

國立交通大學

工業工程與管理學系

碩士論文

應用蟻群演算法於求解供應鏈中整合產品製造
與成品配送兩階段問題

Applying Ant Colony Optimization to Solve Two-Stage
Integrated Production and Distribution Problem in
Supply Chains



研究生：王天威

指導教授：張永佳 博士

中華民國九十六年七月

應用蟻群演算法於求解供應鏈中整合產品製造與成品配送

兩階段問題

學生：王天威

指導教授：張永佳 博士

國立交通大學工業工程與管理學系碩士班

摘要

產品製造與成品配送是供應鏈中相當重要的兩個階段，由於顧客需求的快速變動以及產品的高度客製化，現今許多企業採用接單式生產(make to order)或直接銷售(direct order)模式而被迫減少存貨，但同時仍必須快速回應顧客的要求以維持其競爭力。此種現象增加了供應鏈中產品製造與成品配送作業的互動，並提昇了整合這兩階段研究的實用性。

本研究探討整合供應鏈中產品製造與成品配送兩階段問題，在這種問題中，訂單會先在一組非等效平行機台上(unrelated parallel machine)進行加工，加工完畢後則由具容量限制的車輛配送至顧客的所在地。訂單的完工時間定義為送達顧客的時間，而問題的目標則是求出一個整合製造與配送的排程，以最小化加權後的總完工時間。

然而整合製造與配送兩階段問題的複雜度為未定多項式難度(NP-hard)，為了在合理的時間內求得品質良好的近似最佳解，本研究採用蟻群演算法(ant colony optimization)來求解此類問題，並透過電腦實驗來測試演算法的求解品質及穩健性，期望本研究結果能有助於提昇整合製造與配送兩階段問題的實用性及應用價值。

關鍵字：非等效平行機台排程問題、車輛途程問題、蟻群演算法

Applying Ant Colony Optimization to Solve Two-Stage Integrated Production and Distribution Problem in Supply Chains

Department of Industrial Engineering and Management

National Chiao Tung University

Student : Yung-Chia Chang

Advisor : Yung-Chia Chang

Abstracts

Due to the rapid-changing market demand and highly customized product requirements, many enterprises have adopted make-to-order or direct-order business model. In this type of business model, enterprises are forced to lower the amount of inventory needed across their supply chain but still have to be more responsive to customers' requirements. Reduced inventory creates a closer interaction between production and distribution activities and thus increases the practical usefulness of integrated models.

We consider an integrated production and distribution problem at the individual job level in this study. In this problem, jobs are first processed by one of a set of unrelated parallel machines and then distributed by vehicles with limited capacity to the corresponding customer locations. The completion time of a job is defined as the time when it is delivered to its customer. The objective is to find a joint production and distribution schedule so that the total weighted completion time is minimized. The complexity of the studied problem is NP-hard. Therefore, we used an ant colony algorithm to solve this problem in order to find near-optimal solutions in reasonable computation times. Computational analysis is performed to evaluate the effectiveness and stability of the proposed approach. We expect our research results to help make the study of integrated production and distribution problems more practical and with better application values.

Key words: production and distribution, scheduling, ant colony optimization

誌謝

碩士生涯即將劃上句點，感覺兩年的時光一轉眼就過去了，在交大工工所的這兩年內，除了學科的專業知識及論文寫作的邏輯思考外，我還學到了不少待人處事的道理，使我的思想與行為更加成熟，因此這兩年可以說是我生命中的蛻變點之一。在未來的日子裡，我會永遠懷念我在交大的七百多個日子，以及陪我一起打拼的老師及夥伴們。

能夠順利取得碩士學位，首先要感謝我的父母及我的哥哥，他們讓我能夠衣食無缺地求學。母親總是時時刻刻地關心我的生活，父親常給我未來生涯規劃的建議並在背後給我鼓勵及支持，而哥哥則是與我分享心事的好夥伴。

在學校方面，我能受到張永佳教授的指導可以說是非常幸運的，老師採用啟發式的教導方式，鼓勵我們獨立思考出論文的脈絡，並在我們寫論文的同時，不厭其煩地幫我們修改一開始殘不忍睹的文章。除了論文之外，老師也提供我們很好的研究環境，讓我們能夠舒適地閱讀及進行論文寫作。另外我也要感謝擔任口試委員的唐麗英教授及洪一薰教授：唐老師根據她豐富的統計經驗，提供我論文後段分析的相關建議，而洪老師精準地幫我挑出論文中的一些缺陷，這兩位老師寶貴的意見著實使我的論文更趨於完善。

最後我要感謝的就是研究室的各位夥伴們，博士班的惠誠學長、93級的大師兄一昱皓學長、佩君學姐以及其他研究室的學長姐們對我們都非常照顧，引領我們進入研究生的生活，提供我們課業及生活上的意見，同時也是我們一起玩樂的同伴。和我同屆的阿嚕咪、小舜子與藍肥更是與我同甘共苦的好兄弟，陪我一起作研究，一起玩樂，一起練功，一起挨罵。96級的學弟妹們：薛老大、凱凱、佳儒及佩均則是我們研究室的開心果，互相打鬧的生活讓研究室增添了許多歡笑。雖然和大家相處的時間只有一、兩年，但是這份情誼已如兄弟姐妹那樣的深厚，沒有大家的扶持我無法順利地完成我的碩士學位。最後再一次由衷地感謝我的家人、指導我的諸位老師們以及研究室的夥伴們，謝謝你們！

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 研究背景與動機.....	1
1.2 研究目的	3
1.3 研究方法與流程.....	4
1.4 研究架構	4
第二章 文獻探討.....	6
2.1 非等效平行機台與車輛途程問題.....	6
2.1.1 非等效平行機台問題之相關文獻.....	6
2.1.2 車輛途程問題之相關文獻	8
2.2 蟻群演算法	13
2.2.1 蟻群演算法的理論.....	13
2.2.2 蟻群演算法的演算步驟	14
2.2.3 蟻群演算法的相關應用.....	17
2.2.3.1 蟻群演算法應用於排程問題	17
2.3 供應鏈整合之相關文獻	18
第三章 問題定義與研究方法	24
3.1 問題定義	24
3.2 演算法之設計.....	26
3.2.1 非等效平行機台問題之轉換	26
3.2.2 狀態轉移法則.....	27
3.2.3 費洛蒙更新法則.....	29
3.2.4 蟻群演算法架構.....	31
3.2.5 演算步驟範例說明	34
第四章 演算法測試與結果分析	41

4.1 參數設計	41
4.1.1 訂單權重參數(γ)	42
4.1.2 螞蟻數量(ANT).....	42
4.1.3 費洛蒙濃度初始值(τ_0)	43
4.1.4 費洛蒙濃度與能見度參數(α, β).....	44
4.1.5 參數 q_0	44
4.1.6 區域費洛蒙殘留係數(ρ).....	45
4.1.7 全域費洛蒙殘留係數 ρ'	46
4.1.8 迭代數(TN).....	46
4.1.9 小結	47
4.2 ACO_PDP 之求解品質測試—以特殊問題為例	49
4.3 測試問題設計.....	50
4.4 測試結果分析.....	53
4.4.2 機台數與求解品質之關係	55
4.4.3 訂單數與求解品質之關係	55
4.4.4 車輛容量限制與求解品質之關係.....	55
4.4.5 小結	56
第五章 結論與建議.....	57
5.1 結論	57
5.2 未來研究方向之建議.....	57
5.2.1 整合產品製造與成品配送兩階段問題方面	57
5.2.2 演算法方面	58
參考文獻	59

圖目錄

圖 1 研究流程圖	5
圖 2 真實螞蟻搜尋路徑的概念圖	14
圖 3 演算法架構流程圖	32
圖 4 排程費洛蒙矩陣區域更新	38
圖 5 排程甘特圖	38
圖 6 路徑費洛蒙矩陣區域更新	40
圖 7 訂單權重參數測試圖	42
圖 8 螞蟻數量測試圖	43
圖 9 費洛蒙濃度初始值測試圖	43
圖 10 費洛蒙濃度與能見度參數測試圖	44
圖 11 參數 q_0 測試圖	45
圖 12 區域費洛蒙殘留係數 ρ 測試圖	45
圖 13 全域費洛蒙殘留係數測試圖	46
圖 14 $n = 80$ 迭代數測試圖	47
圖 15 $n = 200$ 迭代數測試圖	48

表目錄

表 1 整合製造—配送兩階段問題相關文獻之整理	22
表 2 作業編號表	27
表 3 訂單之權重與加工時間表	34
表 4 訂單之配送時間矩陣	35
表 5 排程部分能見度資料表	36
表 6 排程部分狀態轉移機率表	37
表 7 配送部份能見度資料表	39
表 8 配送狀態轉移機率表	39
表 9 ACO_PDP 之求解品質測試表	49
表 10 測試問題設計表 (呂學君, 2007)	52
表 11 ACO_PDP、Tabu_PDP 求解能力比較表	54
表 12 相對誤差複回歸分析表	55



第一章 緒論

1.1 研究背景與動機

供應鏈管理是近年來全球最重要的議題之一。由於資訊科技發達所帶動的電子商務潮流，使得公司間的資訊能夠整合，大部分的企業便開始將重心從原本的內部管理，轉移到與上下游之間的供應鏈管理。完整的供應鏈為一個傳遞產品與服務的網路，包含了最上游之原物料商的供應、製造商的加工、經銷商及物流商的成品配送，一直到最終顧客的服務，而這整個流程必須藉由設計良好的資訊流、金流及物流來加以驅動才能有效地運作。供應鏈管理的主要目標是有效運用資源來滿足顧客的需求，並將整體的利潤最大化，但通常供應鏈中各個階段所追求的目標不同，所重視的績效指標也有所衝突。此外，在各階段只追求局部最佳化的情況下，供應鏈中的某一階段可能會為了提高獲利而將成本或風險轉嫁給其他的階段，而這種現象會導致整個供應鏈體系無法得到最大的利益，因此如何有效整合供應鏈是一個非常值得深入探討的主題。

以往供應鏈中的各個階段間通常都會有存貨來做為顧客需求的緩衝，但是不論是原物料、半成品或成品的存貨成本對廠商來說都是相當龐大的資金積壓，其中又以產品製造與成品配送階段間成品存貨成本所佔的比例最高；再者，現今顧客的需求趨向多樣化且變動迅速，今日的存貨未必能滿足明日顧客的需求，因此有越來越多企業在產品製造與成品配送這兩個階段選擇採用接單式生產(make-to-order)與直接銷售(direct-order)的商業模式。在一個接單式生產的環境中，客製化的產品在加工完畢後，不經由任何的庫存直接送達顧客，而且只需要很短的前置時間。以客製化的電腦產業為例，顧客可以任意選擇他們偏好的電腦零組件來組成一台電腦，因此生產的電腦的工廠必須能夠製造出數百種甚至數千種規格的電腦，在這種情況下，在顧客下單前便將所有規格的電腦都組裝且包裝好作為庫存

的作法，就顯得非常不合理且不符合經濟效益。因此當企業將商業模式轉向接單式生產與直接銷售的同時，供應鏈中的產品製造與成品配送兩個階段也因為中介存貨的消失，而更加緊密的鏈結在一起。

傳統的方法在處理產品製造與成品配送的問題時，往往都是分開求解各自的最佳排程及配送路徑，但如前所述，不同的階段的績效指標可能會有所衝突，一個階段單獨求得的最佳解可能對另一個階段產生負面的影響，因而無法達到整體系統的最佳化。舉例來說，假設一間公司在生產各種不同的產品後，利用車輛將成品配送到數個不同的顧客位置，在做成品配送的路徑決策時，通常都會考慮把顧客所在位置相近的訂單分成同一個批次，這種做法雖然可以減少車輛行走距離，降低運送成本，但是對於生產排程而言，這同一批次內的訂單在機台上加工時可能要經過相當繁複的整備工作，而造成過高的生產成本。當產品製造與成品配送的直接串連成為一種新趨勢後，整合這兩個階段來求得整體最佳化更顯得重要且需要更深入的研究。

本研究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(以下簡稱為製造—配送兩階段問題)，其中第一階段考慮產品製造的排程問題，然而實際上在進行加工生產時，即使具有相同功能的機台間也會有效率上的差異，因此為合理地模擬實際生產排程的情況，本研究選定以非等效平行機台(unrelated parallel machine)的排程問題來代表第一階段的生產環境。而第二階段考慮成品配送的問題，其中假設成品是經由多部具有車容量限制的車輛來配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)。而本研究所採用的績效衡量指標為接獲訂單至顧客實際收到成品所花費的時間，乘上訂單的權重轉換而成的總成本。

非等效平行機台排程問題與車輛途程問題皆已經被證實為未定多項式難度(non-deterministic polynomial-time hard, NP-Hard)的問題(Garey and

Johnson, 1979; Lenstra and Rinnooy Kan, 1981), 因此整合後的製造—配送兩階段問題的問題複雜度也至少為 NP-Hard。目前雖然已有不少整合供應鏈中製造與配送兩階段的相關文獻，但大部分研究都是針對策略(strategic)層面，例如 Sarmiento and Nagi (1999)回顧了數種整合產品製造與成品配送的系統模型，並探討整合後可能為企業帶來的利益，但是這些模型皆沒有討論到更細部的排程、配送等作業(operational)層面問題或求解方法。而少數探討作業層面的文獻主要只是透過數學方法來釐清兩階段問題在不同情況與目標函數下的問題複雜度，或利用動態規劃(dynamic programming)與啟發式演算法來求解特殊個案或簡化過的小規模問題（如 Lee and Chen 2001， Hall and Potts 2002， Chang and Lee 2003， Li *et al.* 2003， Chang and Lee 2004 等），相對地較少有研究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。相較於傳統的啟發式演算法或數學方法而言，巨集式啟發式演算法雖然無法準確求得最佳解，但是可以在合理的時間內提供品質佳的近似最佳解，所以非常適合應用於求解此類複雜度高的問題。

蟻群演算法(ant colony optimization)發展至今約有十多年的歷史 (Dorigo, 1992)，經過不斷的改良已成為一套相當具有效率的演算法，並且被成功的應用在許多問題上，如旅行銷售員問題(travel salesman problem)、排程問題(scheduling problem)及車輛途程問題等。蟻群演算法的主要優點是能透過費洛蒙濃度的機制快速地求出近似最佳解，同時也可以避免搜尋過程落入區域最佳解中。本研究以蟻群演算法為基礎，發展一套能應用於求解整合供應鏈中製造—配送兩階段問題的演算法。

1.2 研究目的

本研究探討整合供應鏈中製造—配送兩階段的問題，其中第一階段為非等效平行機台的排程問題，第二階段則為具有容量限制的車輛途程問

題。本研究的主要目的是以蟻群演算法為基礎，發展出一套能兼具求解速度與近似解品質的演算法，並針對各種規模的問題測試演算法的有效性，期望能在合理的時間內求得高品質的近似解。

1.3 研究方法與流程

本研究應用蟻群演算法於供應鏈中整合生產排程與成品配送兩階段問題，其研究方法流程如下：

1. 確定研究問題與問題定義：確定本研究之背景、動機及研究目的，並定義基本的研究假設與問題情境。
2. 設計演算法：以蟻群演算法為基本架構，發展一套應用於本研究問題的演算法，並以 C++ 語言進行演算法程式的撰寫，以求解整合製造—配送兩階段問題。
3. 演算法測試與分析：探討在各種規模之問題與不同的演算法參數設定下，蟻群演算法求解的效率與近似最佳解的品質，並將結果與其他相關研究比較。
4. 結論與建議：根據電腦測試的結果作結論，並建議未來後續的研究方向。而本研究的流程如圖 1 所示。

1.4 研究架構

本研究論文主要架構由五個章節所組成，茲列述如下：

1. 緒論
包含研究動機、目的與問題的背景，以及本研究的架構與研究方法。
2. 文獻探討
整理非等效平行機台、車輛途程問題及整合製造與配送兩階段問題的相關文獻，並概述蟻群演算法的原理及其相關應用。
3. 研究方法
詳細定義本研究所探討的問題模型與基本假設，並以蟻群演法為基本架

構，設計一個可應用在整合兩階段問題的演算法。

4. 演算法測試及結果分析

利用本研究設計的 33 個例題來測試演算法的求解品質及效率，並對測試的結果作相關的分析。

5. 結論與建議

針對測試的結果做出結論，並對後續的相關研究提出數點建議。

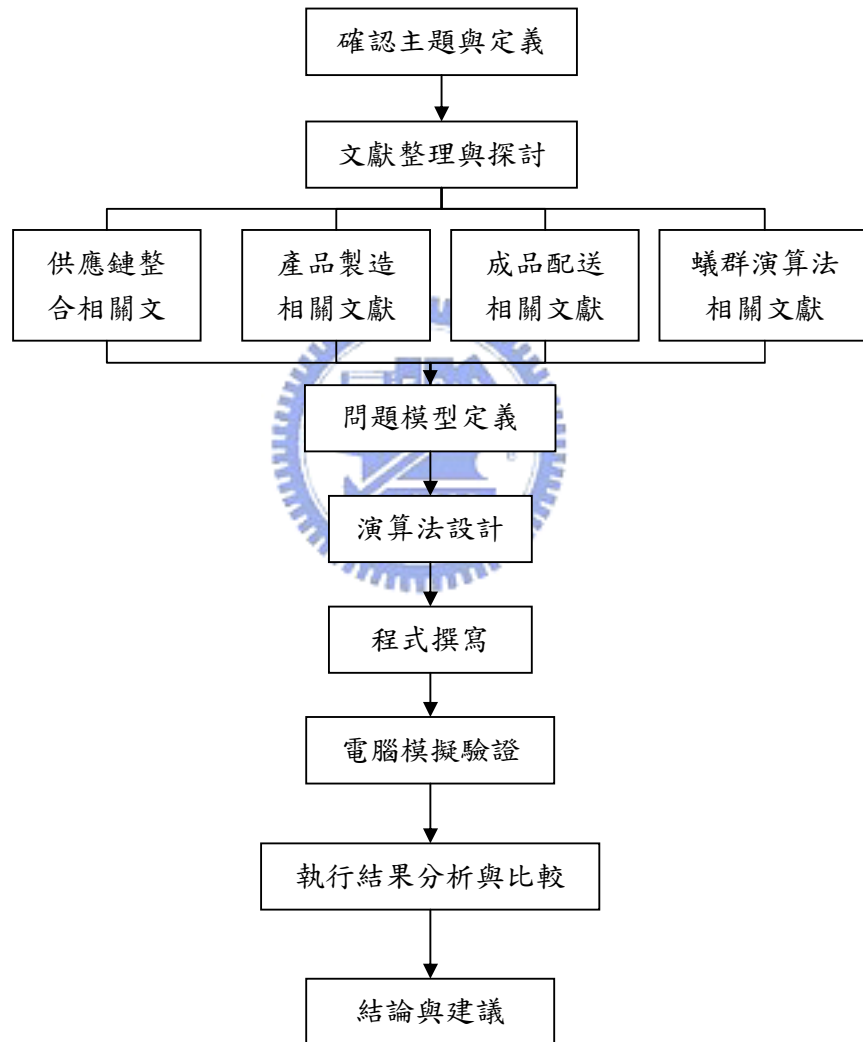


圖 1 研究流程圖

第二章 文獻探討

本章共分為三個小節，其中 2.1 節探討非等效平行機台與車輛途程問題的相關文獻；2.2 節簡述蟻群演算法的原理與演算步驟，並探討相關應用的文獻；2.3 節則是探討供應鏈整合之相關文獻。

2.1 非等效平行機台與車輛途程問題

本研究以非等效平行機台問題模擬第一階段的產品製造問題，以下於 2.1.1 節回顧其相關文獻。而第二階段的成品配送問題，本研究以車輛途程問題模擬之，並於 2.1.2 節回顧其相關文獻。

2.1.1 非等效平行機台問題之相關文獻

在平行機台的生產環境中，每一台機台都有相似的功能，意即待處理之工件皆可在任何一台機台上被加工。而平行機台又可以依機台間的關係分為以下三種類型(Allahverdi, 1994)：

1. 完全一致平行機台(identical parallel machines)：

所有的機台皆具有相同的功能及處理速度，一個工件在任何機台上的加工時間皆完全相同。

2. 等效平行機台(uniform parallel machines)：

不同的平行機台間因為新舊程度的不同或其他因素，造成處理速度上的差異，使得處理同一個工件的加工時間在不同機台間會呈比例關係。

3. 非等效平行機台(unrelated parallel machines)：

不同的機台擁有相同的功能，但是同一個工件在任何機台上的處理時間皆不相同，且之間無任何比例關係。

本研究的第一階段將針對非等效平行機台來做探討，與其他兩種類型相比，針對非等效平行機台的研究顯得較少(Dhaenens-Flipo, 2001)，而現有的相關文獻大部份都針對最大完工時間(makespan)、總完工時間(total

processing time)以及延遲時間(tardiness)三個衡量準則來討論。Bruno *et al.* (1974) 與 Horn (1973)研究非等效平行機台的問題，並以最小化總完工時間為目標。Azizoglu and Kirca (1999)也研究類似的問題，不同的是他們考慮了加入權重後的總完工時間。Hariri and Potts (1991)發展了一套兩階段的啟發式演算法，以最小化最大完工時間為目標，他們在第一階段應用線性規劃來產生前半部的排程，接著在第二階段利用了最早完工時間法(shortest processing time, SPT)來安排剩下的工件。Suresh and Dipak (1994)應用 GAP-EDD 演算法，以最小化最大延遲時間為目標來探討非等效平行機台問題。Weng *et al.* (2001)以最小化加權後平均完工時間為目標，來探討非等效平行機台問題，同時考慮了與機台獨立但與加工順序相依的整備時間，並測試了 7 個不同的啟發式演算法，其中可以分為兩種策略，第一種策略是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。

近幾年來，被發展用來解決複雜組合型問題的巨集式啟發式演算法，也被應用在排程的領域，其中亦不乏非等效平行機台的排程問題（巨集式啟發式演算法將在下一小節做更進一步介紹）。Kim *et al.* (2002)以模擬退火(simulated annealing)為基礎發展了數個啟發式演算法，來求解包含相依整備時間與批量生產的非等效平行機台排程問題，其中他們以最小化最大延遲時間為目標，探討 6 種不同的鄰近解產生方法及相關參數，之後並將其效能與鄰近搜尋法(neighborhood search)做比較。Glass *et al.* (1994)比較了基因演算法(genetic algorithms)、模擬退火法及塔布搜尋法(tabu search)等，應用在最小化最大完工時間之非等效平行機台問題上的求解效能，但不考慮整備時間。在結論中作者提到，根據電腦實驗的結果，基因演算法所得解的品質較差，然而結合其他演算法的混合式基因演算法所得到的解之品質，則與塔布搜尋法和模擬退火法相當。Lee *et al.* (1997)將模擬退火法與 ATCS (apparent tardiness cost with setup)法則結合，以最小化加權總延遲時

間為目標，來求解具有相依整備時間的平行機台問題。Srivastava (1997)也針對不考慮整備時間的問題，提出一個有效率的塔布搜尋法，並說明這個方法可以在合理的時間內，為具實務規模的問題提供高品質的解。陳民俊(2004)利用蟻群演算法為基礎建構兩種啟發式演算法，在最小化總流程時間的目標下求解非等效平行機台問題，並與塔布搜尋法及模擬退火法做比較，結果顯示其演算法在適合的參數設定下具有較好的求解效能。

2.1.2 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，因此成品配送中的車輛途程問題(vehicle routing problems, VRP)在數十年來備受學者及企業的重視，是一個非常值得探討的議題，至今仍有不少新的理論或求解方法相繼被發表出來。VRP 最早是由 Dantzig and Ramser 在 1959 年提出，其基本定義為一組車隊從單一物流中心，將貨品運送到已知的顧客所在位置，每個顧客的需求量皆為已知且都會被服務一次，每部車輛所承載之貨物不得超過其容量限制，車輛在服務顧客完畢後返回物流中心，而問題的目標便是最佳化車輛所行走的路徑，使總成本最小。

除了早期的 VRP 外，有部份學者把物流業實務上所必須考慮的時窗限制，加入車輛途程問題而衍伸出時窗性車輛途程問題(vehicle routing problems with time window, VRPTW)，其強調時效的特性更符合實際情況，也常常是物流業者所需面對的決策問題，因此廣受各界深入討論。

VRP 與前一節介紹的非等效平效平行機台同屬於 NP-Hard 的問題 (Garey and Johnson, 1979)，早期是以一些較簡單的最適解法來求解，但隨著客戶數量及其需求日漸增加，企業所面臨的 VRP 規模也趨於複雜且龐大，傳統的啟發式演算法及其他解法已不敷使用，無法在合理的時間內求得最適解。為了因應問題規模的擴大，40 多年來有許多學者及專家致力於發展出新的求解方法。求解 VRP 的方法主要可以分為以下三類：

1. 最適解法(exact procedures)

此類方法包括整數規劃法(integer programming)、分枝界限法(branch and bound)、動態規劃(dynamic programming)等。Dantzig and Ramser (1959)利用整數規劃模式來解決問題(陳昱皓, 2006)。Eilon (1971)提出 VRP 的一般化動態規劃模式, 其主要的原理是將一個問題分成若干個小問題, 再加以解決。Kolen *et al.* (1987)發表了應用在 VRP 上的分枝界限法, 先利用動態規劃計算出新節點之下界, 然後運用洞悉步驟來捨去那些超過已知可行解目標值界限的節點, 這個方法可以成功求解包含 15 個顧客的問題, 但不適合求解規模太大的問題。

2. 傳統啟發式演算法(classical heuristic method)

VRP 屬於 NP-hard 的問題, 當問題規模較小時, 若以最適解法或其他方法求解可以得到高品質的解, 然而當問題規模很大時, 這些方法的求解時間將隨著節點的增加成指數性的成長, 因此要在有限的時間內找出最佳解便會變得相當困難。而啟發式演算法, 可以節省了需多繁雜的搜尋程序, 並可在限制的時間內找出近似的可行解來取代最佳解, 對於求解大規模問題而言顯得更有效率。

應用在 VRP 的傳統啟發式解法包括了節省法(savings algorithm)、掃描法(sweep algorithm)、先分群再排路線(cluster -first- route -second)以及插入法(insertion heuristic)等。Clarke and Wright (1964)首先提出節省法來求解 VRP。根據資料顯示, 掃描法最早可以回溯 Wren (1971)的著作中(羅敏華, 2003), 而 Wren and Holliday (1972)應用此法求解具有容量限制的 VRP(capacitated vehicle routing problem), 爾後 Gillett and Miller (1974)提出兩階段啟發式的掃描演算法才使得掃描法較為普遍。插入法最早由 Mole and Jameson (1976)所提出, Christofides *et al.*(1981)也將插入法應用在 VRP

上，兩者都利用了 3-opt 交換法改善求解的品質。

3. 巨集式啟發式演算法(Metaheuristics)

傳統啟發式演算法雖然可以減少搜尋過程，使其在特定的時間限制內求得合理的可行解，但是產生新解的規律只與執行方法時所收集到的資訊有關。由於起始解或搜尋方法上的限制，當無法找到任何更佳的目標值時搜尋過程便會停止，因此傳統啟發式演算法所得到的最終解，往往只是區域最佳解。相反地，巨集式啟發式演算法允許那些會導致較差目標值的移步，使得搜尋範圍不會侷限於某個區域內而得到區域最佳解。巨集式啟發式演算法的主要觀念，源自於傳統啟發式演算法，並加入了人工智慧的觀念，其中大部分的方法都是以物理或生物變化過程中的原則為基礎。以下將介紹巨集式啟發式演算法應用於求解 VRP 的相關文獻。

(1) 塔布搜尋法：

塔布搜尋法最早是由 Glover 於 1977 年所提出，在求解過程中，以一初始解為起點，然後產生一鄰近區域，並搜尋鄰近區域以求得最佳臨近解來作為下次搜尋的起點，演算過程會重複此步驟直到達到設定停止搜尋的條件，而搜尋的過程具有彈性記憶結構，可記錄之前已搜尋過的解於塔布列表(tabu list)中，以避免重複性的搜尋而缺乏效率，並大幅增加跳脫出區域最佳解的機會。

Willard (1989) 及 Pureza and Francam 在 1991 年時最早將塔布搜尋法應用在 VRP 當中，Willard 利用 2-opt 交換法或 3-opt 交換法來搜尋鄰近的區域。而 Pureza and Franca 的改善方式是對路徑間單一顧客的交換，或是嘗試將某一路徑中的顧客插入另一條路徑（羅敏華，2003）。Osman (1993) 結合 2-opt 交換法、將節點插入不同的路線與路徑間節點的交換等三種方法來搜尋鄰近的區域，並求解 VRP 基本的 14 個測試例題(Christofides *et al.*, 1979)，結果顯示在部分的例子中塔布搜尋法可以求得已知最佳解。

Gendreau *et al.* (1994)發展出更創新的塔布搜尋法並稱為Taburoute，而此法用以求解車輛途程問題能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，而以基本的14個測試例題比較的結果顯示，大部分例題的求解結果都相當接近已知之最佳解。Renaud *et al.* (1996)將塔布搜尋法應用於求解多個物流中心的VRP，並考慮車輛容量及路徑長度的限制，而發展出一新的演算法稱為FIND。測試23個標竿問題(Christofides and Johnson, 1969; Gillet and Johnson, Chao *et al.*,1993)後的結果顯示，在經過參數的調整後，FIND對於所有的標竿問題都可以達到或超越Chao *et al.*(1993)所找到的最佳解。

(2) 模擬退火法：

此法由 Kirkpatrick *et al.*於 1983 年提出，根據物理退火過程和求解最佳化問題之間的類比關係而發展出來。

Robusté *et al.* (1990) 最早將模擬退火法應用於 VRP。Osman (1993)發展出更複雜的模擬退火法來應用在 VRP 上，以移動一個或兩個顧客至其他路徑作為改善方法，最後測試的效果比 Robusté *et al.*(1990)的方法更好。Van Breedam (1995)利用三種途程間的改善法，再加入模擬退火法的機制來做改善，作者測試 14 個測試例題後，將結果與其他學者所提出的塔布搜尋法和混合式演算法比較，其結果雖然並非每一個例題都有較佳表現，但是所得到的解均比只以單純執行這三種改善法還要好。

(3) 基因演算法：

基因演算法是由 Holland 於 1975 年所提出，其主要理論是根據達爾文進化論中「物競天擇，適者生存」的原則發展而來，以基因演化的過程來尋求最佳解，其過程包含複製、交配與突變的變化。

Baker (2003)將基因演算法應用於求解 VRP，並同時針對原始的基因演算法及混合後的基因演算法作測試，結果其求得的解雖然不如諸位學者所提出的塔布搜尋法，但是差異不大，且在求算部份例題時具有較短的求

解時間。

(4) 蟻群演算法：

蟻群演算法(ant colony optimization, ACO)的初始概念最早是由 Colomi *et al.* (1991)提出，並由 Dorigo 於 1992 年在其博士論文完成架構(熊鴻鈞，2003)，關於蟻群演算法的原理及演算步驟將於 2.3 節中詳細說明。蟻群演算法的主要概念是模擬自然界中蟻群利用費洛蒙機制來求出最短路徑的覓食行為。Bullnheimer *et al.* (1997) 將蟻群演算法的前身—蟻群系統(ant system)應用在 VRP，稱為 ANT-VRP。作者認為在 ANT-VRP 中應建構兩個步驟：車輛分派及費洛蒙更新。在建構 ANT-VRP 時，首先應以蟻群系統應用於旅行銷售員問題(travel salesman problem, TSP)的方式進行路線選擇與費洛蒙更新的動作，當各路徑上的機率值已達收斂時，再接續派車的動作。此外，作者在研究中還利用了 2-opt 交換法作為區域搜尋的機制。

Gambardella *et al.* 在 1999 年推翻了先前 ANT-VRP 的觀念，提出以多目標的方式 (multiple objectives) 求解 VRP。作者加入了時間窗的限制並設定 VRP 的兩個目標：總派車數最小化以及總派車距離最小化，而最小化車輛數將間接可達成總派車距離最小化的目標 (蔡志強，2003)。

2.2 蟻群演算法

本節先概述蟻群演算法的原理，接著針對蟻群演算法的演算步驟作詳細的說明，最後則是簡單介紹其相關應用。

2.2.1 蟻群演算法的理論

蟻群演算法的主要原理源自於對自然界蟻群覓食行為的觀察。根據研究，螞蟻屬於社會性昆蟲的一種，具有分工合作的社會行為。在日常生活常常可以看到，螞蟻沿著同一條路徑去搬運食物，並再沿著同一路徑將食物搬回巢穴，然而牠們並不是透過視覺來達到搜尋路徑的目的，而是透過其所分泌的一種化學物質－費洛蒙，來進行不同螞蟻之間間接的溝通。研究結果顯示，在移動的過程時，螞蟻會分泌費洛蒙並殘留在走過的路徑上，當越多的螞蟻行經相同路徑時，則該路徑的費洛蒙濃度會隨著量的累積而越來越高。當後來的螞蟻在選擇路徑時，便會受到費洛蒙濃度的影響，濃度較高的路徑越有機會吸引螞蟻通過並進而留下更多的費洛蒙。然而路徑上的費洛蒙除了因為螞蟻殘留會增加外，還會受到溫度、溼氣及空氣對流等環境因素環境影響而逐漸消失，這種現象則稱為費洛蒙的蒸發。在螞蟻較少經過的路徑上，其費洛蒙數量最終會因為蒸發而消失，而較多螞蟻行經的路徑，雖然也會有費洛蒙蒸發，但是由於較多數量的螞蟻持續殘留費洛蒙，使得費洛蒙累積的速度大於蒸發的速度，因此最後路徑上的費洛蒙濃度會增加而不會減少，經過不斷的反覆後，大部分的螞蟻便會選擇相同的路徑（Bell and McMullen 2004，羅敏華 2003）。

如圖2所示，在遇到岔路或是障礙物時，螞蟻一開始有相同的機率去嘗試各種可能的路徑，而由於較短的路徑較能夠快速地累積費洛蒙，因此就越能吸引後續的螞蟻也通過該路徑，當經過一段時間後，所有的螞蟻幾乎都會選擇相同路徑，這便是螞蟻尋找最短路徑的原理。

螞蟻演算法就是藉由模擬真實螞蟻的行為，學習螞蟻移動搜尋的模

式，釋放出人工螞蟻(artificial ant)進行最佳決策的搜尋，在每一世代找出更佳的解，並在最後產生一個最佳解。在Dorigo *et al.* (1997)的文獻中提到，這些用來求解的人工螞蟻會模擬自然界真實螞蟻行為的三種特性：

- (1) 螞蟻會傾向選擇擁有較高費洛蒙濃度的路徑。
- (2) 較短的路徑其費洛蒙的累積速度會較為快速。
- (3) 不同的螞蟻透過費洛蒙達到間接溝通的效果。

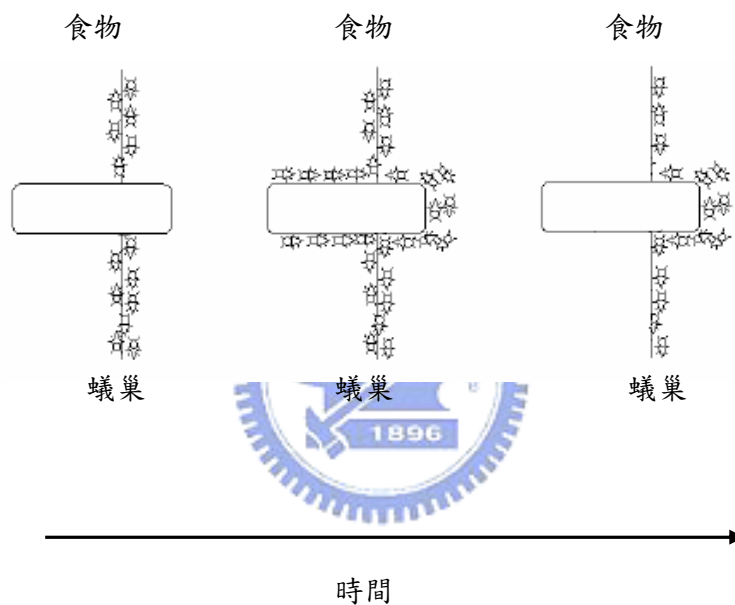


圖 2 真實螞蟻搜尋路徑的概念圖

資料來源：修改自尹邦嚴、王敬育，2004

2.2.2 蟻群演算法的演算步驟

蟻群演算法最初是被運用在解決 TSP，並擁有相當卓越的表現，當時的蟻群演算法被命名為螞蟻系統，在經由不斷的改良及演進後成為現在的蟻群演算法。

執行蟻群演算法的架構可利用虛擬碼表示如下（羅敏華，2003）：

設定參數值與初始費洛蒙濃度

主迴圈(loop)

次迴圈(sub-loop)

利用狀態轉移法則建構路線。

進行區域費洛蒙濃度更新(local pheromone update rule)。

直到所有的螞蟻路線皆產生完成。

進行全域費洛蒙濃度更新(offline pheromone update rule)。

重覆以上步驟，直到達到所設定的停止條件。

在此並以求解 TSP 為例，說明蟻群演算法的詳細步驟。

步驟一：設定初始狀態及各參數的值。

在此一步驟需設定總迭代數(TN)、費洛蒙初始值(τ_0)、費洛蒙濃度權重參數(α)、費洛蒙蒸發係數(ρ , $0 < \rho < 1$)、螞蟻數量(m)、能見度權重參數(β)及狀態轉移法則參數(q_0)。

步驟二：建立各螞蟻的旅程

每隻螞蟻首先會隨機選擇一個城市作為旅程的起點，並接著會依據如式(2-1)與式(2-2)的狀態轉移法則(state transition rule)來選擇下一個拜訪的城市，然而已經被拜訪過的城市則不能再次選擇。

$$j = \arg \max_{u \in S(i)} \{(\tau_{iu})^\alpha (\eta_{iu})^\beta\} \quad (2-1)$$

$$P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{u \in S(i)} (\tau_{iu})^\alpha (\eta_{iu})^\beta} \quad (2-2)$$

其中 τ_{ij} 為目前所在城市 i 到下一個可能的城市 j 路徑上的費洛蒙濃度； η_{ij} 為目前所在城市 i 到城市 j 路徑長度的倒數，又稱為能見度； α 與 β 則分別為 τ_{ij} 與 η_{ij} 的權重參數； $S(i)$ 為螞蟻位在城市 i 時，尚未拜訪過之城市的集

合； q_0 為一給定參數， q 則為一個服從 $U(0,1)$ 的隨機變數。當 $q \leq q_0$ 時，螞蟻會選擇擁有最大 $(\tau_{iu})^\alpha (\eta_{iu})^\beta$ 值且尚未被拜訪過的城市，如式(2-1)所示；當 $q \geq q_0$ 時，螞蟻會依照式(2-2)中的機率分配函數 P_{ij} 來選擇尚未被拜訪的城市，當某一城市的 $(\tau_{iu})^\alpha (\eta_{iu})^\beta$ 值越大時，該城市被螞蟻選中的機率也會越大，意即螞蟻會較傾向選擇路徑較短和費洛蒙濃度較高的城市。透過式(2-1)與式(2-2)，每隻螞蟻有可能參考過去所經歷的較佳路徑，此行為稱為「開發」(exploitation)；同時也可能根據能見度與費洛蒙濃度的機率分配函數，來隨機地選擇尚未嘗試過的路徑組合，這種行為則稱為「探索」(exploration)。

步驟三：費洛蒙更新

為了能夠在未來得到更佳解，螞蟻所殘留的費洛蒙濃度必須更新，以實際地反映各螞蟻的表現及他們求得的解的品質。費洛蒙更新是蟻群演算法學習性功能中最關鍵的一項因素，其可以確保後來的解能夠有所改善。費洛蒙更新法則包含了兩個部份：區域費洛蒙更新與全域性費洛蒙更新。區域性費洛蒙更新在建構過程中，只要螞蟻行經的路線，皆立即更新費洛蒙濃度(如式(2-3)所示)，透過模擬自然界中費洛蒙的蒸發，可以確保不會有任何一條路徑凌越其他路徑，並增加螞蟻發現其他可行解的機會。其中 τ_0 為路徑上的初始費洛蒙濃度， ρ 則為介於0與1之間的費洛蒙蒸發係數。

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + (\rho)\tau_0 \quad (2-3)$$

當所有螞蟻都建構完可行的路徑後，便執行全域性的費洛蒙更新(如式(2-4)所示)，從所有螞蟻產生的解中，選出最好的路線作費洛蒙濃度的增加，其餘路線由於並沒有螞蟻行經，因此作費洛蒙濃度的蒸發。透過全域性費洛蒙更新可以促使螞蟻選擇距離較短的路徑，或是有較高的機率去

選擇最佳解中的部份路段來建構為來的新路徑。其中 L 為此次迭代中的最佳路徑。

$$\tau_{ij}^{new} = (1 - \alpha)\tau_{ij}^{old} + \alpha(L)^{-1} \quad (2-4)$$

步驟四：測試停止條件

當演算法達到終止條件時便停止，否則回到步驟二繼續搜尋。

2.2.3 蟻群演算法的相關應用

蟻群演算法除了在早期應用在TSP以及前述的VRP、非等效平行機台問題外，還被應用來解求許多其他NP-hard問題及各種組合最佳化問題如：二次指派問題（quadratic assignment problem）、各種排程問題（scheduling problem）、著色問題（graph coloring problem）、網路途程問題（networks routing problem）、連續性順序問題（sequential ordering problem）、最短母字串問題（shortest common super-sequence problem）、一般分配問題（general assignment problem）及複合背包問題（multiple knapsack problem）等，且經過實驗證實蟻群演算法應用在這些問題上均可獲得品質良好的解（蔡志強，2003）。以下將簡單介紹蟻群演算法應用於與本研究較為相關的排程問題上。

2.2.3.1 蟻群演算法應用於排程問題

Colormi *et al.* (1994) 首先將蟻群系統應用在排程問題上，他們為了解決以最小完工時間為目標的零工式生產排程問題，而提出所謂的AS-JSP 演算法，研究結果顯示當工件與機器數小於15時，此演算法所找出的解均能達到與最佳解差距在10%以內的水準。

Stützle (1998) 則研究如何將蟻群演算法應用在流程式生產(flow-shop) 排程問題，並發表了MMAS-FSP演算法，研究結果顯示此演算法可以求得不錯的解。Bauer *et al.* (1999) 則考慮了單機總延遲時間排程問題（single

machine total tardiness problem) ，他們提出的演算法ACS-SMTTP 省略了費洛蒙的區域更新，並加入了多種啟發函數來協助求解，如：EDD (earliest due date) 法則與MDD (modified due date) 法則，而求解100個工件之問題的測試結果顯示，ACS-SMTTP 具備求得近似最佳解的能力。

2.3 供應鏈整合之相關文獻

過去已有不少作者研究供應鏈管理中的相關整合問題，Thomas and Griffin (1996)整理了供應鏈整合的相關文獻且作深入的分析，他們將供應鏈整合的模型分為三種類型，分別為買方-賣方協調整合(buyer-vendor coordination) 、製造-配送整合(production-distribution coordination)以及存貨-配送整合(inventory-distribution coordination)，並指出雖然分別討論產品製造及成品配送的文獻相當豐富，但是鮮少有同時考慮這兩個階段的模型，原因在於這類問題較不易求解，且這兩個階段往往被緩衝的存貨所分開並由不同的部門來管理。此外，作者也建議對於此類供應鏈整合問題應有更多研究將重點放在作業層面而非策略層面。

Chen and Vairaktarakis (2005)也指出，大部分的製造—配送—存貨模型都只將焦點放在管理者決策的策略層面，而較少研究細部排程的整合決策。而且在這些模型中，製造與配送階段是經由存貨間接連結，存貨成本也佔了相當高的比例。而這些模型已不符合現今企業採用直接銷售模式以降低存貨成本的概念，因此必須發展更多的學術研究來建構製造與配送間直接互動的模型，以及實務上可行的求解技巧。

Lee and Chen (2001)探討了包含運送部份的機台排程問題，並針對兩種不同類型的問題作研究：第一類問題為半成品經由自動搬運車運送到下一個需加工的機台，其中排程部分為雙機台的流程式生產類型，運送部分考慮了一部或多部車輛及不同的容量限制，問題目標則是最小化最大完工時間；第二類問題討論加工完畢後的成品運送到顧客或是倉庫，機台排程

部分考慮了單一機台、雙機平行機台及雙機流程式生產，運送部份除了單一機台情形下考慮多部車輛外，其餘皆只探討單一車輛的情況，問題目標則是最小化最大完工時間或總完工時間。作者針對各種不同問題，釐清其問題複雜度，或是提出可在多項式時間內求解的演算法。

Hall and Potts (2002)結合了供應鏈管理與排程的概念，討論供應鏈中各種排程、批次及運送的整合問題。有別於以往的排程文獻，作者除了分別以供應商與製造商的觀點來探討決策問題外，還考慮了這兩者的整合決策，以求整體系統成本的最小化。在排程問題部分，作者將一個供應商或製造商視為一個單一機台；在批次運送部份，假設車輛沒有容量上限，而同一個顧客所下的訂單屬於同一個批次，一部車輛一次只會將成品運送到一個顧客點，因此不會涉及車輛途程決策的問題，但必須考慮發車數量所造成的成本。作者針對供應商、製造商及整合的觀點，在追求不同目標式的情況下找出問題的複雜度，並為可在多項式時間內求解的問題提出動態規劃解法。作者並以舉例方式說明藉由整合供應商及製造商的決策，在某些情況下至少可以降低系統總成本的20%，甚至降低成為原來的一半。

Chang and Lee (2004)探討 Lee and Chen (2001)所研究的第二類問題，其中針對機台數量及顧客區域數假設了三種情境：單一機台運送至單一顧客區域、二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。此外還加入車輛容量的限制，並考慮訂單體積的大小。Li *et al.* (2003)針對 Chang and Lee (2004)之研究作了延伸，將所欲服務的顧客數延伸到多個。他們探討單一機台排程整合單一車輛途程決策的問題，假設有多個固定數目的顧客，以最小化訂單交貨時間的總和為目標，並在特定簡化條件下使用動態規劃求得最佳解，作者建議未來的研究方向可以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。Chang and Lee(2003)探討數種整合兩階段的問題，其中包括 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

使用的方法來求解此整合問題，並以最劣情況 (worst-case) 衡量分析這兩種方法的表現。其中向前求解法(forward approach)即是先求解系統問題中的第一個階段，再求解第二個階段；相反地，向後求解法(backward approach)則是依據求解第二階段的資訊來求解第一階段的問題。Chen and Vairaktarakis (2005)探討整合生產排程及運送的模型，排程部分考慮了單一機台或等效平行機台的情況，訂單在加工之後會送到一個或多個顧客手上，運送部份則是屬於批次運送，同一批次內可能包含不同顧客的訂單。問題的目標函數為最小化加權後訂單送達時間與運送成本的總和，其中訂單送達時間同時考慮最大或平均送達時間。作者將不同的機台類型、顧客數與目標函數做搭配提出了八種不同的問題類型，並釐清每種問題的複雜度或對可於多項式時間內求解的問題提出以動態規劃為基礎的演算法。此外作者也研究整合兩階段可能帶來的利益，並發展了一個整合的啟發式方法，在分別與依傳統順序式方法(sequential approach)求解此種兩階段問題的結果比較後，發現在大部分的情況下，整合的方法都有超過 5% 的改善幅度。

Garcia and Lozano (2005)探討流程式生產工廠中，兩個加工站之間半成品排程與運送的問題，其中兩站的排程類型皆為等效平行機台，類似 Lee and Chen (2001)提出的第一類問題，此外還加入時窗限制，問題的目標式則是最大化在時窗限制下，工件加工完成所獲得的利潤。對於此一問題，他們以巨集式啟發式演算法中的塔布搜尋法來求解，並將求得的解與精確求解法的最佳解作比較，結果顯示塔布搜尋法可以在很短的運算時間內，求得高品質的解。

陳昱皓 (2006)探討整合製造—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 VRP 並假設系統中可供運送產品的車輛數目固定。在最小化加權後訂單送達時間之總和的目標下，作者

利用塔布搜尋法求解，並以加權後最短加工時間訂單優先處理法則 (weighted shortest processing time first rule, WSPT) 為基礎來產生起始解，測試結果顯示塔布搜尋法相對於起始解平均約有 20% 的改善率。周碩鴻 (2006) 考慮了與陳昱皓 (2006) 相同的問題，但利用基因演算法來求解，與 WSPT 為基礎產生之起始解作比較的結果顯示基因演算法的改善率會隨著訂單的數量而提升，最高約有 36% 的改善率。

整合製造—配送兩階段問題雖然其基本概念相同，都是透過整合排程與配送來達到最小化系統總成本的目的，但是針對不同的模擬情境或實務上的需求，近年來學者們研究的問題模型也越來越多元化。本研究將現有整合兩階段問題相關文獻的類型整理分類如表 1 所示：



表 1 整合製造—配送兩階段問題相關文獻之整理

作者	產品製造階段	成品配送階段	目標	求解方法
Lee and Chen (2001)	流程式生產	單一或多部車輛	最小化最大	啟發式演算法
	單一機台、雙機 平行機台及雙機 流程式生產	除了單一機台外皆只考慮 單一車輛 單一顧客	完工時間、最 小化總完工 時間。	
Hall and Potts (2002)	單一機台	單一車輛 單一顧客	最小化系統 總成本	動態規劃
Chang and Lee (2003)	流程式生產	單一或多部車輛	最小化最大 完工時間、最 小化總完工 時間。	向前求解法 與向後求解 法輔以最劣 情況分析
	單一機台	單一車輛 單一顧客	最小化最大 延遲時間	
Chang and Lee (2004)	單一機台運送至單一顧客區域		最小化最大	啟發式演算 法輔以最劣 情況分析
	二台等效機台運送至單一顧客		完工時間	
	單一機台運送至兩個顧客			
Li <i>et al.</i> (2003)	單一機台	單一車輛 單一顧客或多個顧客	最小化訂單 交貨時間的 總和	動態規劃
Chen and Vairaktarakis (2005)	單一機台及等效 平行機台	多部車輛批次運送 單一顧客或多個顧客	加權後訂單 送達時間與 運送成本的 總和	動態規劃 啟發式演算 法

Garcia and Lozano (2005)	包含等效平行機台的流程式生產	多部車輛且考慮時窗限制	最大化在時窗限制下加工獲得之利潤	塔布搜尋法
陳昱皓 (2006)	非等效平行機台	多部車輛但車輛數目固定 多個顧客	加權後訂單送達時間的總和	塔布搜尋法
周碩鴻 (2006)	非等效平行機台	多部車輛但車輛數目固定 多個顧客	加權後訂單送達時間的總和	基因演算法

從兩階段相關文獻整理的結果可以發現，大部分的研究為了簡化問題的困難度，在製造階段只考慮單一機台、完全相同平行機台或是等效平行機台，而除了陳昱皓 (2006)與周碩鴻 (2006)外，幾乎沒有其他文獻討論到非等效平行機台與車輛途程的整合。在陳昱皓 (2006)與周碩鴻(2006)所探討的配送階段中，系統中可使用的車輛數固定且不具有容量限制。為了研究更接近實際情況的模型，本論文將探討整合非等效平行機台與多部考慮具容量限制之車輛配送的兩階段問題，並使用蟻群演算法來求解問題。

第三章 問題定義與研究方法

本研究探討的問題由兩個階段所組成，第一階段為非等效平行機台的排程問題，而第二階段為成品配送的車輛途程問題。問題的目標函數為最小化各訂單的總完成時間乘上其權重轉換而成的成本。各訂單的完成時間定義為訂單配送至該顧客的時間，而訂單的權重由設計者依訂單重要性自行決定，重要性高的訂單相對地給定較高的權重。

針對此問題，本研究提出以蟻群演算法為基礎的演算架構，其步驟為先決定所有訂單的排程，再針對排程的結果分配批次，最後決定各批次的車輛路徑。

3.1 問題定義

本研究討論的問題敘述如下：系統內有 n 個訂單分別為 n 個不同的顧客所訂購，為方便敘述本研究假設訂單 i 是由顧客 i 所訂購 ($\forall i$)，每個訂單的體積均為 1 個單位，需經在工廠中之 m 個非等效平行機台中之任一機台加工，訂單 i 在機台 k 的加工時間為 p_{ik} 。被機台處理完成之訂單需經由車輛運送至該顧客處，車輛的容量限制為 l 單位，同一部車輛所裝載之訂單視為一個批次。此外為了最小化車輛使用數，本研究假設所有車輛在裝載訂單至容量上限後才進行配送的動作。訂單從一開始加工直到送達顧客手上所經過的時間間隔，即為總完成時間，以 c_j 表示。 d_{oi} 為物流中心至顧客 i 的配送時間， d_{ij} 為顧客 i 到顧客 j 的配送時間。問題目標函數為 Min

$$Z = \sum_{j=1}^n w_j c_j$$

其中 w_j 表示訂單 j 之權重，表示訂單間之相對重要性，問題

中需決定的相關決策包括訂單加工的順序、訂單在哪一台機台上加工、批次分配以及各批次內訂單配送的路徑。而本研究所探討的問題具有下列的最佳化性質 (optimality property)。

定理 1 對於本研究所探討的問題，在下列情況下會存在一個最佳的排程：

- i. 在排程部份中，當每部機台上加工的訂單間沒有閒置時間時，會存在一個最佳的排程。
- ii. 當每個批次（即每部車輛）的發車時間為該批次內的最大完工時間時，會存在一個最佳的排程。
- iii. 當同一批次內的訂單為連續完工，即同一批次內各訂單完工時間的區間內沒有其他批次的訂單完工時，會存在一個最佳的排程。

證明

- i. 假設機台在加工訂單時會有閒置時間，則我們可以將後續訂單的開始加工時間往前移動，而不會增加系統的總成本。
- ii. 假設某一個批次的發車時間不是該批次的最大完工時間時，此批次的發車時間則可以往前移動至該批次的最大完工時間。而當此批次擁有較早的發車時間時，本問題的系統總成本不會因此而增加。
- iii. 假設現有 a 、 b 、 c 、 d 、 e 等 5 個訂單，其完工順序依序為 a - b - c - d - e ，若存在一個已知為最佳的排程及其分批方式為訂單 a 、 b 、 d 屬於第一個批次，訂單 c 、 e 屬於第二個批次，則我們可以將訂單 c 與訂單 d 交換，使得第一個批次的發車時間可以提早至訂單 c 的完工時間，而第二個批次的發車時間不變。在此一調整步驟後，本問題的系統總成本不會因此而增加。

另外，為求簡化問題的複雜性，本研究訂定問題的基本假設如下：

一、機台加工階段

1. 所有訂單時間起始皆相同（時間原點 0）；
2. 機台在時間為 0 時皆可供使用；
3. 機台加工無整備成本；

4. 不考慮重工或當機；
5. 任一機台同一時間只能加工一個訂單；
6. 每一工件在各機台上的加工時間皆為已知；
7. 每一顧客只訂購一筆訂單。

二、車輛配送階段

1. 單一物流中心（即為工廠）；
2. 可使用之車輛數目無上限；
3. 所有車輛在時間為 0 時皆可供使用，並停放於工廠處；
4. 每一部車的容量限制皆相同；
5. 每一部車的速度皆相同；
6. 車輛行走距離無限制；
7. 每一部車必須裝載訂單至容量上限後才可進行配送；
8. 每部車只運送一次；
9. 顧客位置為已知。



3.2 演算法之設計

本小節將介紹非等效平行機台部份的編碼方式，接著說明狀態轉移法則及費洛蒙更新法則，最後則是本研究提出之蟻群演算法架構的詳細步驟。

3.2.1 非等效平行機台問題之轉換

由於蟻群演算法的邏輯是根據距離及費洛蒙濃度構成的機率法則來選擇下一個行經的節點，因此為了讓蟻群演算法能適用在非等效平行機台部份，本研究參考陳民俊（2003）的編碼方式，將訂單與機台間的對應關係轉換為新的作業編號，而每一個作業編號則相當於一個節點，以 5 個訂單與 2 台機台的對應關係為例，其作業編號如表 2 所示：

表 2 作業編號表

編號	1	2	3	4	5	6	7	8	9	10
訂單	1	2	3	4	5	1	2	3	4	5
機台	1	1	1	1	1	2	2	2	2	2

透過這種轉換方式，可以同時將機台與訂單納入考量，並建構出排程順序的費洛蒙矩陣。因為在平行機台問題中，每一個訂單只需要在一台機台上加工，因此在轉換成作業編號後，同一個訂單所對應的作業編號只能被選擇一次，例如表 2 中的編號 1 與編號 6 皆是代表訂單 1 的作業，但所指派加工的機台不同，所以在決定排程時只能選擇其中一個，當其中一個編號被選擇後（意即訂單 1 已排入排程中），另一個編號便不可以再選。而轉換後建構出的排程費洛蒙矩陣大小為訂單數×編號數。

3.2.2 狀態轉移法則

為了獲得最佳解，螞蟻必須能夠參考過去所經歷過的較佳路徑（在本研究中為一組排程或配送路徑），這種行為稱為「開發」(exploitation)；同時螞蟻也必須能夠發現還未嘗試過的解答組合，而這種行為稱之為「探索」(exploration)。開發的主要功能為改善現有的解，然而這種行為有可能會使螞蟻在求解時陷入區域最佳解中，因此可以藉由探索的機制使螞蟻跳脫至另一個未曾嘗試的解空間，讓螞蟻有機會尋求更好的解。而為了使螞蟻兼備開發與探索的行為，本研究採用Dorigo and Gambardella (1997) 所提出的狀態轉移法則(state transition rule)，來作為螞蟻選擇下一個節點的依據（在排程部份的節點為作業編號；在配送部份的節點則為欲配送的顧客點）。而本研究提出之演算法的求解過程中，共使用兩次轉移法則，分別用於求解排程部份的加工排程，以及求解配送部分的車輛路徑。

排程部份的狀態轉移法則如式(3-1)與式(3-2)所示：

$$j = \arg \max_{u \in S_k(i)} \{(\tau_{iu})^\alpha (\eta_{iu})^\beta\} \quad (3-1)$$

$$P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{u \in S_k(i)} (\tau_{iu})^\alpha (\eta_{iu})^\beta} \quad \text{if } j \in S_k(i) \quad (3-2)$$

其中 i 為螞蟻目前所在的作業順序(即排程中進行的第 i 項作業)， $S_k(i)$ 為第 k 隻螞蟻在作業順序 i 時，尚未被選擇過且可行的作業編號集合， j 為目前決定選擇的作業編號， P_{ij} 表示在第 i 個作業順序會選擇作業編號 j 的機率， τ_{ij} 則是在第 i 個作業順序選擇作業編號 j 的機率費洛蒙濃度， η_{iu} 為在第 i 個作業順序選擇作業編號 j 的能見度。而在執行狀態轉移法則時，必需要先設定一個介於 0 與 1 之間的常數 q_0 ，以控制螞蟻選擇開發及探索的機率。螞蟻要選擇下一個作業時，會先隨機產生一個服從 $U(0,1)$ 的隨機變數 q ，當 $q \leq q_0$ 時，螞蟻會選擇 $S_k(i)$ 內擁有最大 $(\tau_{iu})^\alpha (\eta_{iu})^\beta$ 的作業編號，如式(3-1)所示，此即為螞蟻的開發行為；當 $q > q_0$ 時，螞蟻會依照式(3-2)中的機率分配函數 P_{ij} 來選擇下一個作業編號，而此即為螞蟻的探索行為，當作業編號的 $(\tau_{iu})^\alpha (\eta_{iu})^\beta$ 值越大時，該編號被螞蟻選中的機率也會越大，因此也可稱為輪盤法則。在排程部份中，本研究將能見度定義為 $\eta_{iu} =$

$$\frac{w_{O(u)}^\gamma}{F_u + \Delta d}$$

其中 $O(u)$ 表示作業編號 u 所代表的訂單， γ 代表訂單權重在能見度的重要程度， F_u 是作業編號 u 的完工時間點，等於作業編號 u 所對應機器的可開始加工時間加上其處理時間， Δd 則是作業編號 u 所代表的訂單(顧客)與上一個加工之訂單(顧客)間的距離。如此定義能見度可確保螞蟻在選擇下一作業時，能夠同時考慮下一個作業編號對應訂單的權重、

完工時間，以及與上一個加工訂單間的距離，使得完工時間與地理位置相近的訂單能夠盡量分配到同一個批次，或優先加工具有較高權重的訂單。

而配送部分的狀態轉移法與排程部分相似，差別在於兩者的參數定義不同。配送部分的狀態轉移法則如式(3-3)與式(3-4)所示。

$$b = \arg \max_{u \in S_{kv}(a)} \{(\tau_{au})^\alpha (\eta_{au})^\beta\} \quad (3-3)$$

$$P_{ab} = \frac{(\tau_{ab})^\alpha (\eta_{ab})^\beta}{\sum_{u \in S_{kv}(a)} (\tau_{au})^\alpha (\eta_{au})^\beta} \text{ if } b \in S_k(a) \quad (3-4)$$

其中 a 為車輛目前所在的顧客位置， $S_{kv}(a)$ 為第 k 隻螞蟻的第 v 輛車位於顧客 a 的位置時，尚未被配送過的訂單集合， b 為下一個決定選擇的顧客位置， P_{ab} 表示車輛在訂單 a 的位置時會選擇訂單 b 做為下一個配送點的機率， τ_{ab} 則是在訂單 a 的位置會選擇訂單 b 做為下一個配送點的費洛蒙濃度。螞蟻要選擇下一個訂單作為配送點時，需先隨機產生一個服從 $U(0,1)$ 的隨機變數 q ，當 $q \leq q_0$ 時，螞蟻會選擇尚未配送的訂單中有最大 $(\tau_{ab})^\alpha (\eta_{ab})^\beta$ 的作業編號，如式(3-3)所示；當 $q \geq q_0$ 時，螞蟻會依照式(3-4)中的機率分配函數 P_{ab} 來選擇下一個配送的訂單。在配送部份中，本研究將能見度 η_{ab} 定

義為 $\eta_{ab} = \frac{w_b^\gamma}{d_{ab}}$ ，其中 d_{ab} 為顧客 a 與顧客 b 顧客所在位置間的距離，而如

此定義可使螞蟻為車輛安排下一個配送點時，有較高的機率會選擇權重較高的訂單，或是距離目前所在位置較近的訂單。

3.2.3 費洛蒙更新法則

蟻群演算法在找尋最佳解時，必須仰賴費洛蒙所帶來的間接互動過程，來達到學習或經驗交換的目的。而費洛蒙更新法則又可分為：(一) 費洛蒙區域更新與 (二) 費洛蒙全域更新兩種。茲分別說明如下：

(一) 費洛蒙區域更新

費洛蒙區域更新在求解過程中，只要是螞蟻行經的路徑（在本研究中為一組排程或配送路徑）便會立即更新費洛蒙濃度（如式(3-5)所示），其目的在於避免產生過於強勢的路徑，抑制了適當的探索新路徑行為，進而導致最後求得的解落入區域最佳解中。在本研究方法中，每隻螞蟻都配有兩種費洛蒙矩陣，分別是對應排程部份的排程費洛蒙矩陣及對應配送部份的路徑費洛蒙矩陣。當螞蟻在決定排程時找到下一個加工的訂單便會執行排程費洛蒙矩陣的區域更新；同理當螞蟻在決定配送路徑時，找到下一個配送點時便會執行路徑費洛蒙矩陣的區域更新。

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + (\rho)\tau_0 \quad (3-5)$$

其中 ρ 為表示介於[0,1]的區域費洛蒙蒸發係數； τ_0 為初始費洛蒙濃度。

(二) 費洛蒙全域更新

在同一迭代中，當所有的螞蟻都完成可行解的建構後，便執行費洛蒙全域更新（如式(3-6)所示），增加目前最佳解的費洛蒙濃度，藉此強化該解的參考性。而透過這種獎賞較佳解的方式，可以引導往後迭代中的螞蟻根據這些較佳解的路徑進行開發或探索。

$$\tau_{ij}(t+1) = (1 - \rho')\tau_{ij}(t) + \rho'\tau_{ij}^* \quad (3-6)$$

其中 t 表示迭代數； ρ' 表示介於[0,1]的全域費洛蒙蒸發係數； τ_{ij}^* 表示路徑 (i,j) 因每一迭代中目標函數值最小所額外增加的費洛蒙濃度，其計算方式如式(3-7)所示：

$$\tau_{ij}^* = \begin{cases} \frac{Q}{L_{gb}} & \text{if } (i,j) \in \text{best solution} \\ 0 & \text{otherwise} \end{cases} \quad (3-7)$$

其中 L_{gb} 表示所有螞蟻中最佳解的目標函數值， Q 一般設定為一固定參數，而在本研究中， Q 值設定為第一次迭代中最佳的解的目標函數值乘上初始費洛蒙濃度，而在本研究中， Q 值設定為第一次迭代中最佳的解的目標函數值乘上初始費洛蒙濃度透過這種更新方式，當最佳解的目標函數值越小時，其因獎賞而增加的費洛蒙濃度也會相對的越高。而全域費洛蒙蒸發係數使得費洛蒙會隨著時間蒸發，其目的在於避免某些路徑上費洛蒙無限的累積。此外，如同費洛蒙區域更新，在本研究方法中，當螞蟻找到一組完整的可行解時，會同時執行排程與路徑費洛蒙矩陣的全域更新。

3.2.4 螞蟻演算法架構

本研究所提出的演算法架構以定理 1 為基礎，並可大致分為兩個階段，第一階段是先利用螞蟻求出一組排程，指派各機台加工的訂單以及加工順序。第二階段則是將訂單依照完工的順序分配到車輛，當一部車輛裝載的訂單個數到達其容量上限時，便指派到另一部新的車輛，車輛的發車時間為所裝載之訂單中，最後完工訂單的完工時間（定理 1）。最後求出各車輛的配送路徑，而這部份可以將各車輛的路徑視為多個車輛途程子問題。演算法流程如圖 3 所示，而詳細的步驟說明如下：

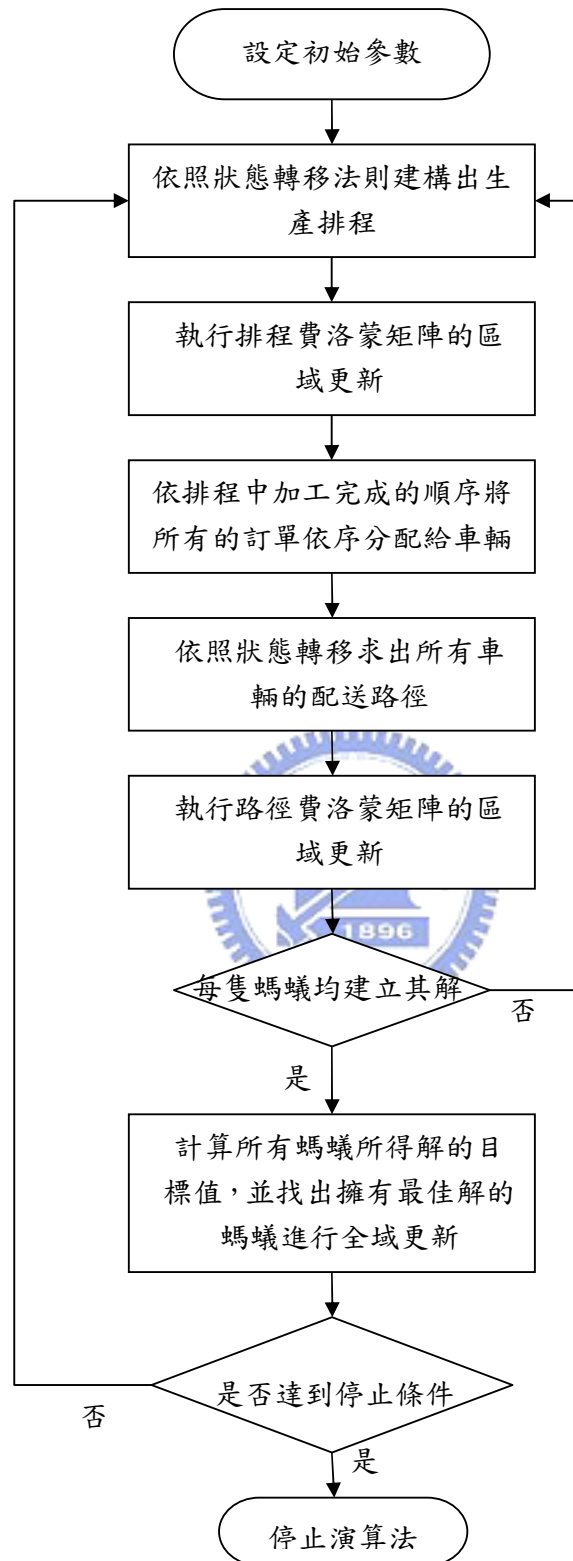


圖 3 演算法架構流程圖

步驟 1：設定初始狀態及參數

設定總迭代數(TN)、費洛蒙初始值(τ_0)、狀態轉移法則參數(q_0)、費洛蒙濃度權重參數(α)、能見度權重參數(β)、訂單權重參數(γ)、區域費洛蒙蒸發係數(ρ , $0 < \rho < 1$)、全域費洛蒙蒸發係數(ρ' , $0 < \rho' < 1$)、螞蟻數量(ANT)。

步驟 2：選擇排程起始點

每一隻螞蟻隨機選擇一個作業編號為起始點，並將其紀錄到作業順序的第一個位置中，並將其他代表相同訂單的編號移除。

步驟 3：建立訂單的生產排程

每隻螞蟻從排程的第二個作業開始，依照如式(3-1)與式(3-2)的狀態轉移法則選擇下一個作業編號，直到所有的訂單都被納入排程為止。

步驟 4：費洛蒙區域更新—排程費洛蒙矩陣

當螞蟻求出所有訂單的排程後，便依照式(3-5)進行排程費洛蒙矩陣的區域費洛蒙更新。

步驟 5：分配批次

依據先完工之訂單先分配的原則，將訂單依序分配給車輛，當目前進行分配的車輛達到容量上限時，便將訂單分配至下一部新的車輛。

步驟 6：各車輛的配送路徑

此一步驟類似步驟 3，當螞蟻決定每部車輛所裝載的訂單為何後，各車輛以物流中心（工廠）為起始點開始建構配送路徑，依照如式(3-3)與式(3-4)的狀態轉移法則選擇下一個配送點。

步驟 7：費洛蒙區域更新—路徑費洛蒙矩陣

當螞蟻求出所有車輛的路徑後，便依照式(3-5)進行路徑費洛蒙矩陣的區域費洛蒙更新。

步驟 8：全域費洛蒙更新

當螞蟻求出所有車輛的路徑後，便形成一組完整的解。計算所有螞蟻的解的目標值，並選出擁有最小目標值的螞蟻進行全域費洛蒙更新，而此一步驟包括更新排程的費洛蒙矩陣以及路徑的費洛蒙矩陣，如式(3-6)與式(3-7)所示。

步驟 9：測試停止條件

當演算法達到停止條件時($t=TN$)則停止，否則回到步驟 2

3.2.5 演算步驟範例說明

本小節將以一小規模的問題為例，說明演算法在求解問題時的流程。考慮 6 個訂單($n = 6$)、2 台機台($m = 2$)、車輛容量上限為 3 個訂單($l = 3$)的問題，而為簡化敘述，本研究在此假設訂單 i 由顧客 i 所訂購($i = 1, 2, 3, \dots, 6$)。各訂單權重、加工時間以及轉換後的作業編號如表 3 所示，各訂單所屬顧客與物流中心間的配送時間矩陣如表 4 所示。

表 3 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	訂單 6
w_i	3.1	2.5	2.9	4.7	1.8	6.2
p_{i1} (作業編號)	15 (1)	40 (2)	34 (3)	18 (4)	27 (5)	45 (6)
p_{i2} (作業編號)	25 (7)	22 (8)	36 (9)	42 (10)	18 (11)	30 (12)

表 4 訂單之配送時間矩陣

d_{ij}	物流中心	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
物流中心	0	137	174	89	78	54	96
訂單 1	137	0	115	110	81	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	81	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 1：設定初始狀態及參數

設定總迭代數 $TN = 30$

費洛蒙初始值 $\tau_0 = 0.0001$

費洛蒙濃度權重參數 $\alpha = 1$

能見度權重參數 $\beta = 0.9$

訂單權重參數 $\gamma = 1$

區域費洛蒙蒸發係數 $\rho = 0.1$

全域費洛蒙蒸發係數 $\rho' = 0.15$

螞蟻數量 $ANT = 5$

狀態轉移法則參數 $q_0 = 0.5$

步驟 2：選擇排程起始點

以 5 隻螞蟻中的螞蟻 1 為例，假設其隨機選取的排程起始點為作業編號 3，意即訂單 3 於機台 1 加工，則同樣代表訂單 3 的作業編號 9 必須從 $S_2(1)$ 移除。

步驟 3：建立訂單的生產排程

從作業順序 2 開始，螞蟻 1 的 η_{2j} 經由計算如表 5 所示， P_{2j} 經由計算如表 6 所示。接著產生一隨機變數 q ，若 $q < 0.5$ 則螞蟻 1 會直接選擇表 6 中具有最大 P_{2j} 的編號 10，若 $q > 0.5$ 則由式(3-2)來決定，意即 P_{2j} 越大的作業編號被螞蟻選擇的機率會較高。

表 5 排程部分能見度資料表

作業編號	計算過程	η_{2j} 值
1	$\frac{3.1}{(34+15)+89}$	0.0225
2	$\frac{2.5}{(34+40)+69}$	0.0174
4	$\frac{4.7}{(34+18)+36}$	0.0534
5	$\frac{1.8}{(34+27)+76}$	0.0131
6	$\frac{6.2}{(34+45)+99}$	0.0348
7	$\frac{3.1}{25+89}$	0.0272
8	$\frac{2.5}{22+69}$	0.0275
10	$\frac{4.7}{42+36}$	0.0603
11	$\frac{1.8}{18+76}$	0.0191
12	$\frac{6.2}{30+99}$	0.0481

表 6 排程部分狀態轉移機率表

作業編號	p_{2j} 值
1	0.0499
2	0.0386
4	0.1183
5	0.0290
6	0.0771
7	0.0603
8	0.0609
10	0.1336
11	0.0423
12	0.1066

步驟 4：區域費洛蒙更新—排程費洛蒙矩陣

假設螞蟻 1 所建立之排程中的作業編號為{3, 10, 8, 11, 6, 1}

則排程費洛蒙矩陣需要更新的部份如圖 4 中的灰色部分所示。而螞蟻 1 的排程甘特圖如圖 5 所示

作業編號

	1	2	3	4	5	6	7	8	9	10	11	12
作業 順序	1											
	2											
	3											
	4											
	5											
	6											

圖 4 排程費洛蒙矩陣區域更新

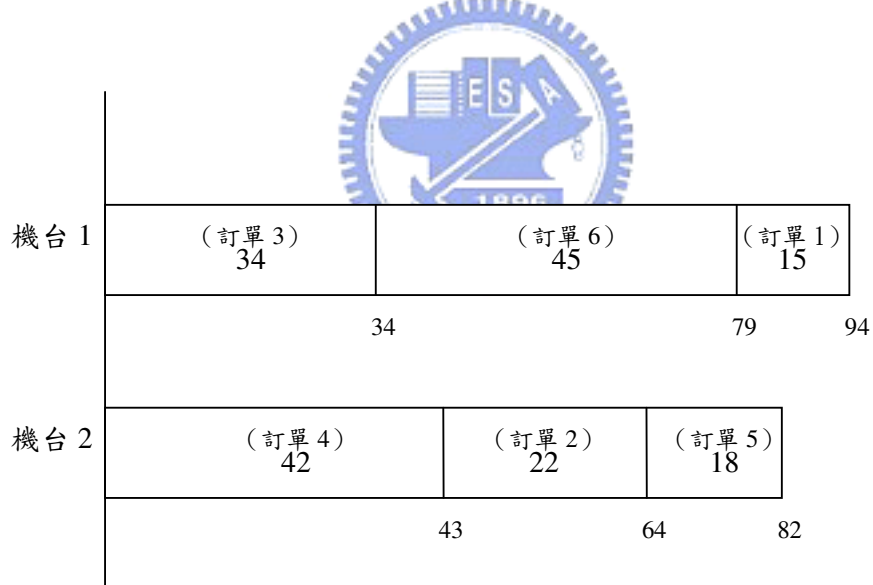


圖 5 排程甘特圖

步驟 5：分配批次

依據先完工之訂單先分配的原則，將訂單 3、訂單 4、訂單 2 分配給車輛 1，其發車時間為 64；將訂單 6、訂單 5、訂單 1 分配給車輛 2，其發車時間為 94。

步驟 6：各車輛的配送路徑

在此步驟，螞蟻 1 將決定車輛 1 裝載之訂單 3、訂單 4、訂單 2 的配送順序，以及車輛 2 裝載之訂單 6、訂單 5、訂單 1 的配送順序，因此可以視為兩個車輛途程子問題。以車輛 1 的路徑為例，其 η_{ob} 及 P_{ob} 的相關資料及算如表 7 及表 8 所示。接著產生一隨機變數 q ，若 $q < 0.5$ 則螞蟻 1 會直接選擇表 8 中具有最大 P_{ob} 的訂單 4，若 $q > 0.5$ 則由式(3-4)來決定，意即 P_{ob} 越大的訂單被螞蟻選擇的機率會較高。

表 7 配送部份能見度資料表

訂單	計算過程	η_{ob} 值
3	$\frac{2.9}{89}$	0.0326
4	$\frac{4.7}{78}$	0.0603
2	$\frac{2.596}{174}$	0.0144

表 8 配送狀態轉移機率表

訂單	P_{ob} 值
3	0.2206
4	0.4081
2	0.0975

步驟 7：區域費落蒙更新—路徑費洛蒙矩陣

假設螞蟻 1 決定車輛 1 的配送路徑為：物流中心—訂單 4—訂單 3—

訂單 2，車輛 2 的配送路徑為：物流中心—訂單 1—訂單 5—訂單 6，
則路徑費洛蒙矩陣需要更新的部份如圖 6 中的灰色部分所示。

	物流中心	訂單 1	訂單 2	訂單 3	訂單 4	訂單 5	訂單 6
物流中心							
訂單 1							
訂單 2							
訂單 3							
訂單 4							
訂單 5							
訂單 6							

圖 6 路徑費洛蒙矩陣區域更新

步驟 8：全域費洛蒙更新

計算所有螞蟻的解，並選出擁有最小目標值的螞蟻進行全域費洛蒙更新，如 3.2.2 節中的式(3-6)與式(3-7)所示。

步驟 9：測試停止條件

當演算法達到停止條件時 ($t = TN$)則停止，否則回到步驟 2。

第四章 演算法測試與結果分析

本研究使用 C++ 語言，將第三章中所設計的蟻群演算法（以下簡稱為 ACO_PDP）撰寫成電腦程式，以測試其求解能力與相關特性，測試的硬體環境為 AMD 64 Sempron 2800+ CPU，1G RAM。

第四章共分為四個小節，其中 4.1 節利用實驗的方式找出演算法中各項參數的最佳設定值；4.2 節測試 ACO_PDP 在求解單一機台且所有訂單權重皆相同的特例時，所得之解與真實理想解的差距，用以評估 ACO_PDP 的求解能力；4.3 節說明本研究測試問題的設計方法；4.4 節則是比較 ACO_PDP 與呂學君(2007)所設計之以禁忌搜尋法為基礎的 Tabu_PDP 演算法於求解測試問題的能力，並探討問題設定對求解品質的影響。

4.1 參數設計

本研究考量問題複雜度與程式的執行時間，選擇以一個 $n = 80$ 、 $m = 4$ 、 $l = 20$ 之問題作為測試的標準，問題的加工時間與配送時間分別為服從 $U(20,80)$ 及 $U(150,250)$ 的隨機亂數。本研究分別針對 ANT 、 τ_0 、 α 、 β 、 ρ 、 ρ' 、 q_0 及 TN 等參數進行測試分析，對每個參數的數值進行 10 次的重複測試，並依據程式執行結果選擇在 ACO_PDP 中最適用的參數值。在測試的過程中，本研究以 $ANT = 5$ （5 隻螞蟻）、 $\tau_0 = 0.0001$ 、 $(\alpha, \beta) = (1, 1)$ 、 $\gamma = 0.05$ 、 $\rho = 0.9$ 、 $\rho' = 0.9$ 、 $q_0 = 0.9$ 作為預設的參數值來進行測試，而迭代數 TN 則設定為 80 代。在執行參數設定測試時，每次只變動單一參數，其他參數固定，以比較不同參數對求解品質的影響。本研究選擇最佳參數值的依據為程式執行 10 次的結果中，以其最小之目標函數值 ($Z = \sum_{j=1}^n w_j c_j$) 為主要考量標準，平均的目標函數值為參考標準，而 10 次執行結果之標準差則為次要參考標準。

4.1.1 訂單權重參數(γ)

γ 決定訂單權重在能見度中的重要程度，當 γ 越大時表示訂單權重對能見度的影響越大。針對 γ 的設定值，本研究探討 0 與 1 之間的 8 種數值，分別為 0、0.05、0.1、0.3、0.5、0.7、0.9 及 1。測試結果顯示(如圖 7 所示)，當 $\gamma = 0.1$ 時所求得的目标函數值最低，表現最為良好，因此本研究選定 $\gamma = 0.1$ 作為在測試ACO_PDP時所使用的參數值。

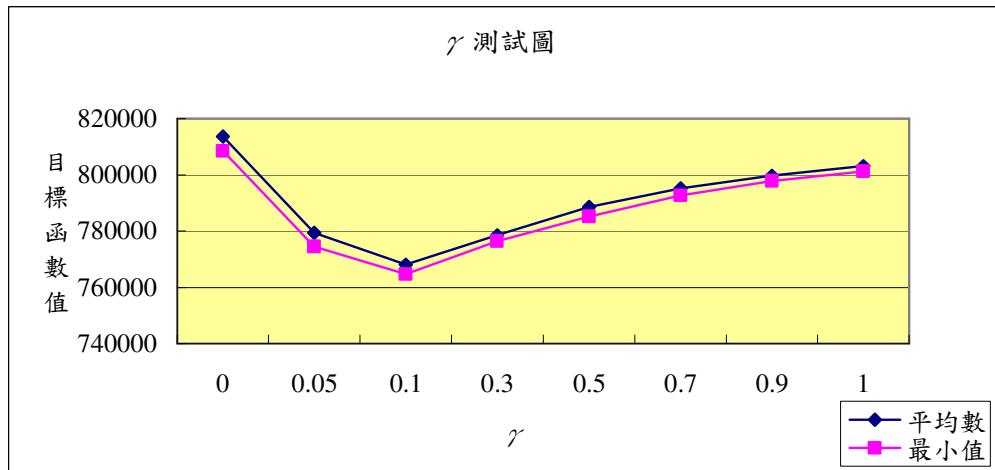


圖 7 訂單權重參數測試圖

4.1.2 螞蟻數量(ANT)

分別測試5、10、20及40隻螞蟻(如圖8所示)，程式執行結果發現以20隻螞蟻表現較好，5隻螞蟻與10隻螞蟻表現較差，而40隻螞蟻並沒有較佳的表現，求得之目標值與20隻螞蟻相當接近，但需耗費相當於20隻螞蟻2倍的運算時間，因此本研究選擇以20隻螞蟻作為在測試ACO_PDP時所使用的螞蟻數量。

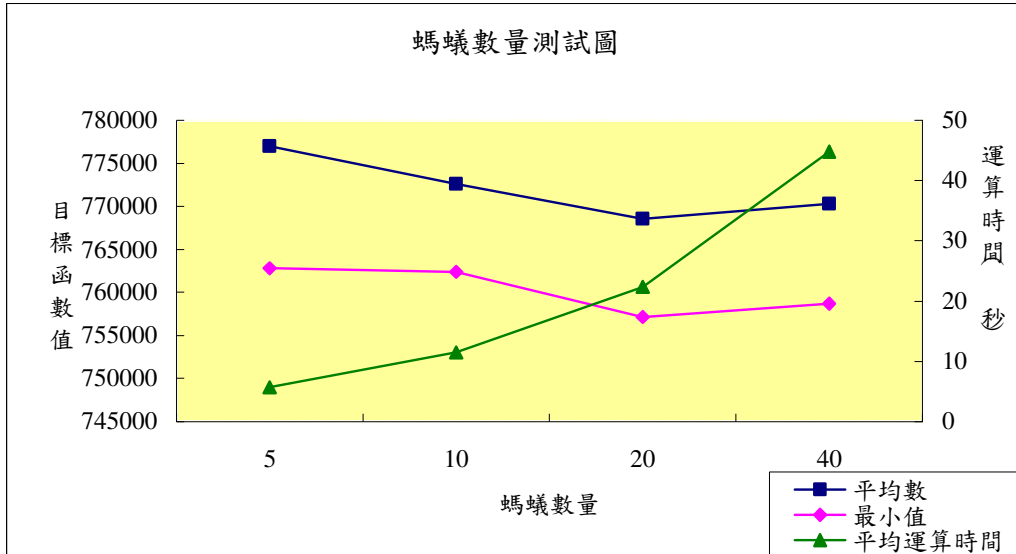


圖 8 螞蟻數量測試圖

4.1.3 費洛蒙濃度初始值(τ_0)

在實際的情況中，沒有螞蟻行經的路徑上其費洛蒙濃度的初始值應該為0，然而為了避免初始費洛蒙濃度過低而使得螞蟻降低探索新路徑的機會，本研究假設 τ_0 趨近於0，並分別嘗試將 τ_0 設定為 10^{-3} 、 10^{-4} 、 10^{-5} 與 10^{-6} 來做比較，根據測試結果顯示(如圖9所示)， $\tau_0 = 10^{-4}$ 時的表現優於其他三種數值。因此本研究選定 $\tau_0 = 10^{-4}$ 作為在測試ACO_PDP時所使用的參數值。

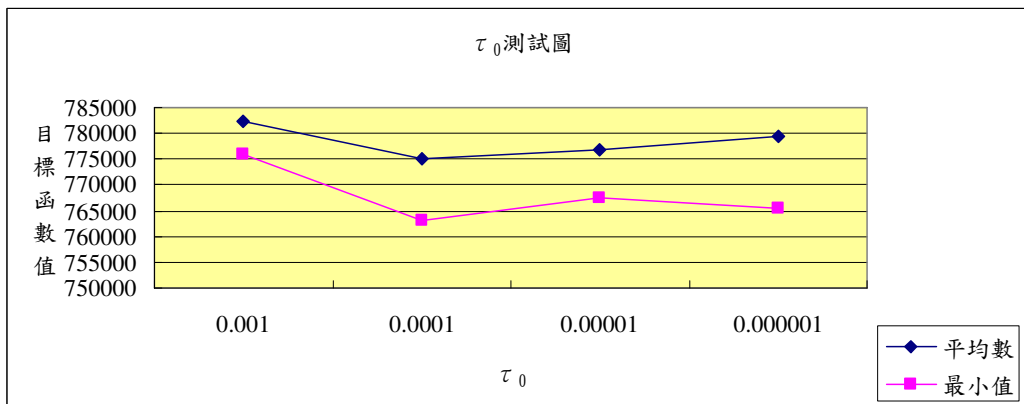


圖 9 費洛蒙濃度初始值測試圖

4.1.4 費洛蒙濃度與能見度參數(α, β)

α 、 β 決定螞蟻在選擇節點時，費洛蒙濃度以及能見度的重要程度，而在ACO_PDP中，費洛蒙濃度與能見度皆為小於1之數值，因此當 α 或 β 越小時表示其重要程度越高。本研究將{0.1,0.5,1,2,5}等五個不同層級的 α 、 β 數值搭配出16種組合，測試結果顯示(如圖10所示)，當 $(\alpha, \beta) = (1, 0.5)$ 時，其求得之最小成本及平均成本皆為參數組合當中表現最好的，因此本研究選定 $(\alpha, \beta) = (1, 0.5)$ 作為在測試ACO_PDP時所使用的參數值。

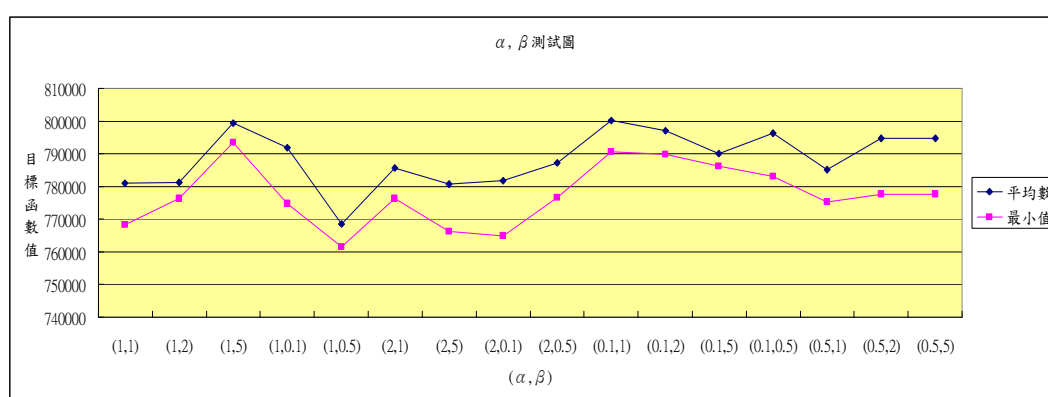


圖 10 費洛蒙濃度與能見度參數測試圖

4.1.5 參數 q_0

參數 q_0 的選擇，主要決定螞蟻建構路線時選擇開發與探索之比重，例如當 $q_0 = 0.3$ 時，螞蟻有30%的機會去選擇開發；而有70%的機會選擇機率探索。本研究測試0與1之間的8種數值，分別為0.1、0.3、0.5、0.7、0.9、0.95、0.99及1。測試結果顯示(如圖11所示)當 $q_0 \geq 0.9$ 時的表現比 $q_0 < 0.9$ 時的表現為佳，且 $q_0 = 0.9$ 、 $q_0 = 0.95$ 、 $q_0 = 0.99$ 與 $q_0 = 1$ 之目標函數值非常接近，而其中又以 $q_0 = 0.99$ 時表現最好，因此本研究選擇 $q_0 = 0.99$ 作為在測試ACO_PDP時所使用的參數值。

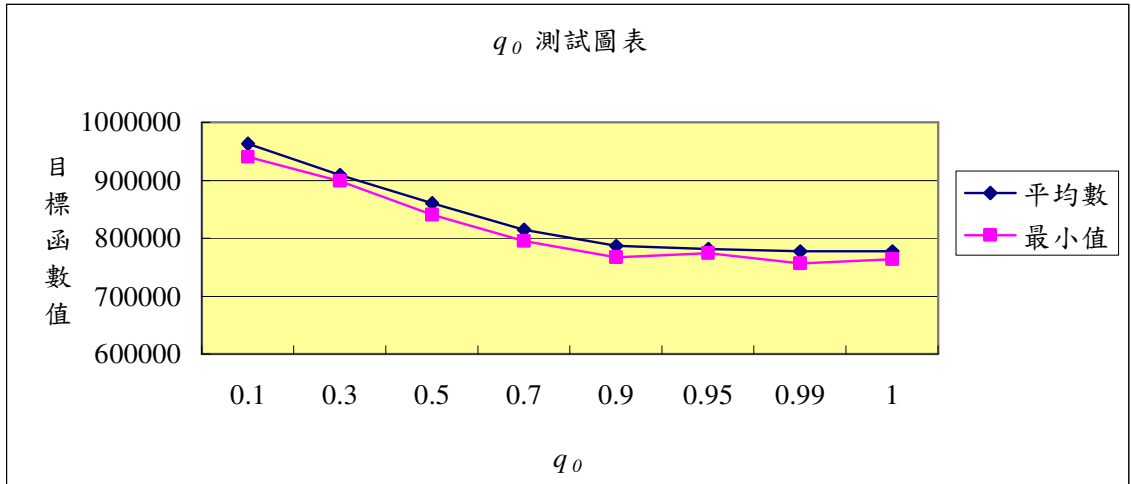


圖 11 參數 q_0 測試圖

4.1.6 區域費洛蒙殘留係數(ρ)

ρ 主要決定路徑上的費洛蒙濃度，在執行區域更新後保留的比例，本研究測試 0 與 1 之間的 8 種數值，分別為 0.1、0.3、0.5、0.7、0.9、0.99 及 1。測試結果發現(如圖 12 所示)，當 $\rho = 1$ 時表現最好，且與其他數值差距相當大，因此可以推論出本研究提出之演算法比較不適合執行區域更新，其原因可能為區域更新過度減弱費洛蒙濃度，而使得螞蟻降低去搜尋過去較佳路徑的機會。

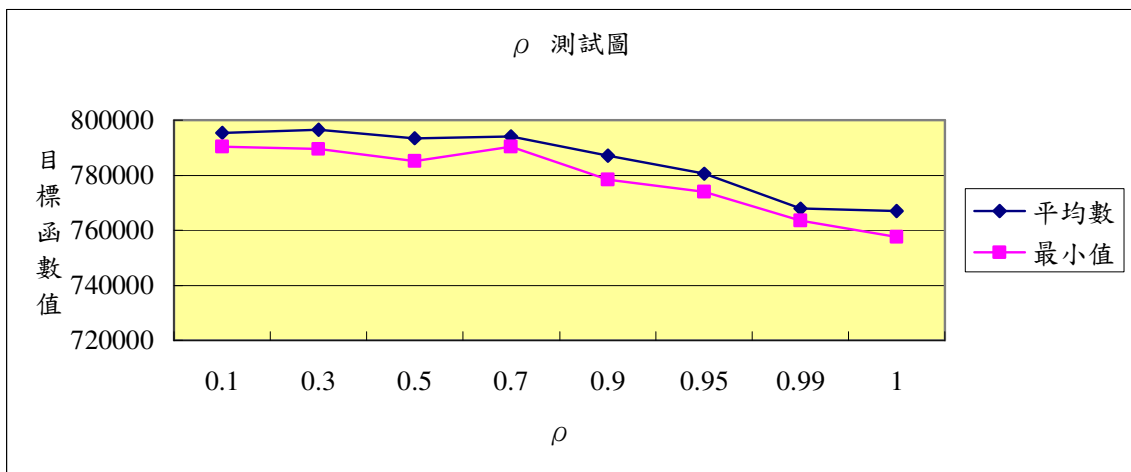


圖 12 區域費洛蒙殘留係數 ρ 測試圖

4.1.7 全域費洛蒙殘留係數 ρ'

參數 ρ' 主要決定路徑上的費洛蒙濃度，在執行全域更新後保留的比例。本研究根據4.1.7小節的結果測試在 $\rho=1$ 之情況下， ρ' 的值為0到1之間的8種數值，分別為0.1、0.3、0.5、0.7、0.9、0.95、0.99及1。測試結果顯示（如圖13所示）當 $\rho'=0.95$ 時求得之平均成本與最小成本皆為8種數值中最良好的，因此本研究選定 $\rho'=0.95$ 作為在測試ACO_PDP時的數值。

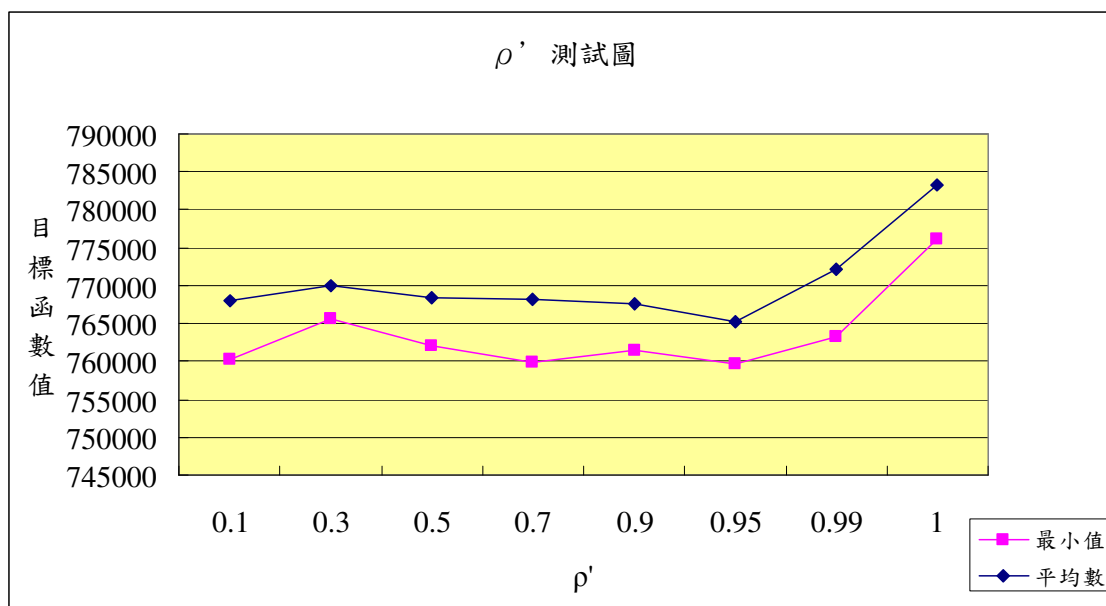


圖 13 全域費洛蒙殘留係數測試圖

4.1.8 迭代數(TN)

在確定以上各小節之參數設定值之後，本小節探討迭代數與運算時間、求解收斂程度間的關係。由於蟻群演算法在螞蟻數量固定的情況下，其求得最佳近似解所需之迭代數會隨著問題規模的擴大而增加，但是如果設定的迭代數太大而超過實際收斂的迭代區間，會使得運算時間過長且也無法求得最佳的解，因此本研究測試 $\frac{n}{4}$ 、 $\frac{n}{2}$ 、 n 、 $2n$ 等4種迭代數，以求出最適合的迭代數設定值。本研究分別以 $n=80$ ， $m=4$ ， $l=20$ 及 $n=200$ ， $m=4$ ， $l=20$ 的兩個問題為測試標準，而兩個問題換算後的四種迭代數分

別為20、40、80、160及50、100、200、400。測試結果顯示（如圖14所示） $n = 80$ 的問題在迭代數為80時即有不錯的表現，所求得之最小總成本為753993.2，運算時間為22.61秒；而當迭代數為160時所求得之最小總成本為752897.2，雖然有些許改善，但其運算時間為45.188秒，約為80迭代的兩倍。而 $n = 200$ 的問題測試結果也顯示（如圖15所示），當迭代數為200時的表現最佳，而迭代數為400時並無法求得更好的解，表示求解過程在第200次迭代左右已收斂。本研究為了同時兼顧演算法的求解品質與效率，因此將ACO_PDP測試之迭代數設定為與訂單數相等。

4.1.9 小結

根據以上各小節的參數測試結果，本研究決定將ACO_PDP於求解測試問題時的各項參數值設定為： $ANT = 20$ 、 $\tau_0 = 0.0001$ 、 $(\alpha, \beta) = (1, 0.5)$ 、 $\rho = 1$ 、 $\rho' = 0.9$ 、 $q_0 = 0.99$ 、 $TN = n$ 。

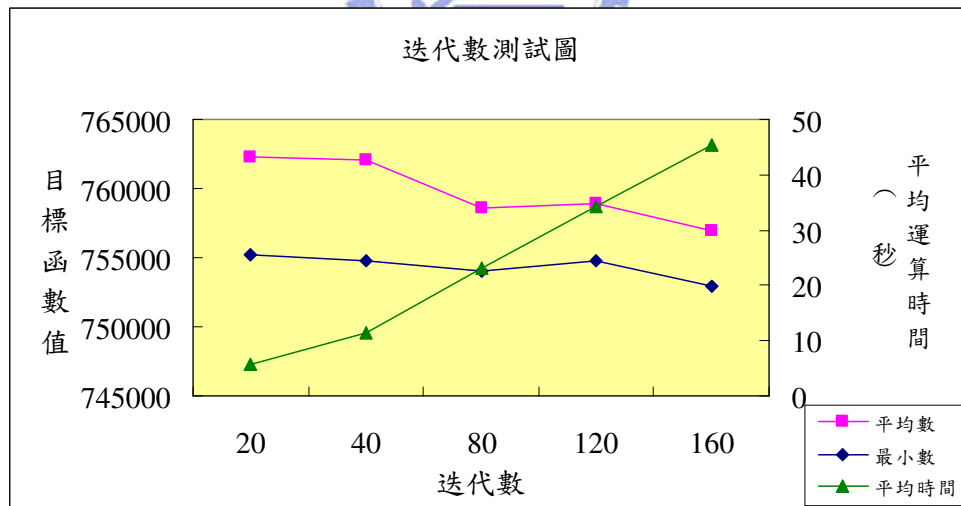
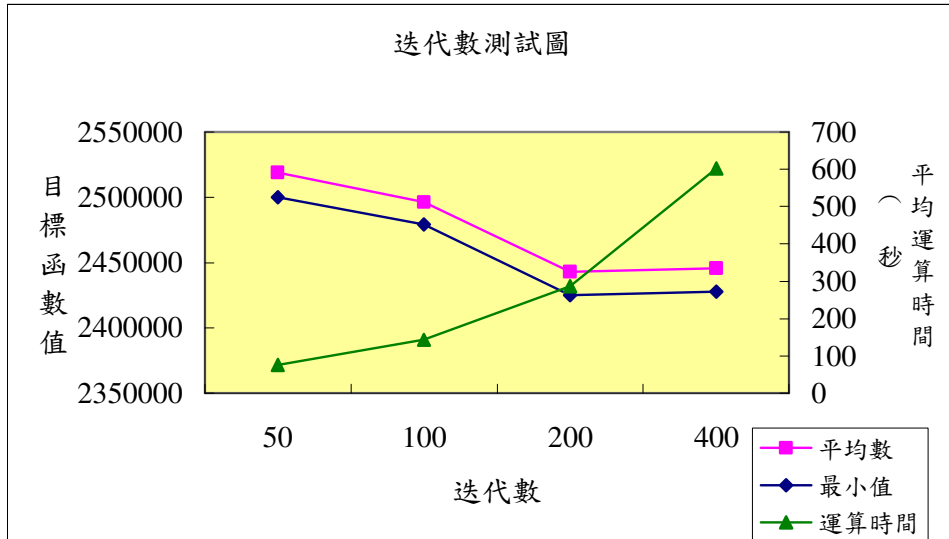


圖 14 $n = 80$ 迭代數測試圖



圖

圖 15 $n = 200$ 迭代數測試圖



4.2 ACO_PDP 之求解品質測試—以特殊問題為例

本節的目的在於測試 ACO_PDP 於求解問題時所能找出之最佳解與真實理想解的差距（即求解品質）。由於目前尚無法找出本研究所探討之問題的真实理想解，故在此節中，本研究考慮一個單一機台，且所有訂單權重皆相同的特例，以窮舉法找出其真实理想解，並與 ACO_PDP 所找出的最佳解比較以探討 ACO_PDP 的求解品質。本研究利用電腦隨機產生 $n = 10$ 、 $n = 20$ 及 $n = 30$ 三種訂單數量，搭配 $l = 5$ 、 $l = 10$ 兩種車輛容量限制的六個測試問題，訂單的加工時間與配送時間矩陣皆服從 $U(100,400)$ 的均勻分配，問題的真实理想解則是利用窮舉法來求得，而 ACO_PDP 的參數設定值則採用 4.1 節中所找出的最佳設定值。測試結果顯示（如表 9 所示），ACO_PDP 可求得測試問題 1、2、4、6 的真实理想解，而測試問題 3、5 的近似最佳解與真实理想解間的相對誤差均小於 1%，顯示本研究所提出之 ACO_PDP 具有良好的求解能力。

表 9 ACO_PDP 之求解品質測試表

測試問題	訂單數	車容限制	真實理想解	ACO	相對誤差(%)
1	10	5	20744	20744	0.000
2	10	10	31347	31347	0.000
3	20	5	50101	50444	0.685
4	20	10	67408	67408	0.000
5	30	5	131956	132158	0.153
6	30	10	162791	162791	0.000

4.3 測試問題設計

由於在目前的中外文獻中僅有呂學君(2007)探討與本研究相同之整合產品製造與成品配送的兩階段問題。呂學君設計了一個以禁忌搜尋法為基礎的演算法—Tabu_PDP，為了與其比較並探討 ACO_PDP 的求解能力，本研究採用與呂學君用來測試 Tabu_PDP 相同的測試問題。呂學君所設計之測試問題的輸入資料可以分為兩大部份，分別為顧客配送時間矩陣，以及訂單在產品製造階段的加工時間。其中配送時間矩陣代表各顧客與物流中心的相對距離關係，而加工時間則包含各訂單在不同機台上的加工時間。為了比較配送時間與加工時間的大小關係對求解結果的影響，他設定了三種不同的情境：(1)加工時間小於配送時間的情況（情境 S）；(2)加工時間與配送時間相當的情況（情境 M）；(3)加工時間大於配送時間的情況（情境 L）。針對上述的三種情境，本研究利用電腦隨機產生出服從均勻分配 (uniform distribution)， $U(150,250)$ 的配送時間矩陣，與服從 $U(20,80)$ 、 $U(150,250)$ 、 $U(700,900)$ 等三種不同大小的加工時間資料，而這三種加工時間資料的平均數及變異數成等比級數成長。

在訂單數量方面，呂學君的測試問題設定了 10、80 與 200 張訂單等三種之訂單數量 (n)；各訂單之權重設定為服從 $U(1,10)$ 的隨機亂數；機台數量 (m) 設定為 2 台、4 台及 8 台三種；車輛容量限制 (l) 分為小容量的 5 個訂單，與大容量的 20 個訂單。當訂單數量為 10 張時，僅考慮 2 台機台以及車輛容量較小的情況；當訂單數量為 80 時則考慮 2 台機台，4 台機台，並同時考慮大小車輛容量的情況；當訂單數量為 200 張時考慮 2 台機台、4 台機台、8 台機台，及大小兩種車輛容量的情況。

經由以上方式，呂學君的測試問題依加工時間規模、訂單數量、機台數量與車輛容量大小之間的搭配組合整理於表 10，共有 33 個測試問題，其中問題 1 到問題 11 為加工時間小於配送時間的情況；問題 12 到問題 22

為加工時間與配送時間相當的情況：問題 23 到問題 33 為加工時間大於配送時間的情況。為方便識別測試問題的性質，呂學君將其測試問題依設定的內容命名，而命名規則為：情境+訂單數_機台數_車輛容量限制。如測試問題 1 的訂單數為 10、機台數為 2、車輛容量限制為 5，而加工時間為 $U(20,80)$ 屬於情境 S，則測試問題 1 的名稱以 S10_2_5 表示。



表 10 測試問題設計表 (呂學君, 2007)

編號	測試問題	加工時間	訂單數(n)	機台數(m)	車容限制 (l)
1	S10_2_5	U(20,80)	10	2	5
2	S80_2_5	U(20,80)	80	2	5
3	S80_2_20	U(20,80)	80	2	20
4	S80_4_5	U(20,80)	80	4	5
5	S80_4_20	U(20,80)	80	4	20
6	S200_2_5	U(20,80)	200	2	5
7	S200_2_20	U(20,80)	200	2	20
8	S200_4_5	U(20,80)	200	4	5
9	S200_4_20	U(20,80)	200	4	20
10	S200_8_5	U(20,80)	200	8	5
11	S200_8_20	U(20,80)	200	8	20
12	M10_2_5	U(150,250)	10	2	5
13	M80_2_5	U(150,250)	80	2	5
14	M80_2_20	U(150,250)	80	2	20
15	M80_4_5	U(150,250)	80	4	5
16	M80_4_20	U(150,250)	80	4	20
17	M200_2_5	U(150,250)	200	2	5
18	M200_2_20	U(150,250)	200	2	20
19	M200_4_5	U(150,250)	200	4	5
20	M200_4_20	U(150,250)	200	4	20
21	M200_8_5	U(150,250)	200	8	5
22	M200_8_20	U(150,250)	200	8	20
23	L10_2_5	U(700,900)	10	2	5
24	L80_2_5	U(700,900)	80	2	5
25	L80_2_20	U(700,900)	80	2	20
26	L80_4_5	U(700,900)	80	4	5
27	L80_4_20	U(700,900)	80	4	20
28	L200_2_5	U(700,900)	200	2	5
29	L200_2_20	U(700,900)	200	2	20
30	L200_4_5	U(700,900)	200	4	5
31	L200_4_20	U(700,900)	200	4	20
32	L200_8_5	U(700,900)	200	8	5
33	L200_8_20	U(700,900)	200	8	20

4.4 測試結果分析

將 ACO_PDP 用於求解 4.3 節所列之 33 個測試問題後，ACO_PDP 的測試結果整理如表 11 所示，其中 Tabu_PDP 解與 Tabu_PDP 運算時間的數據取自呂學君(2007)的研究，而相對誤差的計算方式為：相對誤差=

$$\frac{|ACO_PDP - Tabu_PDP|}{Tabu_PDP} \times 100\%$$
，運算時間比例為 Tabu_PDP 求解運算時間除以 ACO_PDP 求解運算時間。

在求解品質方面，由表 11 可以看出在 33 個測試問題中，ACO_PDP 與 Tabu_PDP 求得問題 S10_2_5 相同之近似解；有 6 個問題 ACO_PDP 可求得比 Tabu_PDP 更好的近似解，分別為編號 S80_2_20、S80_4_20、M200_2_20、M200_4_20、L10_2_5 及編號 L80_4_20 的問題。而其他問題 ACO_PDP 的求解品質雖然較差，但與 Tabu_PDP 的相對誤差大部分在 5% 以內，平均相對誤差則約為 2.2%，顯示 ACO_PDP 在求解此兩階段問題時可得到與 Tabu_PDP 相當的解。另外值得注意的是，ACO_PDP 在編號 10 的問題（S200_8_5）的表現最差，與 Tabu_PDP 的解間之相對誤差約有 7.18%，高於整體平均誤差約 5%，本研究推論造成此一現象的原因為在此問題的輸入資料結構下，ACO_PDP 太快收斂於區域最佳解所導致。

在運算時間部份，ACO_PDP 在所有問題的求解運算時間皆遠小 Tabu_PDP，在 $n = 10$ 的問題中（問題 S10_2_5、M10_2_5、L10_2_5），ACO_PDP 所需要的運算時間甚至只有 Tabu_PDP 運算時間的百分之一；而在複雜度最高的 $n = 200$ ， $m = 8$ 的問題中（問題 S200_8_5、S200_8_20、M200_8_5、M200_8_20、L200_8_5、L200_8_20），ACO_PDP 也只需要 Tabu_PDP 運算時間的十分之一便可求得近似最佳解。

表 11 ACO_PDP、Tabu_PDP 求解能力比較表

編號	測試問題	Tabu_PDP		ACO_PDP		相對誤差 (%)	運算時間 比例 Tabu/ACO
		解	運算時間 (秒)	解	運算時間 (秒)		
1	S10_2_5	39573	5	*39573	0.047	*0.00	106.38
2	S80_2_5	454841	59	468626	9	3.03	6.66
3	S80_2_20	946240	53	*946142	9	*0.01	6.19
4	S80_4_5	298524	85	312605	18	4.72	4.63
5	S80_4_20	734209	77	*731832	19	*0.32	4.13
6	S200_2_5	2007077	2182	2042623	147	1.77	14.86
7	S200_2_20	3391388	2006	3392887	146	0.04	13.74
8	S200_4_5	1178337	3108	1196236	286	1.52	10.85
9	S200_4_20	2346394	3225	2424883	286	3.35	11.26
10	S200_8_5	776710	5650	832507	579	7.18	9.76
11	S200_8_20	1866694	5450	1907412	579	2.18	9.42
12	M10_2_5	77024	5	77301	0.031	0.36	161.29
13	M80_2_5	1407954	65	1447180	10	2.79	6.79
14	M80_2_20	2188612	60	2200766	10	0.56	5.90
15	M80_4_5	772868	100	805637	18	4.24	5.43
16	M80_4_20	1366754	78	1378544	19	0.86	4.21
17	M200_2_5	7813960	2036	8017391	148	2.60	13.77
18	M200_2_20	10005719	2047	*9991870	148	*0.14	13.81
19	M200_4_5	4084874	2756	4127208	289	1.04	9.55
20	M200_4_20	5697236	2958	*5686348	289	*0.19	10.22
21	M200_8_5	2247973	5675	2293409	585	2.02	9.70
22	M200_8_20	3623616	5473	3676867	585	1.47	9.36
23	L10_2_5	221122	5	*220341	0.031	*0.35	161.29
24	L80_2_5	5214798	59	5419479	10	3.93	5.91
25	L80_2_20	7074030	53	7100983	10	0.38	5.10
26	L80_4_5	2751850	84	2822676	19	2.57	4.52
27	L80_4_20	3815557	77	*3802247	19	*0.35	3.98
28	L200_2_5	31305407	2157	31903088	150	1.91	14.36
29	L200_2_20	35544369	2124	36290752	150	2.10	14.15
30	L200_4_5	15660434	2988	16035965	294	2.40	10.17
31	L200_4_20	18408377	3009	18654892	294	1.34	10.23
32	L200_8_5	8205672	5672	8338000	584	1.61	9.71
33	L200_8_20	10186398	5467	10227166	584	0.40	9.36

註:「*」表示 ACO_PDP 求得之近似解較好

本研究針對表 11 的測試數據使用複回歸分析(multiple regression)，以探討影響相對誤差的相關變項。複回歸分析結果如表 12 所示，由表 12 可以看出只有車輛容量限制對相對誤差有顯著影響，而加工時間分配、訂單數與機台數並無顯著影響。以下各小節對各種問題設定的影響作進一步的討論。

表 12 相對誤差複回歸分析表

Regression Summary for Dependent Variable相對誤差 (anova.sta)						
R= .60377562 R2= .36454500 Adjusted R2= .27376571						
F(4,28)=4.0157 p<.01069 Std.Error of estimate: 1.4662						
N=33	Beta	Std.Err. of Beta	B	Std.Err. of B	t(28)	p-level
Intercept			2.532196	0.939413	2.69551	0.011753
加工時間分配	-0.164756	0.150648	-0.341880	0.312604	-1.09365	0.283428
訂單數	0.156756	0.170363	0.003830	0.004163	0.92013	0.365363
機台數	0.230492	0.168379	0.180244	0.131672	1.36888	0.181918
車容限制	-0.523240	0.152872	-0.118694	0.034678	-3.42273	0.001926

4.4.1 加工時間與求解品質之關係

如表 12 的複回歸分析結果顯示，ACO_PDP 在(1)加工時間小於配送時間；(2)加工時間約等於配送時間；(3)加工時間大於配送時間等三種不同情境下的求解能力並無顯著差異。

4.4.2 機台數與求解品質之關係

由表 12 的複回歸分析的結果可以看出問題的機台數量對相對誤差並無顯著影響，顯示 ACO_PDP 的求解能力並不會受到機台數量的多寡影響。

4.4.3 訂單數與求解品質之關係

複回歸分析的結果顯示訂單數對相對誤差沒有顯著的影響，表示 ACO_PDP 的求解能力不會受到訂單數量影響。

4.4.4 車輛容量限制與求解品質之關係

複回歸的分析解果顯示，車輛容量限制對 ACO_PDP 求解品質有顯著影響。在 $l = 20$ 的問題中，ACO_PDP 與 Tabu_PDP 間的平均相對誤差約為

0.1%，而在 $l=5$ 的問題中平均相對誤差較高約為 2.2%，顯示 ACO_PDP 在車輛容量限制較小的問題上表現較差，本研究推論其原因可能是路徑費洛蒙矩陣的影響所導致：當車輛容量較小時，所使用的車輛數會相對較多，而每輛車的路徑都是彼此獨立的，但是在本研究所提出的演算法中，配送階段的路徑費洛蒙矩陣是所有車輛所共用的，因此原本應該互相獨立的各車輛路徑卻會被共同的費洛蒙矩陣影響，導致無法求得各車輛的最佳路徑，當車輛數越多時對求解品質的負面影響也就越大。此一缺陷為本研究當初設計演算法時所未考量的，因此建議未來後續研究的焦點可放在路徑費洛蒙矩陣的改善，讓所有車輛配有獨立的路徑費洛蒙矩陣，以提昇 ACO_PDP 在車輛容量限制較小問題上的求解品質。

4.4.5 小結

以上各小節的測試結果與分析顯示，ACO_PDP 的主要優勢為能快速地求出問題的近似最佳解求解：在求解 $n=10$ 的問題時，ACO_PDP 只需要花費約 Tabu_PDP 運算時間的 0.7% 便可求得近似最佳解，在求解 $n=80$ 及 $n=200$ 等訂單數量較大之問題時，ACO_PDP 的平均運算時間約為 Tabu_PDP 運算時間的 11%。在求解品質方面，雖然 ACO_PDP 在 $n=10$ 的問題上表現優異，但是當訂單數量(n)較大或車輛容量限制(l)較小時，Tabu_PDP 在大部分的問題皆能搜尋到品質較佳的近似解。複回歸分析的結果也顯示，車輛容量限制對 ACO_PDP 與 Tabu_PDP 間的相對誤差有顯著影響，表示 ACO_PDP 在車輛容量限制較小的問題設定下，其穩健性稍嫌不足，因此建議未來的相關研究能朝這方面著手。

第五章 結論與建議

5.1 結論

本研究探討整合供應鏈中產品製造及成品配送兩個階段的問題，隨著接單式生產及直接銷售等商業模式的盛行，這種整合製造—配送兩階段的模型也越來越受到學術界與企業界的重視，其可應用的範疇也相當廣泛，如電腦直銷、郵局、快遞服務、報社印刷報紙及派報或外送之餐飲服務業等，皆可以適用。不同於大部分的相關文獻只以理論的觀點探討整合製造—配送兩階段模型，本研究首次將蟻群演算法成功地應用於求解這類兩階段問題模型，並期望研究成果能作為相關產業實務上的參考依據，讓業者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的目的。

測試結果顯示本研究所設計的 ACO_PDP 在求解訂單數較小的問題時，不論求解品質或求解效率，都比呂學君(2007)所提出之 Tabu_PDP 好；當訂單數量較大或車輛容量限制較小時，ACO_PDP 的平均求解品質則稍微比 Tabu_PDP 差，平均誤差約為 1.83%，但是求解所需花費的時間平均約只有 Tabu_PDP 運算時間的 9%，顯示蟻群演算法在求解兩階段問題時，可以快速地求出近似最佳解，同時具備一定水準的求解品質，非常適合客戶需求反應時間極短的相關產業。

5.2 未來研究方向之建議

本小節針對問題模型及演算法兩方面，對未來後續研究的提出數點相關建議：

5.2.1 整合產品製造與成品配送兩階段問題方面

1. 可加入時窗限制、車輛路徑行使距離限制等條件，使問題更符合產業實際配送的情況。
2. 在訂單的體積方面可以考慮多種不同大小的體積。

3. 可以考慮車輛未裝載至容量上限的情況，並探討派送車輛成本與顧客服務水準間的權衡關係。

5.2.2 演算法方面

1. 本研究提出的蟻群演算法在較小車輛容量限制的問題上仍有改善的空間，適當地改良路徑費洛蒙矩陣的機制，將有助於提高跳脫區域最佳解的機率。
2. 可以讓蟻群演算法加入其他最佳化的機制，例如 2-opt、3-opt 等鄰域搜尋機制。
3. 可將蟻群演算法與其他巨集式演算法結合，例如以蟻群演算法求得的近似解作為禁忌搜尋法的起始解。
4. 對蟻群演算法的參數設定作更進一步的實驗分析，探討參數在各種問題上對求解品質的影響。



參考文獻

尹邦嚴、王敬育 (2004)。使用螞蟻族群最佳化求解資源分配問題。第一屆台灣作業研究學會學術研討會暨 2004 年科技與管理學術研討會。

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。

陳明俊 (2004)。蟻群演算法於具非等效平行機台考量之排程問題研究。私立大葉大學工業工程學系，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。

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。

熊鴻鈞 (2003)。螞蟻族群演算法於生產排成之應用。國立暨南國際大學資訊管理學系，碩士論文。

羅敏華 (2003)。蟻群最佳化演算法於載重限制之車輛途程問題的研究。私立元智大學工業工程與管理研究所，碩士論文。

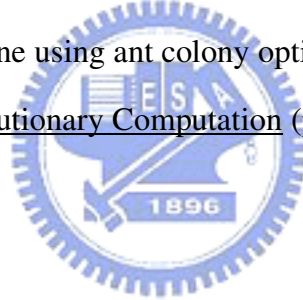
呂學君(2007)。應用禁忌搜尋法求解供應鏈中在考慮車容限制下整合產品製造與成品配送之問題。國立交通大學工業工程與管理學系，研究報告。

Allahverdi, A., J. Mittenthal. " Scheduling on M parallel machines subject to random breakdowns to minimize expected mean flow time ". Naval Research Logistics, 41 (5), pp. 677-682, 1994.

Azizoglu, M., O. Kirka (1999). Scheduling jobs on unrelated parallel machines to minimize regular total cost functions. IIE Transactions, 31, 153–159.

Baker, B. M. and M. A. Ayechev (2003). A genetic algorithm for the vehicle routing problem. Computers & Operations Research, 30, 787-800.

Bauer, A., B. Bullnheimer, R. F. Hartl, and C. Strauss(1999). Minimizing total tardiness on a single machine using ant colony optimization. In Proceedings of the 1999 Congress on Evolutionary Computation (pp. 1445-1450). Piscataway, New Jersey: IEEE Press.



Bell, J. E., P. R. McMullen (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 18, 41–48.

Bruno, L. G., Coffman, E. G. and Sethi, R. (1974). Scheduling independent tasks to reduce mean finishing time. Communications of the ACM, 17, 382–387.

Bullnheimer, B., R. F. Hartl, and C. Strauss (1997). Applying the ant system to the vehicle routing problem. Presented at the 2nd International Conference on Meta-heuristic-Mic97, Sophia-Antipolis, France.

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, Z. L., G. Vairaktarakis (2005). Integrated scheduling of production and distribution operations. Management Science, 51 (4), 614–628.

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.

Clarke, G., J. W. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12, 568-581.

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.

Colomi, A., M. Dorigo, V. Maniezzo, and M. Trubian (1994). Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL), 34 (1), 39-53.

Dhaenens-Flipo, C., G. Finke (2001). An integrated model for an industrial production-distribution problem. International Journal of Production and Economics, 74 (1-3), 93-101.

Dorigo, M., 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.

Eilon, S., C. Watson-Grandy, N. Christofides (1971). Distribution management: mathematical modeling and practical analysis. New York: Hanfer.

Garcia, J. M., S. Lozano (2005). Production and delivery scheduling problem with time windows. Computers & Industrial Engineering, 48, 733–742.

Gendreau, M., A. Hertz and G. Laporte (1994). A tabu search heuristic for the routing problem. Management Science, 40, 1276-1290.

Gillett, B. E., , L. R. Miller (1974). A heuristic algorithm for the vehicle dispatch problem. Operations Research, 22, 240-349.

Glass, C. A., C. N. Potts and P. Shade (1994). Unrelated parallel machine scheduling using local search. Mathematical and Computer Modeling, 20 (2), 41–52.

Hall, N. G. , C. N. Potts (2003). Supply chain scheduling: batching and delivery. Operations Research, 51, 566–584.

Hariri, A. M. A., C. N. Potts (1991). Heuristics for scheduling unrelated parallel machines. Computers and Operations Research, 18 (3), 323–331.

Horn, W. A. (1973). Minimizing average flow time with parallel machines. Operations Research, 21, 846–847.

Kim, D. W., K. H. Kim, W. Jang and F. F. Chen (2002). Unrelated parallel machine scheduling with setup times using simulated annealing. Robotics and Computer Integrated Manufacturing, 18 (3–4), 223–231.

Kolen, A. W. J., A. H. G. R. Kan and H. W. J. M. Trienkens (1987). Vehicle routing with time windows. Operations Research, 35 (2), 266-273.

Lee, Y.H., Bhaskaran, K., and Pinedo, M. (1997). A heuristic to minimize the total weighted tardiness with sequence dependent setups. IIE Transactions, 29, 45-52.

Lee, C. Y., Z. L. Chen (2001). Machine scheduling with transportation considerations. Journal of Scheduling, 4, 3–24.

Lenstra, J., A. Rinnooy Kan. (1981). Complexity of vehicle routing and scheduling problems. Networks, 11, 221-227.

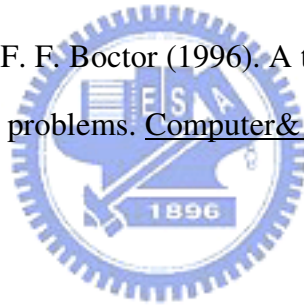
Li, C. L., G. Vairaktarakis, C. Y. Lee. (2003). Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research, 164, 39–51.

Mole, R.H., S.R. Jameson (1976). A sequential route-building algorithm employing a generalized savings criterion. Operational Research Quarterly, 27, 503-511.

Osman, I., (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, 41, 421-51

Garey R. and Johnson D. S. (1979). Computers and intractability: A guide to the theory of NP completeness. New York: W. H. Freeman and Company.

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.



Robusté, F., C. F. Daganzo, and R. Souleyrette II (1990). Implementing Vehicle Routing Models. Transportation Research, 24B, 263-286.

Sarmiento, A. M., R. Nagi (1999). A review of integrated analysis of production-distribution systems. IIE Transactions 31 (11), 1061-1074.

Srivastava, B. (1997). An effective heuristic for minimizing makespan on unrelated parallel machines. Journal of the Operational Research Society, 49, 886-894.

Stützle, T.(1998). An ant approach to the flow shop problem. In Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing(EUFIT'98) (pp. 1560-1564). Aachen: Verlag Mainz.

Suresh, V., C. Dipak (1994). Minimizing maximum tardiness for unrelated parallel machines. Production Economics, 34, 223-229.

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

Van Breedam, A. (1995) Improvement heuristics for the vehicle routing problem based on simulated annealing. European Journal of Operational Research, 86 (3), 480-490.



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.

Willard J. A. G. (1989). Vehicle routing using γ -optimal tabu search, Master of Science Dissertation, The Management School, Imperial College, London.

Wren, A., A. Holliday (1972). Computer scheduling of vehicles form one or more depots to a number of delivery points. Operational Research Quarterly, 23, 333-344.