有游擱鴻 [3]應用拉



格蘭茲鬆弛法(lagrangean relaxition)與臨域搜尋法(neighborhood search)，求解非等效平行機台之  $T_{max}$  排程問題，建議如使用塔布搜尋法或基因演算法可能會得到更好的結果。

Lee and Pinedo[34]求解非等效機台排程問題時，且考慮工件間存在順序相依的整備時間，以作者提出 ATCS(apparent tardiness cost with setups)法則求解總加權延遲時間最小化問題(minimizing the sum of the weighted tardiness)。此法分成三個階段，分別為第一階段計算各工件之相關參數，以統計觀念、因素分析建構工件之權重，第二階段建構一派工法則，最後利用模擬退火法之原理進行求解。

Park et al.[42]則使用類神經網路與一傳統啟發式法求解與 Lee et al.[34]相同條件下的總加權延遲時間最小化問題，並與 Lee 等人所提出的 ATCS 法作比較。

巨集式啟發式演算法因其理論基礎皆不同，故各有不同的搜尋策略，各有其優點，如模擬退火法是以機率的概念決定是否接受移步，搜尋到最佳解的過程較為緩慢，但對於區域深度化的效果相當好；而基因演算法是利用適者生存的觀念，淘汰較差的解，最後留下較佳的解，此法採用多點同步搜尋，對於廣度化搜尋的效果較佳；塔布搜尋法則是有塔布表列的機制，可避免重複搜尋已搜尋過的解，增加搜尋效率。因此有許多學者在求解相同問題的情況下比較數種方法的優劣，更有學者建議混合啟發演算法可能得到更佳的结果，相關文獻如下：

Park and Kim[41]比較模擬退火法及塔布搜尋法在求解等效平行機台排程持有成本(holding cost)最小化問題中包含開始與交期時間限制的表現，而持有成本也即是指總流程時間最小化。結果雖然發現模擬退火法的結果優於塔布搜尋法，但此研究指出這絕對不會是長期穩定的結果，他們與另一些學者 Nowicki et al.[38] and Reeves [46]都相信塔布搜尋法比模擬退火法有更好的長期且穩定的表現。

N.Piersma and Dijk[44]也利用局部搜尋法及鄰近搜尋法來求解非等效平行機台  $C_{max}$  之問題，將結果與基因演算法、模擬退火法、塔布搜尋法等作比較，並建議在非等效平行機台的排程上使用塔布搜尋法可獲得有效的搜尋結果。

Class et al[15] 針對非等效平行機台問題，在  $C_{max}$  的目標下，以有效率的鄰近解配合區域搜尋法(local search)進行求解，並建議塔布搜尋法、模擬退火法及基因演算法如有較佳的鄰近解，且若融合多種演算法之特性可得到較有效率且較佳的搜尋結果。

### 2.1.2 巨集式啟發式演算法應用於車輛途程問題

車輛途程問題是物流配送中的重要議題，在實務上具有很高的應用價值，因此幾十年前便已提出被加以討論，最早是在 1959 年由 Dantzig and Ramser 提出。車輛途程問題可簡單定義為一組車隊，從一個物流中心運送貨品到已知顧客的手中，最後必須回到物流中心，且一個顧客只能被服務一次，載貨量不能超過車容量限制。

但隨著實務上的需要與提高服務水準，車輛途程問題也變得越來越多元化，在學術研究與實務應用上也有不同的延伸與應用，關於車輛途程問題型態甚多，可由表 2-2 來劃分各種類型的車輛途程問題。

表 2-2 車輛途程問題型態劃分

型態	性質
車輛數	一部車輛 多部車輛
車輛種類	單一車種 混合車種
車輛容量限制	所有車輛皆有相同限制 不同車輛有不同限制

	無限制
配送中心	單一配送中心 多個配送中心(起始點不同) 多個配送中心(起始點相同)
需求位置	需求位於節點上 需求位於節線上 混合型
路網方向性	無方向性路網 有方向性路網 混合型路網
貨物類別	同種貨物 多種貨物
需求型態	需求量已知且固定 需求量成隨機分布
里程限制	所有車輛有相同里程限制 不同車輛有不同的里程限制 無限制
時間限制	所有車輛有相同時間限制 不同車輛有不同的時間限制 無限制
裝卸作業型態	裝貨 卸貨 混合型
最佳化目標	最小固定及變動成本 最小總路線成本 最小使用車輛數

資料來源：逢甲大學物流系統規劃實驗室[2]

其中最常被提出討論的即是時窗性車輛途程問題(vehicle routing problems with time window, VRPTW)，而有關時窗限制車輛途程問題的文獻整理可參考 Liu and Shen[36]。除時窗問題之外也有學者研究不確定性環境之相關問題，如 Teodorovic and Pavkovic[54]以模糊理論為基礎，發展出結合掃描法和模糊規則庫的演算法，可處理隨機情境下的問題。

車輛途程問題與非等效平行機台排程一樣皆屬於未定多項式難度的問題，早期也是以精確法求解為主，但近年來問題規模趨於複雜龐大，難

以在合理時間內求得最佳解，因此可以考慮以巨集式啟發式演算法求解，以期在求解大規模問題時，能在合理時間內求得最適解。但隨著時代的不同其適用的求解方法亦不同，可根據 Chao I.-M.等幾位學者在 1998 年所提出車輛途程問題的五個時期加以劃分，如表 2-3 所示。

表 2-3 車輛途程問題的五個時期

年代	演進與演算法
1950	<ul style="list-style-type: none"> <li>● 只能求解規模較小的問題，約十到二十個顧客。</li> <li>● Dantzig 和 Ramser 在 1959 年利用整數規劃模式來解決問題。</li> </ul>
1960	<ul style="list-style-type: none"> <li>● 較大的車輛途程問題，約三十到一百個顧客。</li> <li>● Clarke 和 Wright 在 1964 年提出啟發式節省法(saving methods)來建立車隊配送問題。</li> <li>● Christofides 和 Eilon 在 1969 年利用 2-opt 及 3-opt 方法解決問題。</li> </ul>
1970	<ul style="list-style-type: none"> <li>● 能夠求解大型問題，約一百到一千個顧客。</li> <li>● Gillett 和 Miller 在 1974 年提出兩階段法啟發式的掃描演算法，此法屬於先途程再分群。</li> <li>● Christofides 等人在 1978 年則是提出先分群再途程的方法。</li> <li>● 但電腦的運算速度、效能是一大問題。</li> </ul>
1980	<ul style="list-style-type: none"> <li>● Fisher and Jaikumar 在 1981 年提出利用數學規劃的最佳化方法，可處理約五十個顧客的問題。</li> <li>● Cullen ,Jarvis and Ratliff 在 1981 年建立一種人機互動的啟發式演算法(interactive heuristics)</li> <li>● 同樣的電腦運算速度與效能不足仍是一大問題</li> </ul>
1990	<ul style="list-style-type: none"> <li>● 這時期大都使用模擬退火法、基因演算法、類神經網路法和塔布搜尋法。</li> <li>● Robuste,Daganzo and Souleyrette 在 1990 年、Alfa,Heraguand Chen 在 1991 年都利用模擬退火法來求解車輛途程問題。</li> <li>● Sement and Tailard 在 1993 年、Gendreau,Hertz and Laporte 在 1994 年、Rochat and Taillard 在 1995 年、Xu and Kelly 在 1996 年皆使用塔布搜尋法來求解。</li> </ul>

資料來源：逢甲大學物流系統規劃實驗室[2]

巨集式啟發式演算法主要發展於 1990 年之後，其觀念主要來自於傳統啟發式演算法、人工智慧、進化論等等，並使用學習策略來反覆搜尋，

以期更有效率的達到最佳解，以下介紹幾種應用巨集式啟發式演算法求解車輛途程問題的相關文獻。

(1) 模擬退火法：

Breedam[10]以模擬退火法求解車輛途程問題，其目標函數為最短行車距離，最後作者將自己提出之改良模擬退火法所得到的結果，以十四個國際題庫與其他學者所提出塔布搜尋法和混合式演算法比較，其結果雖然並非每一個都有較佳表現，但作者確信以其降溫過程在未來的研究能夠得到更佳的解。

Wu[57]以模擬退火法求解多場站車輛途程問題，以最短行車距離為目標函數，最後與Perl[43]比較之下結果較佳，此外未來可朝向初始解的產生作深入研究，對於搜尋過程有幫助，且另一方面可針對不確定性需求作深入研究以更符合現實。



(2) 基因演算法：

Baker[7]以基因演算法求解車輛途程問題，在不同的車輛數下，以最小行車距離為目標函數，再以模擬退火法、塔布搜尋法之結果作一比較，結果雖然不如塔布搜尋法，但在短期內有顯著的改善，且求解時間均較其他方法來的短。

Hwang[28]以改良式基因演算法求解多場站車輛途程問題，以運送成本最小化為目標函數，將結果與國際題庫比較，顯示此一改良式基因演算法在求解車輛途程問題方面有極佳的潛力。

(3) 塔布搜尋法：

Renaud et al.[47]考慮在車輛容量及路線長度的限制的情況下，應用塔布搜尋法求解多場站的車輛巡迴路線問題，以最短路線為目標函數並發展一演算法稱為FIND，在該演算法中包含快速改善(fast improvement)、廣

度化及深度化三個階段，該演算法測試了十一個國際標竿問題，與Chao等人提出的十二個新的問題，發現該演算法能計算出較好的解。

Cordeau et al.[16]提出一個unified塔布搜尋法求解具時窗性車輛途程問題，包含週期性與多場站之車輛途程問題，以最短行車距離為目標函數，結果顯示除了可求出效果不差的近似解之外，更強調其求解的速度。

Gendreau et al.[20]發展了一新的塔布搜尋法為「TABUROUTE」，且能夠從任一起始解開始搜尋，甚至是不可行解也能順利進行搜尋過程，此法用以求解車輛途程問題，結果以國際標竿題庫比較顯示均能得到最佳解。其他以塔布搜尋法求解車輛途程問題相關文獻尚可參考林明俊[1]、Brandao [9]、Stephan[50]、Fermin[19]、Ho[25]、Oppen [39]及Tarantilis[53]。

而在其他求解方法中尚有韓復華等人[6]比較最鄰近法、聲音擾動法與搜尋空間平滑法三種啟發式演算法，並結合傳統交換改善法應用於求解車輛途程問題，以最小行車距離為目標函數，並選擇十一個國際標竿題目測試績效並與塔布搜尋法比較，結果顯示塔布搜尋法的精確度較為精準。Mazzeol[37]以螞蟻演算法求解載重限制之車輛途程問題，將車輛體積乘上一權重以求取利潤最大化，結果顯示當節點增加到五十個時表現仍相當良好。

前述提到巨集式啟發式演算法各有優劣，因此有許多相關文獻針對相同的車輛途程問題比較其優劣，更甚者混合數種巨集式啟發式演算法的特性進而發展出新的巨集式啟發式演算法。Osman[40]利用禁忌搜尋法與模擬退火法用以求解車輛途程問題，以最小總運送成本為目標函數，並針對結果進行比較，發現塔布搜尋法不論是使用車輛數或是行車距離皆要比模擬退火法較為優異。Thangiah et al.[55]延伸Osman[40]所設計的塔布搜尋法中的途程間交換改善法，針對相同的問題，利用塔布搜尋法與模擬退火法的特性，發展出一混合搜尋法來求解，結果顯示此混合搜尋法在求解的品質與運算時間方面表現較為優異；Tan et al.[52]求解時窗性車輛途程問題，

以最小總距離為目標函數，將Thangiah et al.[56]的模擬退火法、Glover的塔布搜尋法改良，提出新的演算法用以求解此問題，以Soloman之例題進行分析比較，結果顯示塔布搜尋法與模擬退火法與改良前比較均得較佳的結果，且運算時間也在合理範圍內。

Breedam[11]有鑒於塔布搜尋法與模擬退火法皆適用於求解未定多項式難度的問題，因此針對車輛途程問題，以最小行車距離為目標函數，比較模擬退火法、塔布搜尋法與降落式演算法(descend heuristic, DH)。結果顯示當問題規模小時，差異性尚不明顯，而當問題規模變大時，模擬退火法與塔布搜尋法就明顯優於降落式演算法，主要是因為模擬退火法與塔布搜尋法能針對問題特性設計演算法，因此能得到較好的結果，最後作者更建議如果能融合模擬退火法之緩慢降溫特性與塔布搜尋法之長期記憶架構，可能能得到更好的結果。

Homberger[26]以一混合式演算法求解時窗性車輛途程問題，而此一方法包含 $(\nu, \lambda)$ -演進策略(evolution strategy)與塔布搜尋法，以最少使用車輛數與最短行車距離為目標函數，結果顯示在求解大規模問題時，能得到較佳的結果。

## 2.2 供應鏈整合

供應鏈強調重視整體效益的觀念日益受到重視，如：Walker[56]將多媒體服務其中的兩個階段作一整合，主要探討伺服器與客戶間的關係，第一階段為伺服器，探討訊號運作問題，為求在最短時間內將信號傳送出去；第二階段為客戶，探討訊號發送到客戶端的問題，為求以最小成本將信號傳送到客戶端，雖都是為了提升服務品質，但目標函數不一致，勢必得作一整合。然而有些單一階段的問題就已經複雜難解，如：本研究之非等效平行機台排程和車輛途程問題，皆屬於未定多項式難度的複雜度，整

合此兩階段的問題複雜度至少也為未定多項式難度，較少學者研究整合多階段的最佳化問題，大部分皆僅針對供應鏈中的單一階段作深入研究。至目前為止所搜尋到的文獻，以作業觀點整合多階段問題都是應用精確法或數學證明整合模式居多，如：

Lee and Chen[32]探討機台排程問題包含運送問題，其中包含兩種不同的運送情形，一種為半成品經由自動搬運車運送到其他機台繼續加工；另一種為將成品運送到顧客或是倉庫，且包含了運送容量上限與時間限制，主要貢獻為證明了此類問題複雜度為未定多項式難度，必須重新訂定相關假設以簡化問題，才有辦法以動態規劃求解，此外未來研究方向可分為三個：第一、建議此類問題較適合使用演算法求解；第二、證明此種問題在某些特別的情形下是可在多項式時間內求得解；第三、建構更接近真實生活的模式。

Hall and Potts[23]將生產排程與運送視為一個系統，探討包含批量生產排程與運送之供應鏈整合問題，並使用動態規劃來求解，目標為追求整體成本最小化，此研究並說明了整合此兩階段可降低系統總成本至少百分之二十，甚至更多。

Chang and Lee[12]探討一兩階段問題，包含第一階段生產排程，對工件進行加工處理；第二階段物流配送將產品送達顧客手上，作者將此兩階段視為一個系統，以成本最小化為目標函數，然後利用向前求解法(forward approach)和向後求解法(backward approach)求解此物流排程問題，最後以允許最劣情況分析(worst-case)衡量分析這兩個方法所得到的結果。Li et al.[35]延續 Chang and Lee[12]之研究，將所欲服務的顧客數延伸到多個，並使用動態規劃求解。

因整合多階段問題困難度極高，但少有學者應用巨集式啟發式演算法求解相關問題，僅 Lee[33]利用巨集式啟發式演算法求解在印刷電路板(printed circuit board, PCB)製造系統中，包含存貨成本、時窗懲罰和機台



成本時，多機台的二階段問題。其中比較了塔布搜尋法、模擬退火法、基因演算法與鄰近搜尋法，結果顯示塔布搜尋法在解決此類多機台兩階段問題時，在解的品質上優於模擬退火法、基因演算法與鄰近搜尋法，而求解的時間也僅比鄰近搜尋法差，皆比模擬退火法和基因演算法優。

現在學者常用的巨集式啟發式演算法有塔布搜尋法、基因演算法、模擬退火法等等，各種演算法都有其優缺點，沒有絕對的優劣之分，同樣的方法並非適用於相同的問題，必須視問題類型來決定使用何種演算法，因此演算法中的參數設計與搜尋策略便是成功與否的重要關鍵。由上述文獻回顧可知，大部分學者認為塔布搜尋法在針對非等效平行機台和車輛途程問題的相關研究，與其他常用啟發式演算法相比較，具有求解時間較短與高品質近似解的優點，另從 Lee[33]之文獻可知，對於整合多階段之問題，塔布搜尋法的表現仍是比其他常用的演算法較佳。且即使結果並非如此，多數學者也相信塔布搜尋法具有快速跳躍式的搜尋機制與具有彈性之記憶結構，能夠快速找到近似最佳解，就長期而言還是會有較為穩定且優異的表現。下一小節為塔布搜尋法相關介紹。

## 2.3 塔布搜尋法

塔布搜尋法最早是由 Glover 於 1977 年提出，為一種近似解法。它可從任一起始初解開始，產生一鄰域，再由鄰域中找一最佳鄰近解作為下次搜尋的起點，如此不斷重複下去，直到達到所設定停止搜尋的準則。主要特色在於塔布搜尋法具有彈性記憶結構，可記錄之前已搜尋過的解，避免重複搜尋而缺乏效率，並且增大跳脫出區域最佳解的機會，那些禁止再次搜尋之解即是塔布(tabu)之義，也可稱之為塔布表列(tabu list)。不過此塔布表列有一固定之長度，並採用先進先出的準則，意即塔布表列會隨著搜尋次數而逐步更新，因此進入塔布表列的解經過一定的搜尋次數之後會給予

解禁，允許往那個區域再次搜尋。塔布搜尋法為一種通用之求解概念，使用者可依照自己需求設計不同的搜尋策略。以下簡短介紹塔布搜尋法之基本組成元素[21,22]。

#### (1) 起始解(start solution)

一般較為常用的是利用已知的啟發式解收斂而得，以作為塔布演算法的起始解。

#### (2) 鄰近解(neighborhood solution)

顧名思義是指在任何可行解當作起始解的情況下，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。假設今有一起始解工件加工順序為  $i \rightarrow j \rightarrow h$ ，設計者將鄰近解定義為左右工件順序互換，則鄰近解有  $j \rightarrow i \rightarrow h$  或是  $i \rightarrow h \rightarrow j$  兩種，塔布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)

在所有鄰近解中，選取最佳的移動路徑，但可能由於最佳的鄰近解在塔布表列中，且尚未滿足期望水準(aspiration criterion)，故不能選取此移動路徑，所以必須從剩下的鄰近解中，在選取其最優的移動路徑，移動到下一個最優鄰近解。

#### (4) 塔布表列

在上述的移動過程中，並非所有的移動路徑皆是可行的。為求避免在求解過程中陷入循環解的困擾，因此設立了一塔布表列，利用塔布搜尋法的記憶結構記憶前一次的移動路徑，為免重複搜尋，會讓此移動路徑進入塔布表列，且經過一定搜尋次數之後，才予以解禁並准予再次搜尋。由此可知塔布表列會隨著搜尋次數而不斷更新，當有一新的鄰近解進入塔布表列時，也會有一塔布表列的解被解禁。

#### (5) 深度化搜尋(intensification)

塔布搜尋的記憶結構又可分為「長期記憶」和「短期記憶」，深度化

搜尋策略即是指「短期記憶」的記憶結構，主要就是針對某一特定區域，深入搜尋此一區域的最佳解，此一記憶長度 Glover 與 Laguna 建議為 7 或  $\sqrt{n}$ ( $n$  為工件數)，不過還是可以依照問題性質或需求加以調整。

#### (6) 廣度化搜尋(diversification)

廣度化搜尋是指「長期記憶」的策略，從一開始便記錄所有的移步過程，當反覆搜尋且一直找不到最佳解時，此時找到的可能只是區域最佳解。為了跳脫出此區域，可利用此長期記憶資料，改變鄰近解之定義或另尋新的起始解，針對較少搜尋的區域再做一次深度化搜尋，以增加達到整體最佳解的機會。

#### (7) 期望水準(aspiration criterion)

前述之塔布表列是為避免陷入循環解的困境所設，但由於某些移動可能會產生很好的可行解，但這些移動可能在塔布表列中，因此為了衡量此移動的價值，必須建立一期望水準，通常此一期望水準皆定為搜尋過程所得到的最佳目標值。假設將期望水準定義為此移動之可行解優於現在的最佳解，如果搜尋到一鄰近解，雖然在塔布表列中，但目標值優於現在的最佳解滿足期望水準的條件，所以必須暫時解除其禁忌，使之成為目前的最佳解。

#### (8) 結束準則(stopping criterion)

一般而言，塔布演算法停止的準則有下列三種情況[4]：

- a. 搜尋次數達到設計者一開始所設定的最多次數或是目標值達到事先所設定的值。
- b. 超過某一設定運算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即可停止。
- c. 運算之鄰近解集合扣掉違反塔布搜尋規則之解集合為空集合可停止。

在面對組合優化問題(combinatorial optimization problems)時，使用巨

集式啟發式演算法搜尋此類問題的最佳解，常常會遇到一些阻礙，在 Battiti and Tecchiolli[8]的研究中提到，可將這些阻礙對應到「動態系統理論」中的三種現象：(1)區域最佳解，某些區域內的最佳解；(2)限制迴圈(limited cycle)，一直在反覆搜尋某些解而形成迴圈；(3)混沌吸子(chaotic attractor)，不管如何改變搜尋策略都無法脫離某些特定區域。因此在設計演算法時，就要思考如何才能夠突破這三種障礙。



## 第三章 研究方法

### 3.1 問題描述

本研究以非等效平行機台排程與車輛途程問題為問題主軸，將各訂單的總完成時間分別乘上一權重，將時間轉換為成本，而此權重由設計者依訂單重要性自行決定，重要性高的權重值較大，以此總成本為目標函數。

本研究討論的問題敘述如下：系統內有  $n$  個訂單，需經過  $m$  個非等效平行機台中的任意一機台加工，訂單  $i$  在機台  $k$  的加工時間為  $p_{ik}$ 。將訂單於機台加工完畢之後的時間定義為第一階段完成時間  $o_i$ 。於第一階段完工之訂單均經由  $r$  台車的其中一台車輛運送到顧客手上，而每台車之發車時間為所配送的訂單中第一階段完成時間的最大者，再加上從物流中心到顧客手上的時間，即為總完成時間，以  $c_j$  表示，定義為訂單從一開始加工直到送達顧客手上。其中  $d_{ij}$  為顧客  $i$  到顧客  $j$  的配送時間。問題目標函數為  $\text{Min} \sum_{j=1}^n w_j c_j$ ，其中  $w_j$  表示訂單之重要性。為求簡化問題本研究訂定基本假設如下：

1. 所有訂單之起始時間皆相同。
2. 機台加工沒有整備時間。
3. 訂單在任一機台中的加工時間已知。
4. 不考慮有重工及當機的情況。
5. 一個顧客只下一個訂單，任意兩顧客間的距離已知。
6. 單一物流中心單一車種，車子的速度皆一致，且無車容量限制。
7. 車輛運送僅有卸貨無載貨工作。
8. 一輛車只送一次貨

## 3.2 演算法之設計

### 3.2.1 塔布搜尋法

本研究是以塔布搜尋法求解排程與配送兩階段問題，以總成本最小化為目標函數，此兩階段問題屬於組合最佳化問題，可用下列簡單模式表示：

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

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

$x$ ：表示訂單完工順序的一種組合

$X$ ：表示訂單完工順序組合之所有集合

$f(x)$ ：表示  $x$  所對應的目標值函數

塔布搜尋法基本流程如下：

**Step 1**：設定鄰近解的定義、迭代次數(iterations)、塔布表列與期望水準。

**Step 2**：隨機選取一起始解  $x$ ，此時塔布表列為空集合，然後計算  $x$  對應之目標值  $f(x)$ ，使之成為目前的最佳解  $f(x)^*$ 。

**Step 3**：評估目前最佳解  $x$  的所有鄰近解。

**Step 4**：從所有鄰近解中選取一個新的解，其對應於目標值為鄰近解中最小，且不在塔布表列中，然後移步到此一新的解。

**Step 5**：更新塔布表列、期望水準和長期記憶結構，計算更迭次數。

**Step 6**：如果搜尋到新的解其對應的目標值小於目前的最適解，雖然在塔布表列中，但由於滿足期望水準，故保留此新的解並將之更新成目前的最適解  $f(x)^*$ 。

**Step 7**：當迭代次數超過所預設的次數時停止搜尋，定義目前所得到的解為最適解，否則回到 **Step 3** 繼續搜尋。

### 3.3.2 搜尋策略

根據以往學者使用塔布搜尋法解決各種問題時，可以了解到塔布搜尋法之參數設定對於求解的品質與運算效率有很大的影響，以下為本研究對於各參數之設定。

(1) 起始解：目前國內外相關文獻皆無適用於本研究兩階段問題之下界值，因此本研究根據 Pinedo[45]所提出的考慮權重因素下最短加工時間訂單優先處理法則(weighted shortest processing time first rule, WSPT)的原則設計一啟發式演算法以計算起始解，並作為衡量塔布演算法績效之依據。WSPT 基本概念是以加工時間與訂單權重的比值來決定訂單的順序，比值小者優先排序。本研究發展出之啟發式演算法求解步驟如下。

**Step 1**：先計算每一訂單之權重與在不同機台下加工時間的最小比值，再依照所得到的值將訂單重新排列，並以此重新將訂單編號，使得：

$$\begin{aligned} & \text{Min} (P_{11}/w_1, P_{12}/w_1, \dots, P_{1k}/w_1) \leq \text{Min} (P_{21}/w_2, P_{22}/w_2, \dots, P_{2k}/w_2) \leq \dots \leq \\ & \text{Min} (P_{n1}/w_n, P_{n2}/w_n, \dots, P_{nk}/w_n) \end{aligned}$$

**Step 2**：決定訂單排序之後，依照機台負荷時間決定在那一機台加工。令第  $k$  機台負荷時間為  $l_k$ ， $k=1, 2, \dots, m$ ，初始值均設為零。依照在 **Step 1** 中決定之順序分配訂單，編號小的先分配。在分配第  $i$  個訂單時，分別計算此訂單在每一機器的加工時間與其機台負荷時間之和，以找出其中最小值而將此訂單分配於該機台。假設訂單  $i$  被分配到機台  $q$ ，而該機台之負荷量  $l_q$  就要被更新為  $l_q + p_{iq}$ ，此時間即為訂單  $i$  第一階段完成時間  $o_i$ 。以同樣方法將所有訂單分配給機台。

**Step 3:** 依訂單之第一階段完成時間  $o_i$  由小排到大，完成時間小者優先配送，以此順序再重新將訂單編號，使得： $o_1 \leq o_2 \leq \dots \leq o_n$ 。另外將所有訂單平均分配至欲配送的  $r$  台車輛上，當  $n/r$  為整數時，每台車即配送  $n/r$  個訂單，第一個到第  $n/r$  個訂單即為第一台車所配送；而當  $n/r$  不為整數時，第一個到第  $[n/r]$  個訂單由第一台車配送，其中  $[x]$  表示不大於  $x$  的最大整數，第  $[n/r]+1$  個訂單則視與前後訂單 ( $[n/r]$ 、 $[n/r]+2$ ) 完成時間的差距，差距較小者則第  $[n/r]+1$  個訂單即歸入同一輛車進行配送工作，以此類推決定每台車所欲配送之訂單。每台車之發車時間是依據所配送的訂單中，在第一階段完成時間最大的值。

**Step 4:** 在決定每一台車所配送的訂單之後，接著要決定同一台車之訂單配送順序，由於車輛配送是從物流中心出發，因此第一步就是要找出第一個欲服務的顧客點。尋找方法為將物流中心到各個顧客點的所需時間除以各個顧客(訂單)的權重，取值最小的為第一個服務的顧客。接著再尋找第二個欲配送的顧客，尋找方法為計算從第一個顧客點到剩餘尚未服務顧客點的時間除以尚未服務顧客之權重，取值最小的為第二個服務的顧客。以此類推將每台車所欲配送之訂單順序排好。

**Step 5:** 計算每一工件之總完成時間，將之乘上權重轉換為成本，將所有成本加總即為目標函數。

以下以一簡單例子說明求解過程，現有 5 張訂單、2 台機台、2 輛車，其相關資料如表 3-1 與表 3-2 所示。



表 3-1 訂單之權重與加工時間表

	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5
$w_i$	3.21	1.49	1.07	5.59	6.58
$p_{i1}$	68	72	84	56	89
$p_{i2}$	34	104	101	48	59

表 3-2 訂單之目的地距離矩陣

$d_{ij}$	物流中心	顧客 1	顧客 2	顧客 3	顧客 4	顧客 5
物流中心	0	281	174	463	482	430
顧客 1	281	0	454	210	221	213
顧客 2	174	454	0	634	654	596
顧客 3	463	210	634	0	26	76
顧客 4	482	221	654	26	0	102
顧客 5	430	213	596	76	102	0

**Step 1:** 計算每一訂單之權重與在不同機台下加工時間的最小比值，然後再依照所得到的值決定排序順序，值小的訂單優先排序，從表 3-3 可看到排序順序為訂單 4、訂單 5、訂單 1、訂單 2 與訂單 3。

表 3-3 訂單優先排序表

	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5
$p_{i1}/w_i$	21.04784	<b>48.44428</b>	<b>78.86291</b>	9.93445	13.59334
$p_{i2}/w_i$	<b>10.7215</b>	69.92242	94.69498	<b>8.50982</b>	<b>8.927996</b>
訂單順序	3	4	5	1	2

**Step 2:** 決定訂單排序順序之後，依據接著機台負荷量來決定訂單再哪一機台加工，見表 3-4。首先從訂單 4 開始分配，訂單 4 分配於機台 1 與機台 2 之完成時間分別為 56 與 48，選擇值最小者，故選擇分配於機台 2，此時機台 2 之負荷時間變為 48；訂單 5 分配於機台 1 與機台 2 之完成時間

分別為  $0+89$  與  $48+59$ ，故選擇分配於機台 1，此時機台 1 與機台 2 之負荷時間分別為 89 與 48，訂單 1 分配於機台 1 與機台 2 之完成時間為  $89+68$  與  $48+34$ ，故選擇較小的值並分配於機台 2，此時機台 1 與機台 2 之負荷時間為 89 與 82；訂單分配於機台 1 與機台 2 之完成時間為  $89+72$  與  $82+104$ ，選擇較小值分配於機台 1，此時機台 1 與機台 2 負荷時間為 161 與 82；分配訂單 3 於機台 1 與機台 2 之完成時間為  $161+84$  與  $82+101$ ，選擇較小的值並分配於機台 2。其訂單之第一階段完成時間分別為 48、89、82、161 與 183。

表 3-4 機台負荷時間表

	訂單 4	訂單 5	訂單 1	訂單 2	訂單 3
$l_1$	0	<b>89</b>	89	<b>161</b>	161
$l_2$	<b>48</b>	48	<b>82</b>	82	<b>183</b>
$o_i$	48	89	82	161	183

**Step 3:** 依據  $o_i$  值找出每一輛車所欲配送的訂單，由於  $\lceil n/r \rceil = 2$ ，故可知訂單 4 與訂單 1 首先被分配於車輛 1，再比較訂單 5 之前後訂單(訂單 1 與訂單 2)完成時間，發現訂單 5 與訂單 1 之差距(7)較小，故將訂單 5 也分配於車輛 1，最後結果為車輛 1 配送訂單 4、訂單 1 與訂單 5；車輛 2 配送訂單 2 與訂單 3，見表 3-5。

表 3-5 第一階段訂單完成時間

	訂單 4	訂單 1	訂單 5	訂單 2	訂單 3
$o_i$	48	82	89	161	183

**Step 4:** 已知第一台車配送訂單 4、訂單 1 與訂單 5，尋找第一個欲配送之顧客點，分別計算物流中心到顧客 4、顧客 1 與顧客 5 的時間與其訂單權重之比值。計算結果分別為 86.19(顧客 4)、87.54(顧客 1)、65.27(顧客 5)，

因此第一個服務順序為顧客 5。接著再計算從顧客 5 到尚未服務顧客的時間與其訂單權重之比值。計算結果為 66.41(顧客 1)與 18.19(顧客 4)，所以第二個服務為顧客 4。以此類推求得車輛 1 訂單配送順序為訂單 5、訂單 4 與訂單 1；以此類推車輛 2 訂單配送順序為訂單 2 與訂單 3。

**Step 5:** 由表 3-6 可知車輛 1 發車時間為 89，車輛 2 發車時間為 183，計算各訂單之總完成時間為表 3-6 所示，將各訂單之總完成時間分別乘上各訂單之權重，然後加總即為目標函數總成本，見表 3-7。

表 3-6 各訂單之總完成時間

	發車時間	顧客 5	顧客 4	顧客 1
車輛 1	89	519	621	842
		顧客 2	顧客 3	
車輛 2	183	357	991	

表 3-7 總成本之計算

	顧客 1	顧客 2	顧客 3	顧客 4	顧客 5
$c_j$	842	357	991	621	519
$w_i$	3.21	1.49	1.07	5.59	6.58
	2702.82	531.93	1060.37	3471.39	3415.02
總成本	<b>11181.53</b>				

(2) 鄰近解定義：常用之鄰近解有兩種，分別為兩兩鄰近交換法、插入交換法，為了節省運算時間，本研究以兩兩交換法作為鄰近解之搜尋方法。兩兩鄰近交換法為利用任意相鄰兩個訂單，經由互相交換的過程，移步到新的解。以圖 3-1 和圖 3-2 為例，圖 3-1 表示一起始解， $S_i$  表示機台加工或車輛配送訂單順序。本次搜尋將兩相鄰之訂單 5(J5)和訂單 1(J1)互相交換位置，新的鄰近解即變成圖 3-2 所示，車輛 2 變成先配送訂單 1，最後再配送訂單 5。

a) 起始解：

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
機台 1	訂單 4   訂單 5						
機台 2	訂單 3	訂單 2	訂單 1				
車輛 1	訂單 3	訂單 2	訂單 4				
車輛 2	訂單 5	訂單 1					

圖 3-1 兩兩交換法之起始解示意圖

b) 鄰近解

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
機台 1	訂單 4   訂單 5						
機台 2	訂單 3	訂單 2	訂單 1				
車輛 1	訂單 3	訂單 2	訂單 4				
車輛 2	訂單 1	訂單 5					



圖 3-2 兩兩交換法之鄰近解示意圖

(3) 塔布表列長度：以 Glover[22,23]的建議設定為七，即是須經過七次的搜尋才能重複搜尋過去所搜尋過的解。

(4) 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) 塔布限制。
- (ii) 對應目標值優於目前所得之最佳解  $f(x)^*$ 。
- (iii) 塔布限制內最佳解。

(5) 停止準則：本研究預設當迭代次數達到五百次時即停止。

(6) 績效衡量指標 =  $\frac{\text{起始解} - \text{塔布搜尋之近似解}}{\text{起始解}} \times 100\%$

(7) 本研究之塔布搜尋法流程圖如圖 3-3 所示。



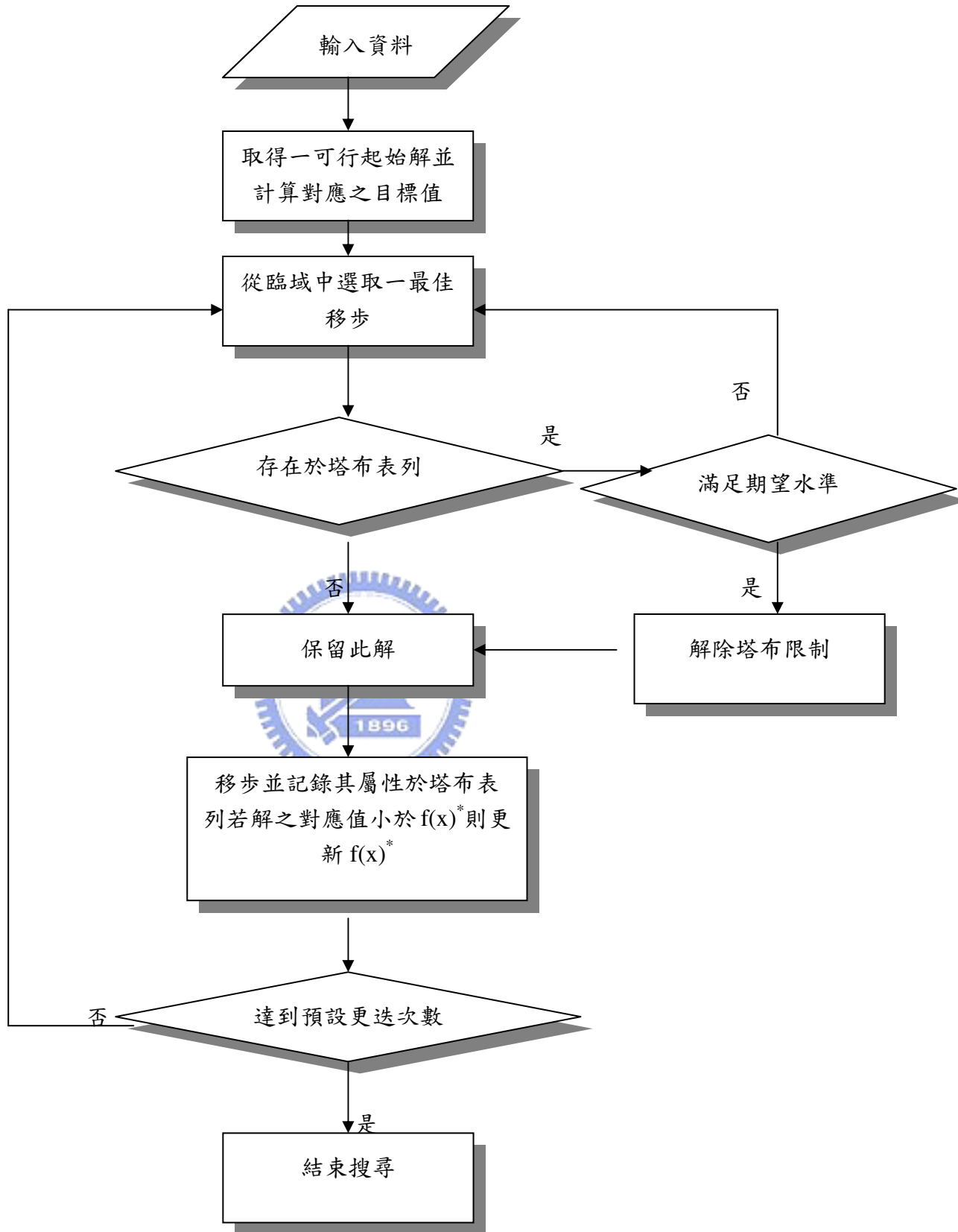


圖 3-3 塔布搜尋法流程圖

## 第四章 電腦模擬與測試結果

### 4.1 產生測試問題

本研究利用 C++ 語言撰寫程式，用以評估本研究塔布搜尋法之求解績效。由於本研究是探討區域性配送問題，因此選擇配送點分佈較為密集之題庫，以 Taillard 所提供之 tai385 檔案中的前 120 點來模擬車輛途程問題。且由於此時間矩陣之資料皆在 200 到 500 之間，因此產品製造加工時間則由電腦模擬產生出三組資料，皆服從均勻分配(uniform distribution)，分別為  $U(50,100)$ 、 $U(200, 500)$ 與  $U(800, 1000)$ 。此外也測試不同機台數與不同訂單數下的情境，其中機台數分別為 2、4 與 8，訂單數分別為 10、20、40、80 與 120。而車輛數隨著訂單數而增加，當訂單數為 10 時，車輛數為兩輛；訂單數為 20 與 40 時，車輛數為三輛；訂單數為 80 與 120 時，車輛數為四輛，以這些參數設定來測試塔布演算法的有效性與穩定性。本章之測試問題皆使用個人電腦在 CPU-Intel Pentium 4 3.00GHz 與 RAM-1G 的環境下進行。

### 4.2 模擬結果

本研究模擬測試問題可分為三個部份：(1)加工時間小於配送時間並服從  $U(50,100)$ ，以問題 D 表示；(2)加工時間約等於配送時間並服從  $U(200,500)$ ，以問題 B 表示；(3)加工時間大於配送時間並服從  $U(800,1000)$ ，以問題 P 表示。同樣的參數組合皆做三次重覆實驗並取其平均。

### 4.2.1 測試問題 D

經由參數變更設定之測試後，相關結果顯示如表 4-1。從表 4-1 觀察到當訂單數增加時，電腦運算時間急遽增加，以表 4-1 為例，當訂單數由 20 增加到 40 時，求解時間約為原來訂單數 20 的 10 倍；訂單數由 40 增加到 80 時求解時間為原來訂單數 40 的 10 倍。這與未定多項式難度問題的規模變大時，求解時間會呈現指數成長的特性相符，從圖 4-1 可更清楚看出此一求解時間急遽增加之特性。且本研究也測試過訂單數為 160 與 200 之問題，發現求解時間分別為 3 小時與 11 小時。考量到求解效率的因素，求解此類問題所花費的時間成本太大，故本研究之測試問題訂單數最大為 120。對於更大規模的問題時，本研究建議尋找其他較為快速的方法替代。除了訂單數外，從表 4-1 中也可觀察到當機台數增加時，電腦運算時間也會隨之增加，但增加幅度不若訂單數那麼大，大約呈現線性關係，如：機台數為 4 的運算時間約為機台數 2 的兩倍，機台數 8 的運算時間約為機台數 4 的兩倍，由此可知機台數對運算時間的影響不若訂單數大。

而在改善率部份，從圖 4-2 可觀察到訂單數越大時，其改善率的表現越佳。可能原因為訂單數較小時，解空間較小，所能改善的比率不大，而當訂單數漸增時，解空間越來越大，塔布演算法所能改善的空間也就變大許多。以表 4-1 為例，當訂單數為 10 與 20 時改善率大約只有百分之十，而當訂單數增加到 40 與 80 時，其改善率增加到百分之二十六以上，甚至最高接近百分之四十，就成效來說可說相當良好。但在訂單數為 120 時，改善率有下滑的趨勢，原因可能為解空間已經過大，搜尋次數可能不足，須再增加搜尋次數或採用其他更具效率的搜尋策略方可能使得改善率更進一步地提升。再由機台數觀察，可發現到當訂單數相同時，其改善率變動不會太大，這種現象說明了本研究之塔布演算法的穩定性良好，並不會因為機台數的不同而且產生改善率大幅波動的現象。



表 4-1 測試問題 D 之測試結果表

最佳解	訂單數=10	20	40	80	120
機台數=2	31722.88	86582.19	307057.31	1459690.88	3513643.25
4	N/A	69920.24	232804.34	1059734.00	2298402.92
8	N/A	N/A	211210.89	767060.31	1689519.87
改善率	10	20	40	80	120
2	10%	12%	27%	33%	32%
4	N/A	12%	34%	39%	34%
8	N/A	N/A	26%	42%	43%
運算時間(秒)	10	20	40	80	120
2	1.13	6.66	54.72	744.97	3428.38
4	N/A	9.75	90.11	1065.57	4316.18
8	N/A	N/A	197.82	2145.92	6353.30

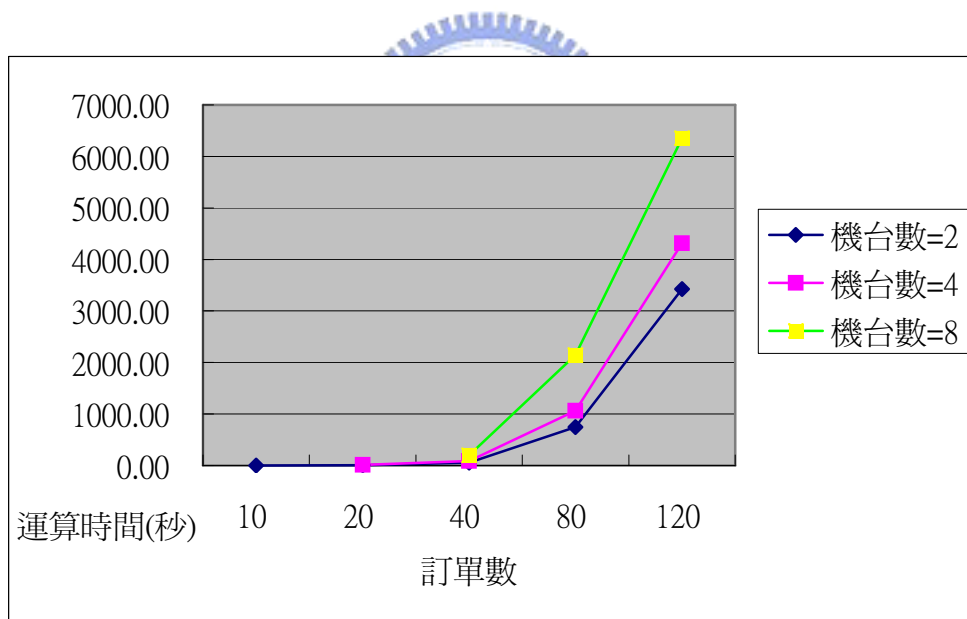


圖 4-1 訂單數與運算時間之關係圖(測試問題 D)

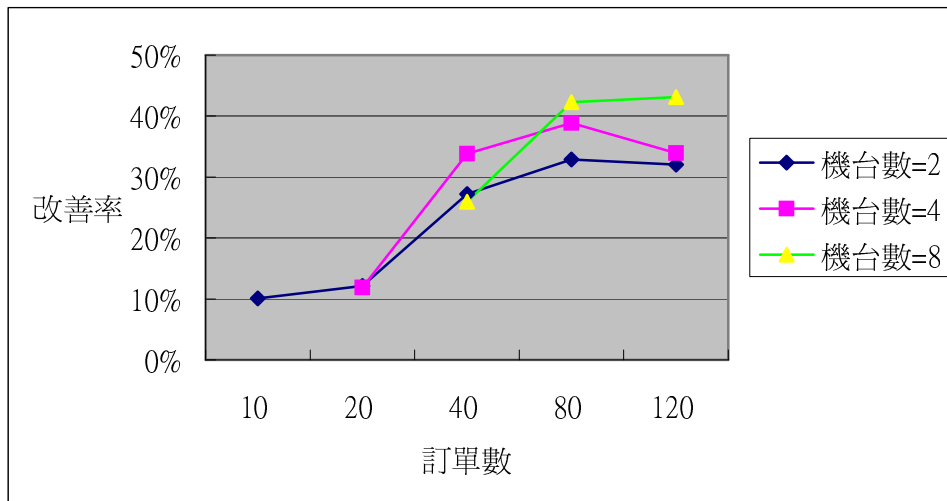


圖 4-2 訂單數與改善率之關係圖(測試問題 D)

#### 4.2.2 測試問題 B

第二部份為產品加工時間與配送時間皆處在相當的區域間，經由測試之後結果如表 4-2 所示，從表中可發現到主要特性與問題 D 類似。從圖 4-3 可更清楚看出當訂單數增加時，求解時間也會急遽增加，即訂單數變兩倍時，求解時間約為原來的十倍，同樣地訂單達到 160 與 200 時，求解時間就會變得相當費時，不符經濟原則。另外隨著訂單數增加，從圖 4-4 可看出改善率有顯著提升，從小訂單(訂單數為 10 與 20)的百分之十左右提昇到大訂單(訂單數為 40、80 與 120)的百分之三十左右。塔布演算法仍然能夠有效改善此兩階段問題之成本，此外演算法之穩定性在可由不同機台數的情境下，觀察得知改善率仍是相當穩定而不會大幅波動的現象，本研究之塔布演算法具有高度的穩定性。

表 4-2 測試問題 B 之測試結果表

最佳解	訂單數=10	20	40	80	120
機台數=2	69324.22	215805.77	864704.14	4638726.00	11034931.33
4	N/A	137390.36	563811.34	2668395.50	6113460.33
8	N/A	N/A	368612.04	1716420.83	4034696.75
改善率	10	20	40	80	120
2	10%	10%	15%	27%	26%
4	N/A	13%	22%	32%	28%
8	N/A	N/A	25%	36%	30%
運算時間(秒)	10	20	40	80	120
2	0.82	7.48	56.08	738.17	3418.94
4	N/A	9.46	87.30	1046.72	4258.08
8	N/A	N/A	194.12	2270.17	6224.74

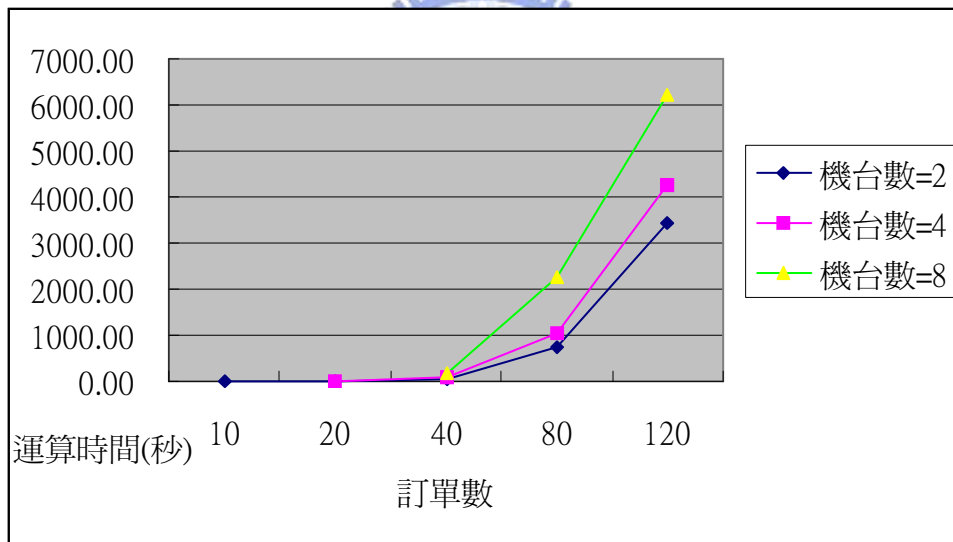


圖 4-3 訂單數與運算時間之關係圖(測試問題 B)

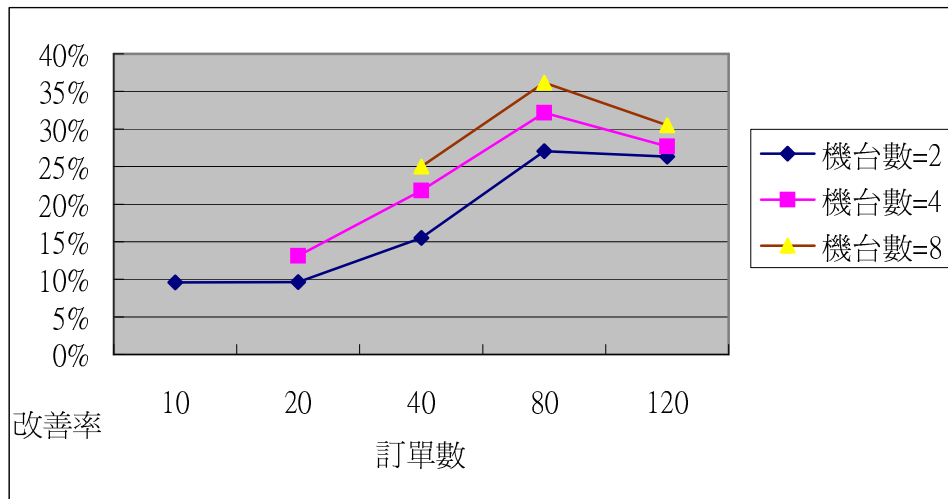


圖 4-4 訂單數與改善率之關係圖(測試問題 B)

### 4.2.3 測試問題 P

第三部份則是模擬產品製造加工時間大於配送時間時的情境，模擬測試後可從表 4-3 觀察到訂單數增加時，求解時間會大幅增加，從圖 4-5 可更清楚觀察到此現象，而在機台數增加時，求解時間呈現小幅增加，此現象與前兩個測試問題沒有太大的差異。而在改善率部份，較為不同的是在訂單數 40 時，前兩個測試問題改善率都約可達到百分三十，而本測試問題改善率只有百分之十左右，且在不同機台數下，改善率波動的起伏也過大，演算法呈現不穩定的現象。在其他部分，訂單數小 (訂單數為 10 與 20) 時改善率可達百分之十左右，而訂單數大 (訂單數為 80) 時可達百分之三十，雖在訂單數 120 時改善率略降，但改善率依舊有百分之二十三的水準，仍然具有高度有效性。

表 4-3 測試問題 P 之測試結果表

最佳解	訂單數=10	20	40	80	120
機台數=2	146013.26	476036.38	2024693.58	11108592.33	27343025.33
4	N/A	287108.78	1186850.87	6030547.67	14224423.00
8	N/A	N/A	727180.77	3466723.92	8035616.00
改善率	10	20	40	80	120
2	6%	15%	13%	24%	24%
4	N/A	14%	15%	28%	23%
8	N/A	N/A	6%	33%	25%
運算時間(秒)	10	20	40	80	120
2	1.03	6.70	54.68	738.31	3460.98
4	N/A	10.27	86.77	1053.71	4192.03
8	N/A	N/A	193.68	2306.47	5918.54

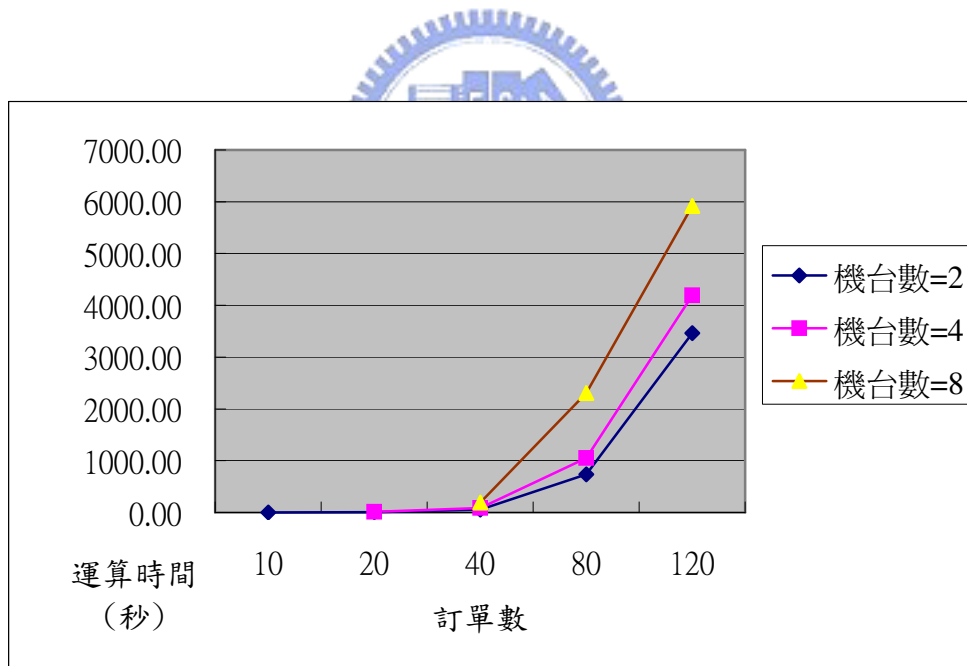


圖 4-5 訂單數與運算時間之關係圖(測試問題 P)

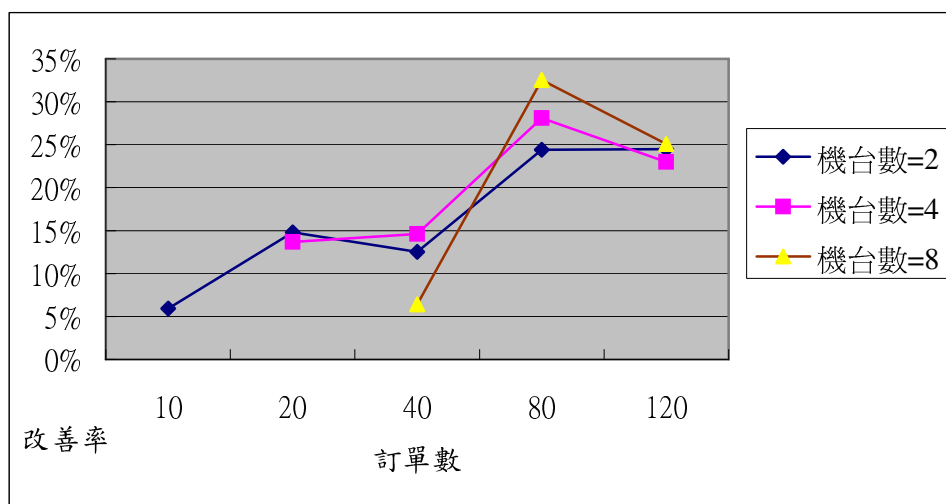


圖 4-6 訂單數與改善率之關係圖(測試問題 P)

#### 4.2.4 測試問題 D、B、P 之比較

比較上面三個測試問題之結果，可發現到不論加工時間為多少時，其求解時間皆不會起伏很大，除了說明本演算之穩定性外，也可知道加工時間的大小並不會影響到運算時間。而在改善率部份，三個測試問題皆會隨訂單數增加而提升，但如果將訂單數固定來看，其改善率並沒有太大的波動，訂單數小時(訂單數為 10 與 20)，改善率約為百分之十，而訂單數大時(訂單數為 40、80 與 120)改善率可提昇至約百分之三十。因此比較此三個測試問題之後，發現不論加工時間與配送時間之大小關係為何並不會影響到求解時間與改善率。

### 4.3 小結

經測試過此三個問題(D、B、P)之後，可針對以上結果歸納出幾項結論：

1. 本研究之塔布搜尋法在求解此排程與配送兩階段問題時，當訂單數增加，求解時間也會隨之快速增加，但改善率可獲得大幅度的提升，在訂單小時(訂單數為 10 與 20)改善率平均約百分之十二，訂單數大時(訂單

數為 40、80)改善率平均約可達到百分之三十，直到訂單數達到 120 時，改善率才略為下滑，整體平均改善率可達百分之二十，最高為百分之三十九，表示本研究之塔布演算法在求解此排程與配送兩階段問題具高度的有效性。

2. 本研究之塔布搜尋法除了在測試問題 P 中改善率呈現些許的不穩定之外，在其餘的測試問題中皆表現了極高的穩定性，此外不論機台數大小為何，只要訂單數相同其改善率皆呈現穩定的狀態，不會大幅波動，整體顯示本研究之塔布演算法具有高度的穩定性。
3. 在機台數增加時，其求解時間約為線性成長，機台數對於求解時間的影響不若訂單數那麼大，因此機台數的參數設定較不受限制，但由於決策者都會考量機台成本，依據訂單數大小來決定機台數的使用，因此本研究僅測試機台數為 2、4、8 時的情境供企業參考比較。
4. 經由上述三個測試問題之結果，可發現到不論產品加工時間與配送時間為何種比例關係，在訂單數不超過 120 的情形下，皆能在有效的時間內求得最適解，並且此最適解與初始解比較之，可降低成本約百分之二十。

## 第五章 結論與建議

### 5.1 結論

本研究提出一塔布演算法，用以求解非等效平行機台排程與車輛配送兩階段供應鏈整合問題，並以目前所能快速找到之合理初始解當作衡量指標，經模擬測試結果發現，可降低成本約百分之二十。主要貢獻可分為學術與實務兩方面。在學術方面：應用塔布搜尋法求解此兩階段問題，在訂單數在 120 以下時能在合理的時間內求得最適解，並經由模擬測試得知，本研究之塔布搜尋法的有效性與穩定性良好；在實務方面：現今產業競爭日趨激烈，由測試問題的模擬結果得知，使用本演算法可有效降低成本約百分之二十，可供企業應用改善工廠的產品製造與配送運作，進而降低製造與配送之總成本，讓企業獲得整體利益的提升，達到供應鏈上下游雙贏的結果。



### 5.2 未來研究方向

本論文之後續建議研究方向如下：

1. 可結合其他巨集式啟發演算法，擷取其長處，以期能更有效率求得更精確的最適解。
2. 可去除更多假設限制，如：車容量限制、時窗限制、加入整備時間或卸貨時間，希望能更符合真實情境。
3. 證明此兩階段問題的合理下界解，使之衡量演算法的績效時，能更具說服力。
4. 探討塔布搜尋法的參數設定，如：塔布表列之長度變化、鄰近解之定義或是衍生新的搜尋策略，使得塔布演算法的搜尋效果更具效率與精確性。



## 參考文獻

### [中文部分]

1. 林明俊，「隨機環境下多車種車派車問題之研究」，中原大學工業工程研究所碩士論文，民國八十七年。
2. 逢甲大學物流系統規劃實驗室，物流管理，  
<http://140.134.72.87/main/mbie/logistics.htm>，民國九十三年七月。
3. 游擱鴻，「以啟發式方法求解不相關平行機器排程問題」，朝陽大學碩士論文，民國八十六年。
4. 陳正雄、張百棧，「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，民國八十九年。
5. 劉素雲，「回應顧客需求之供應鏈整合模式之探討」，國立彰化師範大學商業教育學系，碩士論文，民國八十九年。
6. 韓復華、楊智凱、卓裕仁，「門檻接受法、噪音擾動法與搜尋空間平滑法在車輛路線問題之應用研究與相關比較」，運輸計畫季刊，第9卷，第三期，113-144，民國八十五年。

### [英文部分]

7. Baker, B.M. and M.A. Ayechev, "A genetic algorithm for the vehicle routing problem", Computers & Operations Research, vol.30, pp.787-800, 2003.
8. Battiti, R. and G. Tecchiolli, "The reactive tabu search", ORSA Journal on Computation, vol.6, pp.126-140, 1994.
9. Brandao, J., "A tabu search algorithm for the open vehicle routing problem", European Journal of Operational Research, vol.157, pp.552-564, 2004.
10. Breedam, A. V., "Improvement heuristics for the vehicle routing problems based on simulated annealing", European Journal of Operational Research,

vol.86, pp. 480-490, 1995.

11. Breedam, A.V., "Comparing descent heuristics and meta-heuristics for the vehicle routing problem", *Computers & Operations Research*, vol.28, pp.289-315, 2001.
12. Chang, Y. C. and C.Y. Lee, "Logistics scheduling: analysis of two-stage problems", *Journal of Systems Science and Systems Engineering*, vol.12, no 4, pp.385-407, 2003.
13. Chen, J.F.and T. H. Wu, "Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints", *Omega* 34, pp.81-89, 2006.
14. Chinyao, L., "Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines", *Computers and Operations Research*, vol.32, issue.8, pp.2013-2025, 2005.
15. Class, C.A., C.N. Potts and P. Shade, "Unrelated parallel machine scheduling using local search", *Mathematical and Computer Modelling* vol.24, issue: 9, November, pp.11-19, 1994.
16. Cordeau, J. F., G. Laporte and A. Mercier, "A unified tabu search heuristic for vehicle routing problem with time windows", *Journal of the operational research society*, vol.52, pp.928-936, 2001.
17. Ewa, F., "Preemptive scheduling with changeovers: using column generation technique and genetic algorithm", *Computers & Industrial Engineering* vol.37, issue 1-2, October, pp.81-84, 1999.
18. Fatima, G., "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: heuristic mixing method machines and precedence", *Computers & Industrial Engineering*, vol.37, issue 1-2, 1999.
19. Fermín, A.T.M. and R D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers & Operations Research*, vol.33, pp. 595-619, 2006.

20. Gendreau, M., A. Hertz and G. Laporte, " A tabu search heuristic for the vehicle routing problem", *Management Science*, vol.12, pp.76-90, 1994.
21. Glover, F., " Tabu search part I", *ORSA Journal On Computing*, vol.1, pp.190-206, 1989.
22. Glover, F., " Tabu search part II", *ORSA Journal On Computing*, vol.2, pp.4-32, 1990.
23. Hall, N.G. and C.N. Potts, " Supply chain scheduling : batching and delivery", *Operation Research*, vol.51, No.4, pp.566-584, 2003.
24. Hart, E, P.Ross and D.W.Corne, " Evolutionary scheduling: a review", *Genetic Programming and Evolvable Machines*, vol.6, pp.191-220, 2005.
25. Ho, S.C. and D.Haugland, " A tabu search heuristic for the vehicle routing problem with time windows and split deliveries", *Computers & Operations Research*, vol.31, pp.1947-1964, 2004.
26. Homberger, J. and H. Gehring, " A two-phase hybrid meta-heuristic for the vehicle routing problem with time windows", *European Journal of Operational Research*, vol.162, pp.220-238, 2005.
27. Hsieh, J. C., P. C. Chang and L. C. Hsu, " Scheduling of drilling operations in printed circuit board factory", *Computers & Industrial Engineering*, vol.44, pp.461-473, 2003.
28. Hwang, H. S., " An integrated distribution routing model in multi-supply center system", *International Journal of Production Economics*, vol.98, pp.136-142, 2005.
29. Kim, D. W., K. H. Kim, W. Jang and F. F. Chen, " Unrelated parallel machine scheduling with setup times using simulated annealing", *Robotics and Computer-Integrated Manufacturing*, vol.18, issue 3-4, June-August, pp.223-231, 2002.
30. Kim, D. W., D. G. Na and F.F. Chen, " Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective",

Robotics and Computer Integrated Manufacturing, vol.19, pp.173-181, 2003.

31. Lawler, E.L. and J.Labetoulle, " Preemptive scheduling of unrelated parallel processors by linear programming", Journal of the ACM, vol.25, pp.612-619, 1978.
32. Lee, C. Y. and Z. L. Chen, " Machine scheduling with transportation considerations", Journal of Scheduling, Vol.4, pp.3-24, 2001.
33. Lee, I., " Artificial intelligence search methods for multi-machine two-stage scheduling with due date penalty, inventory, and machining costs", Computers & Operations Research, vol.28, pp.835-852, 2001.
34. Lee, Y. H. and M. Pinedo, " Scheduling jobs on parallel machines with sequence-dependent setup time", European Journal of Operational Research, Vol.100, pp.464-474, 1997.
35. Li, C. L., G. Vairaktarakis and C. Y. Lee, " Machine scheduling with deliveries to multiple customer locations", European Journal of Operational Research, vol.164, pp.39-51, 2005.
36. Liu, F. H. and S. Y. Shen, " An overview of a heuristic for vehicle routing problem with time windows", Computers & Industrial Engineering, vol.37, pp.331-334, 1999.
37. Mazzeo, S. and I. Loiseau, " An ant colony algorithm for the capacitated vehicle routing", Electronic Notes in Discrete Mathematics, vol.18, pp.181-186, 2004.
38. Nowicki, E. and C. Smutnicki, " The flow shop with parallel machines : a tabu search approach", European Journal of Operation Research, vol.106, pp.226-253, 1998.
39. Oppen, J. and A.Lzkketangen, " Arc routing in a node routing environment", Computers & Operations Research, vol.33, pp.1033-1055, 2006.
40. Osman, I. H., " Meta-strategy simulated annealing and tabu search

algorithms for the vehicle routing problem”, *Annals of Operations Research*, vol 41, pp.421-451, 1993.

41. Park, M.W. and Y. D. Kim, ” Search heuristic for a parallel machine scheduling problem with ready times and due dates”, *Computers Industrial Engineering*, vol.33, no.3-4, pp.793-796, 1997.
42. Park, Y., S.Y. Kim and Y. H. Lee., ” Scheduling jobs on parallel machines applying neural network and heuristic rules”, *Computers and Industrial Engineering*, vol.38, pp.189-202, 2000.
43. Perl, J., ” A unified warehouse location-routing analysis“, *Transportation Research*, 19B, pp.381-396, 1983.
44. Piersam, N. and W. V. Dijk, ” A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search”, *Math Computer Modeling*, vol.24, no.9, pp11-19, 1996.
45. Pinedo, M., “ Scheduling: theory, algorithms, and systems”, *Prentice Hall international series in industrial and systems engineering*, New Jersey, 1995.
46. Reeves, C.R., ” Improving the efficiency of tabu search for machine sequencing problems”, *Journal of the Operational Research Society*, 44(4), pp.375-383, 1993.
47. Renaud, J., G. Laporte and F. F. Boctor, ” A tabu search heuristics for the multi-depot vehicle routing problems”, *Computer & Operations Research*, vol.23, no.3, pp.229-235, 1996.
48. Ruiz, R. and C. Maroto, ” A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility”, *European Journal of Operational Research*, vol.169, pp.781-800, 2006.
49. Savelsbergh, M., ” Local search in routing problems with times”, *Annals of Operations research*, vol.4, pp.285-305, 1985.
50. Stephan, S., ” A tabu search heuristic for the truck and trailer routing

- problem”, *Computers & Operations Research*, vol.33, pp.894-909, 2006.
51. Suresh, V. and D. Chaudhuri, ” Bicriteria scheduling problem for unrelated parallel machines”, *Computers & Industrial Engineering*, vol.30, no.1, pp.77-82, 1996.
52. Tan, K.C., L. H. Lee, Q. L. Zhu and K. ou, ” Heuristic methods for vehicle routing problem with time windows”, *Artificial Intelligence in Engineering* vol.15, pp.281-295, 2001.
53. Tarantilis, C.D., ” Solving the vehicle routing problem with adaptive memory programming methodology”, *Computers & Operations Research*, vol.32, pp. 2309-2327, 2005.
54. Teodorovic, D. and G. Pavkovic., ” The fuzzy set theory approach to vehicle routing problem when demand at nodes is uncertain”, *Fuzzy sets and systems*, vol.82, pp.307-317, 1995.
55. Thangiah, S.R., I.H. Osman, and T. Sun, “ Hybird genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows“, *Technical Report UKC/OR94/4*, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK, 1994.
56. Walker, R., P. Forster and S. Banthorpe, ” Content production and delivery for interactive multimedia services a new approach”, *BT Technol Journal*, vol.15, No.2, 1997.
57. Wu, T.H., L. Chinyao and J. W. Bai, ” Heuristic solutions to multi-depot location-routing problems”, *Computers & Operations Research*, vol.29, pp.1393-1415, 2002.

## 附錄一 配送時間

資料來源：<http://neo.lcc.uma.es/radi-aeb/WebVRP/>

訂單編號	座標 X	座標 Y		訂單編號	座標 X	座標 Y
1	5166	1480		61	5254	1761
2	5403	1631		62	5281	1778
3	5030	1372		63	5278	1783
4	5469	1830		64	5301	1784
5	5495	1832		65	5306	1792
6	5394	1844		66	5333	1799
7	5403	1868		67	5342	1796
8	5415	1875		68	5337	1816
9	5426	1876		69	5288	1762
10	5440	1885		70	5306	1748
11	5451	1891		71	5291	1746
12	5457	1926		72	5282	1730
13	5365	1877		73	5264	1739
14	5350	1843		74	5264	1718
15	5322	1844		75	5253	1718
16	5342	1861		76	5240	1730
17	5325	1869		77	5242	1716
18	5302	1823		78	5198	1714
19	5284	1862		79	5225	1689
20	5382	1890		80	5245	1668
21	5385	1886		81	5263	1661
22	5376	1887		82	5259	1632
23	5372	1896		83	5242	1624
24	5374	1873		84	5252	1605
25	5377	1863		85	5245	1583
26	5372	1845		86	5265	1583
27	5365	1832		87	5287	1595
28	5365	1822		88	5271	1614
29	5355	1865		89	5286	1629
30	5401	1893		90	5277	1651
31	5456	1933		91	5310	1588

32	5614	1855		92	5319	1588
33	5636	1867		93	5330	1605
34	5610	1913		94	5360	1595
35	5592	1934		95	5343	1616
36	5644	1923		96	5329	1617
37	5694	1892		97	5355	1636
38	5709	1917		98	5340	1644
39	5697	1919		99	5310	1612
40	5725	1953		100	5302	1629
41	5657	1948		101	5317	1636
42	5673	1965		102	5301	1672
43	5683	1971		103	5270	1678
44	5674	1973		104	5291	1678
45	5712	1992		105	5289	1687
46	5651	1974		106	5296	1718
47	5663	1989		107	5331	1709
48	5680	2006		108	5334	1727
49	5114	1669		109	5357	1749
50	5143	1670		110	5344	1755
51	5176	1590		111	5365	1771
52	5222	1620		112	5379	1773
53	5224	1592		113	5387	1778
54	5232	1587		114	5401	1775
55	5214	1635		115	5411	1790
56	5234	1641		116	5426	1801
57	5211	1666		117	5433	1813
58	5190	1739		118	5391	1811
59	5217	1759		119	5365	1806
60	5249	1770		120	5362	1811